



HAL
open science

Logistic optimization in disaster response operations

Juan Carlos Rivera Agudelo

► **To cite this version:**

Juan Carlos Rivera Agudelo. Logistic optimization in disaster response operations. Business administration. Université de Technologie de Troyes, 2014. English. NNT : 2014TROY0031 . tel-03358747

HAL Id: tel-03358747

<https://theses.hal.science/tel-03358747>

Submitted on 29 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse
de doctorat
de l'UTT

Juan Carlos RIVERA AGUDELO

**Logistic Optimization
in
Disaster Response Operations**

Spécialité :
Optimisation et Sûreté des Systèmes

2014TROY0031

Année 2014

THESE

pour l'obtention du grade de

**DOCTEUR de l'UNIVERSITE
DE TECHNOLOGIE DE TROYES
Spécialité : OPTIMISATION ET SURETE DES SYSTEMES**

présentée et soutenue par

Juan Carlos RIVERA AGUDELO

le 27 octobre 2014

Logistic Optimization in Disaster Response Operations

JURY

M. P. LACOMME	MAITRE DE CONFERENCES - HDR	Président
M. H. M. AFSAR	MAITRE DE CONFERENCES	Directeur de thèse
M. Á. CORBERÁN SALVADOR	PROFESOR CATEDRÁTICO	Rapporteur
M. O. PÉTON	MAITRE ASSISTANT	Examineur
M. C. PRINS	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. K. SORENSEN	PROFESSOR	Rapporteur

*A mi sobrina, Laura Sofía Caparroso Rivera, y mi sobrino, Simón Caparroso Rivera,
las personas que más extrañé durante el transcurso de este proyecto.*

*To my niece, Laura Sofía Caparroso Rivera, and my nephew, Simón Caparroso Rivera,
the people whom I most missed during the course of this project.*

Acknowledgements

I would like to express my sincerely gratitude to all people who were very helpful during these three years of PhD experience. This experience gave me the opportunity to know different countries and cultures, to meet a lot of friendly people, and to improve my academic and research skills.

First of all, I would like to thank my thesis supervisors, Christian Prins and Murat Afsar, for giving me the opportunity to work with them. Their valuable advice enriched my knowledge and skills on optimization techniques, logistics and many other domains. Their guidance, support, feedback and confidence allowed me to successfully conclude this thesis research. I really enjoyed to work with them.

I also would like to thank the jury members, Ángel Corberán, Kenneth Sörensen, Philippe Lacomme and Olivier Péton, for accepting to judge this dissertation and being part of this important process in my life. Their suggestions and comments allow me to improve the quality of this manuscript and to find improvement opportunities and interesting perspectives for future researches.

I specially thank my wife, Natalia, for her confidence, support, sacrifice and love. I thank my parents, Julio Cesar and Beatriz, for their unconditional support to reach this degree regardless we separated during three years. Thanks to my sister, Paula, for being my confidant and a role model. I would like to thank my niece and nephew, Laura Sofía and Simón, for motivating me to be better every day. I thank my brother in law, Carlos, for his support. I would like to thank Liliana, Camilo, Lina and Estela for their understanding and confidence.

I would like to thank all the friends I met during this PhD experience. I thank to my partners at UTT, William, Andrés, Natalia, Elyn, Guillermo, Julien, Valeria and Ozlem. I will miss our academic discussions and coffee pauses. And I also thank to Alex, Karina,

Carolina, Anthony, Gustavo, Angélica, Yoly and Yoceline for their friendship. I will never forget their helping in the hardest moments, specially when we arrived to France. All of you become our family in France.

I thank LOSI Professors, Caroline Prodhon, Nacima Labadie, Andrea Santos, Faicel Hnaïen, Lionel Amodeo and Faruk Yaloui for their advice and help when I needed. I thank Veronique Banse, Marie-José Rousselet, Bernadette Andre, Pascale Denis and Isabelle Leclercq for helping me in many administrative duties.

Finally, I want to thank to Juan Guillermo Villegas for his friendship and for facilitating the contact with some professors and people in Troyes, specially the last days before the journey to France.

Juan Carlos Rivera
January, 2015.

Abstract

The vehicle routing problems (VRP) are combinatorial optimization problems in which a set of vehicles must visit a set of nodes scattered over a geographical region. These problems are known because their several applications and their complexity since they belong to the NP-Hard class of problems. VRP models can be applied to humanitarian relief operations, where relief aid such as food, water, sanitation supplies, medicine, vaccinations, medical equipment, rescue teams, rescue equipment and other, must be delivered to required sites. In humanitarian response operations, the survival rate of people is affected by the arrival time of relief aid at required sites. In this research, several vehicle routing optimization models are proposed for relief distribution after a disaster strikes, in which the sum of arrival times is minimized.

The state of the art is divided in three sections. The first one overviews the different applications of optimization techniques to humanitarian logistics. The second section focuses on vehicle routing problems for relief distribution with emphasis in service-based objective functions which better reflect humanitarian goals under urgency situations. The third section explores the main solution methods applied to vehicle routing optimization problems.

In Chapter 3 the *multitrip cumulative capacitated single-vehicle routing problem* (mt-CCSVRP) is introduced. In this case a single-vehicle with limited capacity and range is supposed to be available and it allows to perform one or more trips while the sum of arrival times is minimized. An exact method based on a shortest path algorithm with resource constraints is proposed to solve this problem. The mt-CCSVRP is formulated as a shortest path model where each trip becomes a node (trip-node) and the sites to visit are resources. The resulting network is directed and acyclic due to the special properties of the core problem. The solution method is an adaptation of Bellman-Ford algorithm to a directed and acyclic graph with resource constraints and a cumulative objective function. Several dominance rules, upper bounds and lower bounds are proposed to speed up the search procedure. Computational results are reported on instances derived from benchmark

sets for the classical CVRP and compared with the resolution of two mixed integer linear models software.

In Chapter 4 the *multitrip cumulative capacitated vehicle routing problem* (mt-CCVRP) is defined, which generalize the cumulative capacitated vehicle routing problem (CCVRP) to multiple trips and the mt-CCSVRP to multiple vehicles. In this chapter four mixed integer linear programs are proposed to solve the CCVRP. The proposed models are compared with one from literature to extend the best one for the mt-CCVRP. A non-trivial mixed integer linear program without vehicle nor trip indexes is also presented for the mt-CCVRP. Three hybrid metaheuristics are developed too: a multi-start iterated local search (MS-ILS), a memetic algorithm with population management (MA|PM) and a multi-start evolutionary local search (MS-ELS). The three metaheuristics call a variable neighborhood descent (VND) with some $O(1)$ move evaluations. A two phases *split* procedure allows MA|PM and MS-ELS to alternate between two different solution spaces. A comparison of the three metaheuristics is presented for the mt-CCVRP while MS-ILS is also compared with metaheuristics from literature for the CCVRP as special case. Moreover, the metaheuristics are able to retrieve the optimal solutions of the mixed integer linear program, which can be computed for small instances using a commercial solver.

The *cumulative capacitated vehicle routing problem with indirect deliveries* (CCVRP-ID) is introduced in Chapter 5. This problem relaxes the CCVRP by allowing unvisited sites if their requirements are provided by an auxiliary vehicle. A mixed integer linear model is presented, which extends the CCVRP model to allow indirect deliveries. An MS-ELS metaheuristic is also developed to solve the CCVRP-ID. Results compare the performances of both solution methods for the CCVRP-ID and CCVRP as special case.

Chapter 6 presents the general conclusions of this research and future research directions.

Keywords: Operations research, Combinatorial optimization, Metaheuristics, Transportation, Humanitarian assistance, Business logistics.

Résumé

Les problèmes de tournées de véhicules (VRP) sont des problèmes d'optimisation combinatoire dans lesquels un ensemble de véhicules doit visiter un ensemble de nœuds répartis sur une zone géographique. Ces problèmes sont connus en raison de ses nombreuses applications et ses complexités, car il appartient à la classe NP-difficile de problèmes. Les modèles VRP peuvent être appliqués aux opérations de secours humanitaire, où les produits vitaux tels que la nourriture, l'eau, produits d'hygiène, la médecine, les vaccinations, les équipements médicaux, les équipes de secours, l'équipement de sauvetage et d'autres, doivent être livrés aux sites nécessaires. Dans les opérations d'intervention humanitaires, le taux de survie des personnes est affecté par le moment d'arrivée des secours sur les lieux. Dans cette recherche, plusieurs modèles d'optimisation de tournées de véhicules sont proposées pour la distribution des secours après une catastrophe, dans lesquels la somme des temps d'arrivée est minimisée.

L'état de l'art est divisé en trois sections. La première décrit d'un point de vue général les différentes applications de la recherche opérationnelle à la logistique humanitaire. La deuxième section examine les modèles développés pour les problèmes de tournées des véhicules pour la distribution des secours avec un accent sur les fonctions objectives basées sur des services qui reflètent mieux les objectifs humanitaires dans des situations d'urgence. La troisième section explore les principales méthodes de résolution appliquées à des problèmes de tournées des véhicules.

Dans le Chapitre 3, le problème de tournées de véhicules cumulatif avec capacités et routes multiples où les tournées sont réalisées avec un seul véhicule (mt-CCSVRP) est introduit. Dans ce cas un seul véhicule avec capacité limitée est disponible et il permet d'effectuer un ou plusieurs voyages tandis que la somme des temps d'arrivée est minimisée. Une méthode exacte basée sur un algorithme du plus court chemin avec des contraintes de ressources est proposée pour résoudre ce problème. Le problème est formulé de telle façon que chaque voyage se transforme en un super-nœud et les sites à visiter deviennent des ressources. Ce

réseau est orienté et acyclique grâce aux propriétés du problème de base. La méthode de solution est une adaptation de l'algorithme de Bellman-Ford pour une fonction objective cumulative. Plusieurs règles de dominance, bornes supérieures et bornes inférieures sont proposées pour accélérer la procédure de recherche. Les résultats numériques sont présentés sur des instances issues de la littérature pour le CVRP classique et comparés avec deux modèles linéaires en nombres entiers.

Le problème de tournées de véhicules cumulatif avec capacités et routes multiples (mt-CCVRP) généralise le problème de tournées de véhicules cumulatif avec capacités (CCVRP) à de multiples voyages et le mt-CCSVRP à plusieurs véhicules. Quatre modèles linéaires en nombres entiers sont proposés dans le Chapitre 4 pour résoudre le CCVRP. Les modèles proposés et un modèle de la littérature sont comparés pour étendre le meilleur pour le mt-CCVRP. Ce chapitre présente également un modèle linéaire en nombres entiers pour le mt-CCVRP et trois métaheuristiques hybrides: un recherche locale itéré à démarrages multiples (MS-ILS), un algorithme mémétique avec gestion de la population (MA|PM) et un recherche locale évolutive à démarrages multiples (MS-ELS). Les trois métaheuristiques appellent une recherche locale à voisinage variable avec des quelques évaluations $O(1)$. MA|PM et MS-ELS font également usage de la procédure split et alternent entre deux espaces de solutions différentes. Une comparaison des trois métaheuristiques est présentés pour la mt-CCVRP tandis que l'MS-ILS est également comparé avec des métaheuristiques de la littérature pour le CCVRP. Ces méta-heuristiques sont capables de obtenir les solutions optimales trouvées par résoudre le problème linéaires en nombres entiers à l'aide d'un solveur sur des petites instances.

Le problème de tournées de véhicules cumulatif avec des livraisons indirectes (CCVRP-ID) est introduit dans le Chapitre 5. Ce problème est une version du CCVRP qui permet d'avoir des sites non visités si leurs besoins sont livrés par des véhicules supplémentaires. Un modèle linéaire en nombres entiers est présenté ainsi qu'une métaheuristique MS-ELS. Les résultats comparent les performances des deux méthodes de résolution pour les modèles de CCVRP et CCVRP-ID.

Le Chapitre 6 présente les conclusions générales de cette étude et des orientations des futures recherches.

Mots-clés: Recherche opérationnelle, Optimisation combinatoire, Métaheuristiques, Transport, Aide humanitaire, Logistique (organisation).

Contents

	Page
Acknowledgements	v
Abstract	vii
Résumé	ix
Contents	xiv
List of Figures	xvi
List of Tables	xviii
List of Algorithms	xix
Abbreviations	xxi
Notation	xxv
1 General Introduction	1
1.1 Operations Research	1
1.2 Transportation and Routing	3
1.3 Humanitarian Logistics	4
1.4 Disaster Response Operations	9
1.5 Performance Measurement	9
1.6 Purpose of the Thesis	12
1.7 Contributions	13
1.8 Structure of the Manuscript	15
2 State of the Art	17
2.1 Problematics in Humanitarian Logistics	18

2.1.1	Inventory Systems	18
2.1.2	Facility Location Problems	19
2.1.3	Transportation Problems	20
2.1.4	Infrastructure Recovery	21
2.1.5	Other Applications	22
2.2	Vehicle Routing Optimization Models in Disaster Response Operations . . .	23
2.2.1	Cumulative Objective Functions	24
2.2.2	Alternative Objective Functions	28
2.3	Vehicle Routing Solution Approaches	29
2.3.1	Exact Methods	30
2.3.2	Constructive Heuristics	32
2.3.3	Improvement Heuristics	33
2.3.4	Metaheuristics	34
2.3.5	Matheuristics	36
2.4	Concluding Remarks	37
3	The Multitrip Cumulative Capacitated Single-Vehicle Routing Problem: An Exact Solution Approach	39
3.1	Introduction	39
3.2	Problem Definition and Mixed Integer Linear Models	40
3.2.1	Flow-Based Model	42
3.2.2	Set Partitioning Model	46
3.3	General Description of the Algorithm	47
3.4	Auxiliary Network for the Shortest Path Problem	48
3.4.1	Generation of the Set of Trip-Nodes	49
3.4.2	Generation of the Arc-Set	51
3.5	Shortest Path Algorithm	52
3.5.1	Lower Bounds	55
3.5.2	Dominance Rules	57
3.5.3	Initial Solutions	60
3.6	Computational Experiments	60
3.6.1	Implementation and Instances	61
3.6.2	Impact of Dominance Rules on Auxiliary Network Construction . . .	61
3.6.3	Comparing the Mathematical Models and the Exact Method	62
3.6.4	Test of the Shortest Path Algorithm on Larger Instances	65
3.7	Concluding Remarks	69

4	The Multitrip Cumulative Capacitated Vehicle Routing Problem: Mathematical Formulations and Solution Algorithms	73
4.1	Introduction	73
4.2	Problem Definition and Mixed Integer Linear Models	75
4.2.1	Mathematical Formulations for the Cumulative Capacitated Vehicle Routing Problem	76
4.2.2	Mathematical Formulations for the Multitrip Cumulative Capacitated Vehicle Routing Problem	81
4.3	Multi-Start Iterative Local Search	84
4.3.1	Principle and General Structure	85
4.3.2	Solution Representation	86
4.3.3	Precomputations	87
4.3.4	Initial Solutions	90
4.3.5	Variable Neighborhood Descent	90
4.3.6	Perturbation	95
4.4	Split Procedure for the mt-CCVRP	96
4.4.1	The Split Procedure and The Route-First Cluster-Second Principle	96
4.4.2	Two-Level Split Procedure for the mt-CCVRP	97
4.5	Memetic Algorithm with Population Management	102
4.5.1	Principle and General Structure	104
4.5.2	Solution Representation	105
4.5.3	Initial Population	106
4.5.4	Selection Operator	106
4.5.5	Crossover Operator	107
4.5.6	Variable Neighborhood Descent	107
4.5.7	Concatenation	109
4.5.8	Population Updating	109
4.5.9	Mutation Operator	110
4.6	Multi-Start Evolutionary Local Search	111
4.6.1	Principle and General Structure	111
4.7	Computational Experiments	113
4.7.1	Implementation and Instances	113
4.7.2	Parameter Tuning	114
4.7.3	Comparison of Different Models for the CCVRP	116
4.7.4	Results on Larger Instances for the CCVRP	118
4.7.5	Results on Small Instances for the mt-CCVRP	122

4.7.6	Results on Larger Instances for the mt-CCVRP	124
4.7.7	Comparison between CCVRP and mt-CCVRP solutions	129
4.8	Concluding Remarks	131
5	The Cumulative Capacitated Vehicle Routing Problem with Indirect Deliveries	135
5.1	Introduction	135
5.2	Problem Definition and Mixed Integer Linear Model	137
5.3	Multistart Evolutionary Local Search	141
5.3.1	Principle and General Structure	141
5.3.2	Solution Representation	143
5.3.3	Precomputations	143
5.3.4	Heuristic for the Initial Solution	144
5.3.5	Variable Neighborhood Descent	145
5.3.6	Post-Optimization Procedure	148
5.3.7	Perturbation	149
5.4	Computational Experiments	150
5.4.1	Implementation and Instances	150
5.4.2	Parameter Tuning	150
5.4.3	Results on Small Instances	152
5.4.4	Results on Larger Instances	159
5.5	Concluding Remarks	162
6	General Conclusions and Future Research Directions	165
	Annexes	
A	Résumé Étendu de la Thèse en Français:	
	Optimisation Logistique pour les Opérations Post-Catastrophe	169
	Bibliography	191

List of Figures

	Page
1.1 Classification of disasters based on time and location.	7
1.2 Framework for disaster operations and associated facilities and flows.	8
1.3 Example of survival rate function.	10
3.1 Example of an mt-CCSVRP solution for an instance with $n = 5$, all $q_i = 10$ and $Q = 20$	42
3.2 Example of a solution on the shortest path network.	48
3.3 Impact of n and L_{max} on $ K $ for the mt-CCSVRP (linear scale).	69
3.4 Impact of n and L_{max} on $ K $ for the mt-CCSVRP (logarithmic scale).	70
3.5 Impact of n and L_{max} on the total running time for the mt-CCSVRP (linear scale).	70
3.6 Impact of n and L_{max} on the total running time for the mt-CCSVRP (logarithmic scale).	71
4.1 Example of a solution for an instance with $n = 7$, $R = 2$ and $Q = 35$	87
4.2 Example of 2-opt move within a route.	92
4.3 Example of 2-opt move within a reversal route.	92
4.4 Example of λ -interchange within a route.	93
4.5 Example of 2-opt move on two trips done by the same vehicle.	94
4.6 Example of λ -interchange on two trips done by the same vehicle.	94
4.7 Example of λ -interchange on two trips done by distinct vehicles.	95
4.8 Example of the first phase split graph with $n = 7$	98
4.9 Example of the second phase split graph.	99
4.10 Interpretation of the shortest path given by Figures 4.8 and 4.9.	99
4.11 Probability function of selection operator.	107
4.12 Crossover example.	107
4.13 Evolution of average deviation to best known solution on CMT instances.	129

5.1	Example of a CCVRP-ID solution for an instance with $n = 10$ and $R = 2$.	139
5.2	Comparison between optimal solutions for the CCVRP and the CCVRP-ID.	139
5.3	Example of node reallocation move for the CCVRP-ID.	147
5.4	Example of transportation mode change moves for the CCVRP-ID.	148

List of Tables

	Page
1.1 Commercial supply chains vs. Humanitarian relief chains.	5
3.1 Impact of dominance rules on auxiliary network generation for instances with $n = 20$ sites.	63
3.2 Results of the mt-CCSVRP solution approaches on small instances with $n = 20$ sites.	64
3.3 Results of the shortest path approach for the mt-CCSVRP for instances with $n = 25$ sites.	66
3.4 Results of the shortest path approach for the mt-CCSVRP for instances with $n = 30$ sites.	67
3.5 Results of the shortest path approach for the mt-CCSVRP for instances with $n = 40$ sites.	68
4.1 Notations from Silva <i>et al.</i> (2012) for a sequence σ of nodes.	88
4.2 Comparison of the mathematical models for small CCVRP instances.	117
4.3 Results on CMT instances (Christofides <i>et al.</i> , 1979) for the CCVRP.	120
4.4 Results on GWKC instances (Golden <i>et al.</i> , 1998) for the CCVRP.	121
4.5 Results of the mathematical models on small mt-CCVRP instances.	122
4.6 Results of metaheuristics on small mt-CCVRP instances.	123
4.7 Results on CMT instances (Christofides <i>et al.</i> , 1979) for the mt-CCVRP.	125
4.8 Results on GWKC instances (Golden <i>et al.</i> , 1998) for the mt-CCVRP.	126
4.9 Improvement of initial solution (average of 5 runs).	129
4.10 Comparison of the average arrival times of CCVRP and mt-CCVRP for CMT instances (Christofides <i>et al.</i> , 1979).	130
4.11 Comparison of the average arrival times of CCVRP and mt-CCVRP for GWKC instances (Golden <i>et al.</i> , 1998).	131
5.1 Summary of the performed tests.	151

5.2	Results of the mathematical model on small CCVRP-ID instances with $n = 15$ sites.	153
5.3	Results of the mathematical model on small CCVRP-ID instances with $n = 30$ sites.	154
5.4	Results of MS-ELS on small CCVRP-ID instances with $n = 15$ sites.	157
5.5	Results of MS-ELS on small CCVRP-ID instances with $n = 30$ sites.	158
5.6	Results of MS-ELS on CMT instances (Christofides <i>et al.</i> , 1979).	160
5.7	Results of MS-ELS on GWKC instances (Golden <i>et al.</i> , 1998).	161

List of Algorithms

	Page
Algorithm 3.1 Shortest path algorithm for the mt-CCSVRP.	54
Algorithm 4.1 MS-ILS for the mt-CCVRP.	86
Algorithm 4.2 VND for a solution S'	91
Algorithm 4.3 First level Split for the mt-CCVRP.	101
Algorithm 4.4 Second level Split for the mt-CCVRP.	102
Algorithm 4.5 Extraction of mt-CCVRP solution after Split.	103
Algorithm 4.6 MA PM for the mt-CCVRP.	105
Algorithm 4.7 MS-ELS for the mt-CCVRP.	112
Algorithm 5.1 MS-ELS for the CCVRP-ID.	142
Algorithm 5.2 Post-Optimization for a solution S	149

Abbreviations

2E-VRP:	Two-Echelon Vehicle Routing Problem.
2L-CVRP:	Two-dimensional loading capacitated vehicle routing problem.
ACO:	Ant Colony Optimization.
ALNS:	Adaptive Large Neighborhood Search.
B-RSP:	Biobjective Ring-Star Problem.
CARP:	Capacitated Arc Routing Problem.
CCVRP:	Cumulative Capacitated Vehicle Routing Problem.
CCVRP-ID:	Cumulative Capacitated Vehicle Routing Problem with Indirect Deliveries.
C_m RSP:	Capacitated m -Ring-Star Problem.
CVRP:	Capacitated Vehicle Routing Problem.
DAG:	Directed Acyclic Graph.
DMP:	Delivery Man Problem.
DS:	Decision Science.
EA:	Evolutionary Algorithm.
ELS:	Evolutionary Local Search.
ERCSP:	Elementary Resource-Constrained Shortest Path Problem.
GA:	Genetic Algorithm.
GA PM:	Genetic Algorithm with Population Management.
GRASP:	Greedy Randomized Adaptive Search Procedure.
GVRP:	General Vehicle Routing Problem.
IFRC:	International Federation of Red Cross and Red Crescent.

ILS:	Iterative Local Search.
IPS:	Integer Programming Search.
k -TRP:	k -Traveling Repairmen Problem.
LB:	Lower Bound.
LMDP:	Last mile distribution problem.
LR:	Linear Relaxation.
MA:	Memetic Algorithm.
MA PM:	Memetic Algorithm with Population Management.
MDVRP:	Multi-Depot Vehicle Routing Problem.
MILP:	Mixed-Integer Linear Programming.
MLP:	Minimum Latency Problem.
MS:	Management Science.
MS-ELS:	Multi Start Evolutionary Local Search.
MS-ILS:	Multi Start Iterative Local Search.
mt-CCSVRP:	MultiTrip Cumulative Capacitated Single-Vehicle Routing Problem.
mt-CCVRP:	MultiTrip Cumulative Capacitated Vehicle Routing Problem.
mt-VRP:	MultiTrip Vehicle Routing Problem.
OR:	Operations Research or Operational Research.
OVRP:	Open Vehicle Routing Problem.
PDVRP:	Pick up and Delivery Vehicle Routing Problem.
RCL:	Restricted Candidate List.
ROADEF:	Société Française de Recherche Opérationnelle et d'Aide à la Décision.
RSP:	Ring-Star Problem.
SA:	Simulated Annealing.
SBRP:	School Bus Routing Problem.
TDTSP:	Time-Dependent Traveling Salesman Problem.
T-PM:	Two-Phase Metaheuristic.
TRP:	Traveling Repairman Problem.

TS:	Tabu Search.
TSP:	Traveling Salesman Problem.
TTRP:	Track and Trailer Routing Problem.
UNISDR:	United Nations Office for Disaster Risk Reduction.
VND:	Variable Neighborhood Descent.
VNS:	Variable Neighborhood Search.
VRP:	Vehicle Routing Problem.
VRPTT:	Vehicle Routing Problem with Trailers and Transshipments.
VRPTW:	Vehicle Routing Problem with Time Windows.

Notation

A	Arc-Set of the graph H for the first phase of the Split procedure.
A'	Arc-Set of the graph H' for the second phase of the Split procedure.
A_i	Set of allocated sites to the visited site i in the CCVRP-ID.
B_k	Set of visited required sites in the trip k .
C_k	Cost (sum of arrival times) of the trip k , when it starts at time $t_0 = 0$.
D_k	Duration of the trip k .
E	Set of edges (i, j) in the graph G .
E'	Set of edges in the auxiliary graph G' .
e_{ij}	Cost of the arc (i, j) in the graph H for the first phase of the Split procedure.
e'_{ij}	Cost of the arc (i, j) in the graph H' for the second phase of the Split procedure.
G	Graph where related vehicle routing problems are defined.
G'	Auxiliary graph for the shortest path formulation.
H	Auxiliary graph for the first phase of the Split procedure.
H'	Auxiliary graph for the second phase of the Split procedure.
K	Set of all feasible trips.
LB_p	Lower bound of the best complete path which extends the path p .
L_k	Length of the trip k .
L_{max}	Maximum length tour.
M_k	Number of required sites in the trip k .
M_{max}	Maximum number of required sites in a trip.

$MaxChildren$	Maximum number of generated children in the MS-ELS metaheuristic.
$MaxIter$	Maximum number of iterations in the metaheuristics MS-ILS and MS-ELS.
$MaxStart$	Maximum number of starts in the metaheuristics MS-ILS and MS-ELS.
N_b	b -th Neighborhood in a Variable Neighborhood Descent (VND) metaheuristic.
n	Number of required sites.
$nchildren$	Number of children-chromosomes in the population POP_C for the MA PM metaheuristic.
$ngen$	Number of generations in the MA PM metaheuristic.
$npop$	Number of parent-chromosomes in the population POP_P for the MA PM metaheuristic.
POP_C	Population of children-chromosomes for the MA PM metaheuristic.
POP_P	Population of parent-chromosomes for the MA PM metaheuristic.
Q	Vehicle capacity.
q_i	Demand of the required site i .
R	Number of available vehicles.
R_p	Label of the path p .
$r_k^i (r_p^i)$	Binary parameter indicates if site i is visited by trip k (path p), $r_k^i = 1$ ($r_p^i = 1$), or not, $r_k^i = 0$ ($r_p^i = 0$).
S	Feasible solution.
s_i	Service time of the required site i .
s_{min}	Minimum setup time.
T	Giant-tour solution.
t_i	Arrival time at site i .
t_0^k	Starting time of the trip k .
$V = \{0, \dots, n\}$	Set depot and required sites.

$V' = V \setminus \{0\}$	Set of required sites.
w_{ij}	Traversal time of arc (i, j) .
$w'_{ij} = w_{i0} + w_{0j}$	Traversal time of replenishment arc (i, j) .
W_k	Total load of the trip k .
w_{min}	Minimum arc traversal time between two required sites.
w_{min}^0	Minimum arc traversal time incident to the depot.
X	Set of nodes in the graph H for the first phase of the Split procedure.
X'	Set of nodes in the graph H' for the second phase of the Split procedure.
Z	Objective function value (sum of arrival times) of a solution.
Z_p	Objective function value (sum of arrival time) of the multitrip or path p .
Z_k^p	Objective function value (sum of arrival time) of the trip k in the multitrip or path p .
β	Parameter used to compute the loading time of a trip.
λ	Maximum number of nodes interchanged by the neighborhood λ -interchanges.
σ	Sequence of sites.
Θ_p	Set of feasible successors for path p .
θ_k	Last visited site in the trip k .
Ω_k	Set of non-dominated partial paths ending at the trip-node k .
μ_k	Setup time (or loading time) of the trip k .
Δ	Parameter to control population diversity in the MA PM metaheuristic.
δ	Parameter to compute travel times of auxiliary vehicles in the CCVRP-ID.

Chapter 1

General Introduction

This thesis is devoted to the solution of vehicle routing problems raised from a humanitarian disaster logistics context. Some authors have noted that relief distribution has not got enough attention in the literature. Relief distribution can be challenged due to different objectives or constraints that can arise and affect the decision making.

Contrary to classical commercial logistics, where the objectives emphasize on profits, cost and return rates, humanitarian logistics focuses on saving lives and preventing suffering in the case of a humanitarian disaster, as well as reducing their impact on the economy and the environment, and returning to a state of normalcy after the disaster.

In that way, models for transport, routing and relief distribution in general can play an important role in humanitarian logistics by taking into account the specific goals raised by the emergency situation. Relief distribution is modeled by adapting classical models to relief operations in humanitarian logistics where the special characteristics present different challenges.

This chapter presents an overview on humanitarian logistics as a set of optimization problems and the importance of transport and routing optimization models and algorithms in relief distribution during the response operations following a disaster.

1.1 Operations Research

ROADEF (Société Française de Recherche Opérationnelle et d'Aide à la Décision) defines Operations Research (OR) as the discipline of scientific methods used to make better decisions (ROADEF, 2014). It streamlines, simulates and optimizes the architecture

and operation of production systems and organizations. The Operations Research Society describes operations research as a ‘real world’ discipline of applying appropriate analytical methods with a focus on improving the complex systems and processes that underpin everybody’s daily lives, Operational Research is the ‘science of better’ (The Operational Research Society, 2014). The terms management science (MS) and decision science (DS) are sometimes used as synonyms of Operations Research.

Operations Research is a scientific discipline at the frontier of mathematics, computer science and economics, which uses techniques such as optimization, simulation, game theory and heuristic methods, to find the best possible solution to quantitative problems. The methods developed in this research make use of exact and heuristic optimization techniques.

Exact optimization methods aim at finding the best between the possible. That implicitly means there exists a set of alternative solutions expressed by a set of decision variables, which can be compared by a set of performance measures called objective functions. These alternatives are subject to a set of limitations or constraints. The features of the different components define different types of problems, for instance variables can be reals, integers or binaries, and objectives and constraints can be expressed by linear, quadratic or non-linear (convex or concave) functions. Optimization techniques include linear optimization, mixed integer optimization, quadratic optimization, no linear optimization and dynamic programming. For instance, mixed integer linear programs and dynamic programming are used to model routing problems in this thesis.

On the other hand, heuristic methods are used when the exact optimization techniques mentioned before require too much computational time. They include classical heuristics rules like greedy constructive and improving heuristics, and metaheuristics such as genetic algorithms (GA), simulated annealing (SA), tabu search (TS), iterated local search (ILS) and evolutionary local search (ELS). Heuristics are frequently applied to solve combinatorial NP-Hard problems as the ones referred in this thesis. Four metaheuristic algorithms are developed in this thesis to deal with routing problems in humanitarian relief distribution.

In this thesis, humanitarian logistics and relief operations are studied. The algorithms developed and described in this manuscript are considered to support relief operations by using mathematical formulations for vehicle routing optimization models, an exact algorithm and metaheuristic methods.

1.2 Transportation and Routing

Transportation issues can be defined as optimization problems where routes to move goods between locations, from origins to destinations, must be determined while optimizing a relevant objective. Transportation is the most important economic activity of business logistics systems. Previous researches estimate that in commercial applications, around one to two thirds of the logistics costs of companies are spent on transportation. In disaster logistics, the structure of the distribution network directly affects relief costs throughout the relief chain as well as the response time.

In the capacitated vehicle routing problem (CVRP), a fleet of vehicles is available to pick up or deliver people or goods from a set of depots to a set of required sites. In the most classical constraints, each site must be visited exactly once, vehicle capacity should be respected, vehicles must start at and return back to depot and route length can not exceed a range. Additional features have led to many variants such as Open VRP (OVRP), Multi-Depot VRP (MDVRP), Pick up and delivery VRP (PDVRP), General VRP (GVRP), Green VRP, Multitrip VRP (mtVRP), and other.

VRPs are known as NP-hard combinatorial optimization problems and no polynomial exact methods are known to solve them. The number of variations, their applications, their complexity and their economic impact give to vehicle routing problems an important practical and theoretical interest.

In disaster situations, it makes sense to think that each particular house or building within the affected region could require relief or humanitarian aid, thus becoming a potential demand point. In a severe crisis that affects a large area, it becomes impractical to consider the fine details in designing and deploying humanitarian aid distribution networks. To cope with these difficulties, demand points are aggregated into demand zones and needed products are grouped into generic humanitarian packages, which can include food, water, sanitation supplies, medicine, medical equipment, rescue equipment, etc. Due to the urgency and with the objective of saving lives and avoid suffering, delivery of the humanitarian aids from suppliers to demand points must be done in the fastest way. Depending on the structure of the problem, the constraints and the performance measures, different mathematical formulations are found in the literature but, compared to industrial logistics, an important gap remains. Indeed, despite all variations presented in the literature, routing applied to disaster response operations have not taken enough attention.

1.3 Humanitarian Logistics

In the last decade, several disasters around the world have caused millions of victims and massive destruction of infrastructure and environment. The effects of large-scale disasters are increasingly devastating. Responding to these emergencies in the most effective manner to reduce the loss of life and suffering becomes more and more important.

Examples of recent disasters are hurricane Katrina in the United States in 2005, hurricane in Haiti in 2008, earthquake in Haiti in 2010, earthquake in Chile in 2010, floods and landslides in China in 2010, earthquake and tsunami in Indian Ocean in 2004 and in Japan in 2011. Between 2003 and 2012, according to the International Federation of Red Cross and Red Crescent Societies (IFRC, 2013), 670 disasters occurred on average every year, which have caused 115 thousand deaths and 216 millions of affected people. Seismic events have induced the greatest number of deaths, on average 67 818 per year, which represents 63.66% of the people reported killed in disasters. On the other hand, flood events have affected the largest number of people, on average 108 million per year, which is equivalent to 49.89% of the total number of affected persons by disasters. Information about disaster is updated by IFRC in its web page. IFRC reports also classify disasters by continents, countries and other different disaster classifications.

The enormous scale of these disasters, which can be checked in databases like EM-DAT, AirDisaster.com, Natural Disaster Reference Database, The British Association for Immediate Care, Social Studies Network for Disaster Prevention in Latin America, Disaster Resource Guide or DisasterRelief.org, has brought attention to the need for methodology and technology for effectively managing relief supply chains. Although a great deal of research and technology is available for commercial supply chains, Beamon (2004) notes that the challenges associated with managing humanitarian relief chains following large-scale emergencies are often quite different. Table 1.1, taken from Beamon (2004), presents the main differences between commercial supply chains and humanitarian relief chains. Thomas & Kopczak (2005), van Wassenhove (2006) and Apte (2009) also compare and contrast humanitarian relief chains and commercial supply chains.

Altay & Green III (2006) define disaster operations as the set of activities that are performed before, during, and after a disaster with the goal of preventing loss of human life, reducing its impact on the economy, and returning to a state of normalcy. Other definitions are focused on the type of activities, for instance Blecken (2010) define humanitarian logistics

Table 1.1: Commercial supply chains vs. Humanitarian relief chains.

Source: Beamon (2004).

	Commercial Supply Chain	Humanitarian Relief Chain
Demand Pattern	Relatively stable, predictable demand patterns. Demands occur from fixed locations in set quantities.	Demand is generated from random events that are unpredictable in terms of timing, locations, type, and size. Demand requirements are estimated after they are needed, based on an assessment of disaster characteristics.
Lead time	Lead time determined by the supplier - manufacturer - distribution center - retailer chain.	Approximately zero lead times requirements (zero time between the occurrence of the demand and the need for the demand), but the actual leadtime is still determined by the chain of material flow.
Distribution network configuration	Well-defined methods for determining the number and locations of distribution centers.	Challenging due to the nature of the unknowns (locations, type and size of events, politics, and culture), and "last mile" considerations.
Inventory control	Utilizes well-defined methods for determining inventory levels based on lead time, demand and target customer service levels.	Inventory control is often unreliable, incomplete or non-existent.
Information system	Generally well-defined, using advanced technology.	Information is often unreliable, incomplete or non-existent.
Strategic goals	Typically: to produce high quality products at low cost to maximize profitability and achieve high customer satisfaction.	Minimize loss of life and alleviate suffering (Thomas, 2002).
Performance measurement system	Traditionally: focused on resource performance measures, such as maximizing profit or minimizing costs.	Primary focus on output performance measures, such as the time required to respond to a disaster (Thomas, 2002) or ability to meet the needs of the disaster (customer satisfaction).
What is "demand"?	Products.	Supplies and people.

as the process of planning, implementing and controlling the efficient, cost-effective flow and storage of goods, materials and equipment as well as related information, from points of origin to points of consumption for the purpose of meeting the beneficiary's requirements. Although there is not a commonly accepted definition in the literature, Tomasini & van Wassenhove (2009) remark that there is no doubt about the three main principles surrounding the humanitarian issues: humanity, neutrality and equity. The last ones are described in Hellingrath & Widera (2011) as follows: first, humanity figures as a main objective of humanitarian organizations and its actions. Second, neutrality can be understood as a restricted room for maneuver. Third, equity obliges humanitarian actors to ensure equal treatment of beneficiary peers.

To complement the above definition, IFRC (2011) defines a disaster as a sudden, calamitous event that seriously disrupts the functioning of a community or society and causes human, material, and economic or environmental losses that exceed the ability of a community or society to cope with using its own resources. The combination of hazards, vulnerability and inability to reduce the potential negative consequences of a risk is a classical cause of disasters. According to Apte (2009), a disaster is not necessarily a sudden event, but suddenness is one of the factors that impacts the capacity to prevent the disaster consequences.

IFRC classifies disaster types as natural hazards, which can be geophysical, hydrological, climatological or biological, and technological or man-made hazards. It provides definitions for each type of disaster while Green III & McGinnis (2002) discuss a broad classification of these events by causation.

Apte (2009) presents a different classification based on time and location. The author argues that these factors play important roles to determine the level of difficulty of humanitarian logistics operations as depicted in Figure 1.1. For instance, preparation activities can play an important role in the disasters classified in the first quadrant in Figure 1.1. In the second quadrant, preparation can help but pre-positioning becomes challenging. Response operations are more and more present along second, third and fourth quadrants.

Altay & Green III (2006) also define the emergency response as the global response to catastrophic and disaster events, without considering the daily responses of ambulance, police, or fire departments to routine emergency calls. Indeed, there exists a general agreement about a clear demarcation between what are termed routine emergencies

(Hoetmer, 1991), also called everyday emergencies (Fischer III, 2003), and more serious emergencies induced by disasters and catastrophes.

A routine event is typically managed with the resources of a single governmental agency, or partial resources from several, using standard procedures, and with minimal dislocation. Operationally, the transition to a higher category of emergency occurs when resources become stressed, when non-standard procedures must be implemented to save life or when special authorities must be invoked to manage the event (Landesman, 1996, 2001; Auf-Der-Heide, 1989). Readers interested in research on daily emergencies are referred to Chaiken & Larson (1972), Swersey (1994) and Brandeau *et al.* (2004).

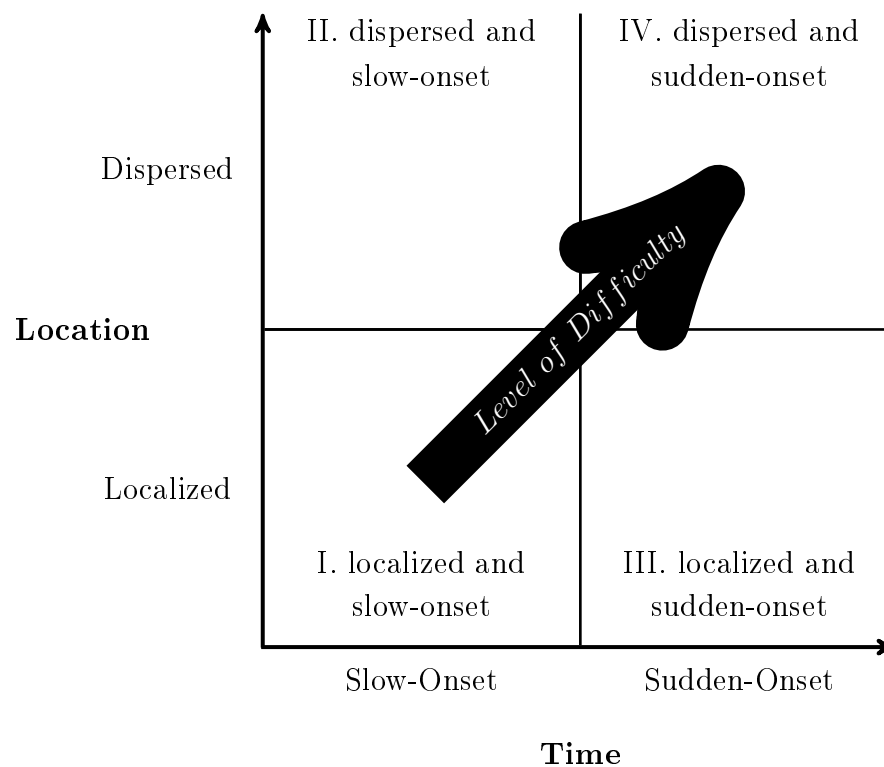


Figure 1.1: Classification of disaster based on time and location.

Source: Apte (2009).

Some authors (Thomas & Kopczak, 2005; Tomasini & van Wassenhove, 2009) have shown that humanitarian logistics significantly contributes to the efficiency and effectiveness of humanitarian operations, where logistics costs are between 40% and 60%, but can sum up to 80% of the total spent (Blecken, 2010).

Altay & Green III (2006) classify humanitarian logistics operations in four phases: mitigation, preparedness, response and recovery. Other classifications have been presented by the United Nations Office for Disaster Risk Reduction (UNISDR, 2004). First two concern the pre-crisis. Mitigation is the application of measures that will either prevent the onset of disasters or reduce their impacts, while preparedness activities train the community to react when a disaster occurs. When the crisis occurs, the post-crisis phases take place. Response is the deployment of resources and emergency procedures to preserve life, property, the environment, and the social, economic, and political structure of the community. Recovery involves the actions in order to stabilize the community and to restore some semblance of normalcy after the immediate impact of the disaster.

While a large body of the recent research is focused on the preparation, planning of disaster evacuation and alleviation, little attention has been put on humanitarian response problems during and after disasters (Hamedi *et al.*, 2012).

Figure 1.2, taken from Caunhye *et al.* (2012), summarizes the relationship between the main emergency logistics activities, their corresponding phase (pre-disaster and post-disaster) and their associated facilities and flows. The arcs indicate the activities as well as the directions of the main flows they provoke. The edges (undirected arcs) do not indicate flows but rather express that there is a relationship between the two entities linked.

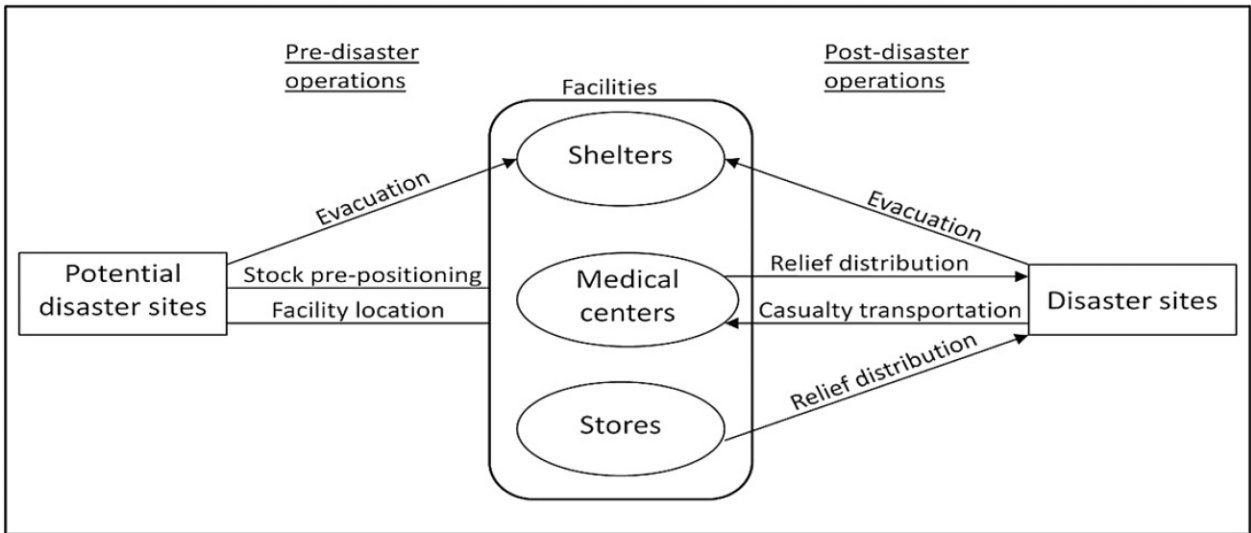


Figure 1.2: Framework for disaster operations and associated facilities and flows.

Source: Caunhye *et al.* (2012).

1.4 Disaster Response Operations

In the sequel, disaster response operations represent the set of the short-term activities with the purpose of mitigating the urgent needs of a population after a disaster. Terms like humanitarian relief are synonymous but daily emergencies are not considered as disasters. Some authors like Apte (2009) make a distinction between disaster response and humanitarian relief where different emergency levels are considered.

The distinguishing factor in disaster response operations, in contrast with pre-disaster and long-term post-disaster operations, is that suffering and vulnerability can increase rapidly in the absence of timely aid.

The logistics activities in this category involve relief distribution, evacuation of people, treatment of disaster casualties, transfer of rescue equipment and personnel, assignment and coordination of rescue teams and volunteers. According to Yi & Özdamar (2007), logistics support and evacuation are two major activities in disaster response.

An important logistic issue in disaster response operations is to determine the transportation routes for first aids, supplies, casualties, rescue personnel and equipment between supply points and a large number of destination nodes, geographically scattered over the disaster region. In this sense, vehicle routing models can be considered for delivery in disaster context by using service-based objective functions to reflect the different priorities for delivering humanitarian aid (Campbell *et al.*, 2008). On the other hand, the arrival time of relief supplies at the affected communities clearly impacts the survival rate of citizens and the amount of suffering. Figure 1.3 shows an example of a survival rate function. The next section describes different system performance measurements used in VRPs, including humanitarian logistics as well as commercial applications.

1.5 Performance Measurement

The goal of performance measurement is to establish the relationship between decision variables and performance criteria or objectives. It is clear that different application contexts can have different objectives. This difference is even stronger between commercial applications and humanitarian interventions. For instance, commercial supply chains use cost-based objectives or a combination between cost and service level, while humanitarian logistics chains can vary their objectives depending on the disaster impact, phase, and

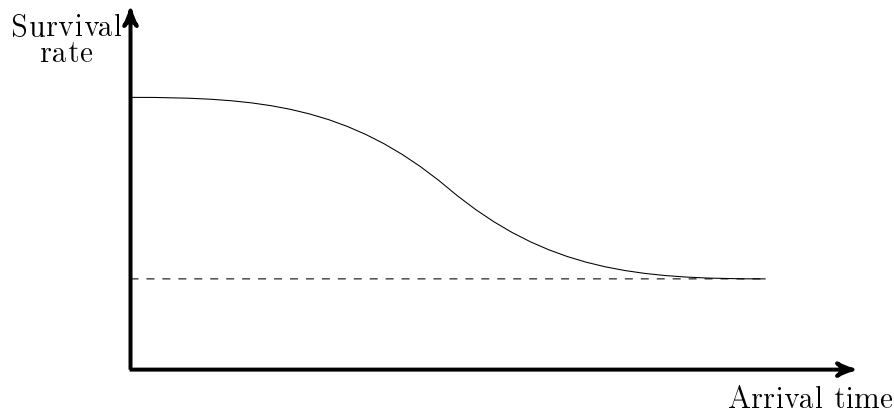


Figure 1.3: Example of survival rate function.

external circumstances. When activities are performed to mitigate risks and preparation, cost-based objectives can be considered but covered population or risks are also important. During response operations, monetary measures are dominated by response time which is more related to the amount of human lives in danger or the amount of suffering. The lack of appropriate performance indicators appears as an important issue in humanitarian logistics (Beamon, 2004).

Humanitarian logistics, as well as other service-based applications, pursues alternative objectives, where financial and economical measures are sacrificed for more strategic and important goals. Economical criteria can be handled by constraints, for instance a maximal budget. Multiobjective approaches is another alternative.

The following examples illustrate different routing performance measures found in the literature:

- Total cost minimization: This measure represents the most common objective in industrial logistics and vehicle routing applications. Total cost can also be translated in a total duration or a total length because these quantities are frequently related. Stochastic applications use the total expected cost, while in robust optimization the worst cost is minimized.
- Minimization of the number of vehicles: The number of vehicles can be consider as a fixed cost of the distribution system. Although some applications consider it as a constraint, many of them include it as an objective.
- Total profit maximization: This variant becomes useful when is not feasible to visit all

customers and it is necessary to choose the ones who contribute with greater benefits for the system.

- Longest route minimization: This objective consists in minimizing the tour length of the vehicle traveling the longest distance. This problem is often of interest when minimization of the time taken to visit all points is more important than the total distance traveled. The best solutions make use of all available vehicles in an attempt to reduce the distance traveled by those vehicles with the largest tours, leading to a more equitable sharing of loads between the vehicles.
- Sum of (weighted) arrival time minimization: Contrary to the previous measures, this is a service-based objective which gives priority to customer needs instead of company costs. It has been recently used in humanitarian distribution due to it favours rapid response. Weighted cases have been considered too, as cost function which take in account the fuel consumption.
- Maximal arrival time minimization: This objective, which has been also considered in humanitarian logistics operations, aims to reduce the worst service level received by required sites or customers. Note that this objective is very close to longest route minimization, but without considering the last arc in the route.
- Covered sites maximization: Different cases are found with this criteria. The most common version is to avoid visiting all customers to visit closer customers within a bounded distance. This case is useful when a second routing level can be used. Another frequent version is presented when it is not possible to visit all customers: as many as possible must be visited. Sometimes, the goal is to deliver as many goods as possible when depot capacity is lower than total demand. In that case, equity constraints are useful to deliver proportional quantities to every required site.
- Minimization of a sum of several objectives: Multiobjective cases can consider the sum of weighted objectives to simply the search of good solutions. Other cases consider penalty functions which penalize constraint violations, for instance the time windows constraints.

Although it is not the rule, the first three objectives are more concerned with commercial logistics while second three have emphasis on humanitarian logistics.

1.6 Purpose of the Thesis

While many papers about relief distribution have put attention on features like uncertainty on demands or travel times, the special emphasis of this research is the timely response.

More precisely, the aim of this thesis is to study a new family of capacitated vehicle routing problems, raised by the relief aid deliveries in the response phase of humanitarian logistics: the Cumulative Capacitated Vehicle Routing Problems (CCVRP). Although it has more applications, it is used here to model relief distribution after a humanitarian disaster where urgency is the most critical feature. As mentioned before, arrival time to required sites has a great impact on survival rate of citizens and suffering alleviation.

The CCVRP is a capacitated vehicle routing problem where a limited fleet is based at the depot, goods must be delivered to a set of required sites and the objective function becomes the sum of arrival times at required sites.

In disaster response operations the CCVRP models the delivery of relief aid (vaccinations, rescue teams, or equipment for instance) where early arrivals can be translated in the increase of the survival rate and alleviation of suffering. The objective function, the sum of the arrival times, can be interpreted as the maximization of the survival rate.

Three different variations to the original CCVRP, which was first introduced by Ngueveu *et al.* (2010), are considered and studied: a) the multitrip cumulative capacitated single-vehicle routing problem (mt-CCSVRP), b) the multitrip cumulative capacitated vehicle routing problem (mt-CCVRP) and c) the cumulative capacitated vehicle routing problem with indirect deliveries (CCVRP-ID). The two first problems consider the case where total vehicle capacity is not enough to deliver the required needs with one tour per vehicle: multiple trips become necessary. The third variant considers the case where the vehicle fleet delivers all required needs by single trips, but some sites can be left unvisited to be supplied directly in a second distribution step from a visited site.

Solution methods for each problem version differ because their differences allow to take advantage of their structure to improve solutions or accelerate the search procedure. Specific algorithms for the CCVRP are not developed, but it can be solved as a special case of more general versions by some of the proposed algorithms. Mathematical formulations

have been proposed for the CCVRP to compare different modeling strategies and variable definitions.

The mt-CCSVRP, studied in Chapter 3, is the case where only one vehicle is available to visit the required sites (for instance a helicopter), but multiple sorties are allowed. An additional constraint, which is not present in the CCVRP, avoids to perform routes which exceed a length limit. Although the interest of this problem version is more theoretical than practical, its mathematical properties allows to apply an exact method for its resolution.

In the mt-CCVRP, studied in Chapter 4, multiple trips are allowed for a limited fleet of vehicles. Three metaheuristic algorithms have been developed: a Multi-Start Iterative Local Search (MS-ILS) (Section 4.3), a Memetic Algorithm with Population Management (MA|PM) (Section 4.5), and a Multi-Start Evolutionary Local Search (MS-ELS) (Section 4.6). All developed methods use a Variable Neighborhood Descent (VND) procedure as improving phase. Elaborated moves are developed due to the special features of the objective function (the minimization of the sum of arrival times), and the multitrip consideration. The main difference between MS-ILS and MS-ELS in the depicted implementations is the presence of a Split procedure in the MS-ELS algorithm, also used by the MA|PM and described in Section 4.4.

Finally, the CCVRP-ID is a relaxation of the CCVRP where the demand of each site can be delivered by a vehicle initially based at the depot, or dispatched from an assigned nearby site by an auxiliary transportation mode. The MS-ILS and MS-ELS metaheuristics from the mt-CCVRP are adapted to solve this problem and manage the new considered features. The aim of this new proposed problem is to allow earlier arrivals to last sites in a route while intermediate sites can be dispatched from another intermediate site to reduce the average arrival time.

This research takes place in a wider program on disaster logistics funded by Troyes University of Technology (UTT).

1.7 Contributions

In the state of the art (Chapter 2) a survey about optimization techniques applied to humanitarian logistics is presented. The papers are classified by applications: inventory systems, infrastructure recovery, location, transportation and resource allocation. More

detailed descriptions are presented for vehicle routing applications, where not only humanitarian logistics applications are considered, but also applications where the special objective functions considered could be meaningful. A brief state of the art about VRP solution techniques is also presented.

In Chapters 3, 4 and 5 three new problems are defined. These problems are raised from relief operations. Nevertheless, they have other applications in logistics and scheduling, for example in routing for maintenance operations and green logistics.

The first proposed problem is the *multitrip cumulative capacitated single-vehicle routing problem*, abbreviated as mt-CCSVRP and defined in Chapter 3. In this problem, like in the CCVRP, there exists a set of required sites with known demands, a capacitated vehicle is based at the depot and the objective function becomes the sum of arrival times. New features allow to perform multiple sorties by the available vehicle with a limited range. The range is limited because of the fuel consumption.

To solve the mt-CCSVRP three exact approaches are proposed: two integer linear mathematical models and an exact algorithm based on a resource-constrained shortest path formulation. The first mathematical model is a flow-based model where variables are indexed by sites and no vehicle nor trip indexes are needed. The second mathematical model is a set partitioning formulation where columns correspond to trips and required sites becomes resources. The shortest path algorithm is composed by two sequential phases. In the first phase, three dominance rules are defined to speed up the network construction. In the second phase, five lower bounds and five dominance rules are developed to find the shortest path. A comparison of the three approaches is performed which shows that the resource-constrained shortest path algorithm outperforms others.

The second problem, the *multitrip cumulative capacitated vehicle routing problem* (mt-CCVRP), is defined in Chapter 4. In this problem, which extends the CCVRP to multiple trips, a limited fleet of vehicles must serve to a set of required sites. No time limit constraints are considered. An integer linear mathematical model and three metaheuristics are developed to solve the mt-CCVRP.

Four alternative mathematical models for the CCVRP are also proposed in Chapter 4. A comparison between them and the model proposed by Ngueveu *et al.* (2010) is presented. This comparison allows to evaluate the different model approaches and their

possible extensions to the multitrip version of the problem. The proposed mathematical formulation for the mt-CCVRP is a flow-based model similar to the proposed for the mt-CCSVRP.

The three metaheuristic algorithms are: a) a Multi-Start Iterated Local Search (MS-ILS), b) a Memetic Algorithm with Population Management (MA|PM), and c) a Multi-Start Evolutionary Local Search (MS-ELS). All of them use a Variable Neighborhood Descent (VND) algorithm as improving procedure, where moves become more involved than in classical problems because of the different objective function and their multitrip feature. Precomputations are designed to accelerate the VND moves.

A splitting procedure is developed and applied in the MA|PM and MS-ELS metaheuristics. This procedure, adapted from Prins (2004), optimally translates a sequence of sites into a solution for the mt-CCVRP. Two levels are used: the first level defines the set of sites to be visited by each vehicle and the second one defines the trips of each multitrip.

The *cumulative capacitated vehicle routing problem with indirect deliveries* (CCVRP-ID) is defined in Chapter 5. This problem relaxes the CCVRP in the sense that trips do not necessary visit all required sites, but the unvisited ones must collect their demands from a neighboring visited site, via an auxiliary transportation mode. A limited fleet is available.

An mixed integer linear formulation and a Multi-Start Evolutionary Local Search (MS-ELS) are developed to handled this problem. New improving heuristics are defined and their moves have a complexity of $O(1)$. New precomputations are also defined to speed up the new proposed moves. This problem shows that when two transportation levels are used the average arrival time to required sites can decrease.

The different models and solution methods proposed in this thesis show that the sum of arrival times objective raises different implementation challenges, as well as new opportunities of applications in humanitarian logistics context.

1.8 Structure of the Manuscript

The manuscript is structured as follows:

Chapter 2 reviews the problems related to cumulative routing problems, and their solution

methods. This chapter is divided in three sections: a) problematics in humanitarian logistics, b) vehicle routing optimization models in disaster response operations, and c) vehicle routing solution approaches. The section problematics in humanitarian logistics refers to several operation research developments applied to humanitarian logistics problems. Vehicle routing optimization in disaster response operations describes related routing problems and makes emphasis on problems that use cumulative objective functions or alternative objectives addressed to humanitarian logistics. In vehicle routing solution approaches, the main solution strategies are described which are closely related to the solution approaches developed in this research.

In Chapter 3 the multitrip cumulative capacitated single-vehicle routing problem is defined. Two mathematical models are proposed, and an exact approach based on a resource-constrained shortest path formulation and a dynamic programming solution is developed. The computational testing describes and compares the results of the three approaches.

Chapter 4 introduces the multitrip cumulative capacitated vehicle routing problem which generalizes the single-vehicle case. Mathematical models are proposed for the mt-CCVRP as well as the CCVRP. Three metaheuristic algorithms are presented, which make use of VND metaheuristics and splitting procedure. Numerical evaluations are performed to compare the different models and metaheuristic approaches.

The cumulative capacitated vehicle routing problem with indirect deliveries is described in Chapter 5. A mathematical model and a metaheuristic algorithm are developed. Numerical tests allow to compare the performance of the proposed methods.

Finally, general conclusions and future research directions close the thesis in Chapter 6.

Chapter 2

State of the Art

In this chapter, a literature review is presented and classified in three sections: a) Problematics in humanitarian logistics, b) Vehicle routing optimization models in disaster response operations, and c) Vehicle routing solution approaches. As many kinds of models are found in humanitarian logistics, this chapter is focused on optimization and metaheuristic techniques which are the main tools used in the contributions of this thesis. Simulation, decision analysis, game theory and other techniques are not included in this research.

The first section, Problematics in humanitarian logistics, explores a general point of view about optimization models applied to humanitarian logistics. Models are grouped based on the field of application as follows: inventory systems, infrastructure recovery, facility location problems, transportation problems and other applications which include for instance scheduling and personnel assignment.

The section on vehicle routing optimization models in disaster response operations reviews the papers which present applications of vehicle routing problems in disaster response operations or other domains where similar objective functions are pertinent. These applications are grouped according to their main characteristics, such as single or multiple vehicles, objective function, special constraints, etc.

Finally, the last section on vehicle routing solution approaches section describes the main strategies to solve vehicle routing problems, which mainly include exact methods, greedy heuristics, improvement heuristics, metaheuristics and matheuristics. The methods developed in this thesis are based on some of the algorithmic structures from the literature review. These adaptations are not trivial due to the specific characteristics of the objective function and other features of the addressed problems.

2.1 Problematics in Humanitarian Logistics

A recent trend is to apply operations research techniques to facilitate logistic operations in humanitarian disaster. Examples include, but are not limited to, inventory systems, facility location, infrastructure recovery, evacuation, casualties transport, resource (volunteers or rescue personnel and equipment) allocation, and vehicle routing. Short-notice evacuation, facility location, and stock pre-positioning are drafted as the main pre-disaster operations, while relief distribution and casualty transportation are categorized as the main post-disaster operations (Caunhye *et al.*, 2012). Several models have been proposed to address the humanitarian disaster problems which are found along the four phases of disaster operations defined by Altay & Green III (2006) and described in Section 1.3: mitigation, preparedness, response and recovery.

Caunhye *et al.* (2012) remark that the first optimization models in emergency logistics were developed in the late 1970s following several maritime disasters in the late 1960s and 1970s, such as the Torrey Canyon shipwreck off the coast of England in 1967 and that of Argo Merchant near Massachusetts in 1976. More recently, optimization models focus on oil spills or maritime disasters in general (Charnes *et al.*, 1976, 1979; Belardo *et al.*, 1984; Psaraftis & Ziogas, 1985; Psaraftis *et al.*, 1986). Kembal-Cook & Stephenson (1984) first addressed the need for logistics management in relief operation in Somalia, while Ardekani & Hobeika (1988) found similar logistics needs in relief operations for the 1985 Mexico City earthquake. From the 1980s up to now, research has also explored other large-scale emergencies such as hurricanes, floods, earthquakes, famine and nuclear accidents.

The following are the main and more recent optimization applications in humanitarian logistics operations.

2.1.1 Inventory Systems

Inventory systems are related to decisions like review strategy, reorder level or order quantity, and they can take in account the probability of stock-out and expected costs.

Beamon & Kotleba (2006a) indicate that relief inventory modelling is specially challenging compared to the commercial cases due to the irregular demand. They propose a model to determine order quantities and reorder points for pre-positioned stocks for an emergency relief response in South Sudan by supposing an uniform demand distribution. The same application case is analyzed by Beamon & Kotleba (2006b) with a multi-supplier

inventory model and unique demand pattern.

Das & Hanaoka (2014) develop a method which considers uniform distribution in demands and lead-times while Davis *et al.* (2013) focus on the coordination of internal and external supplies in the preparedness domain.

As it can be seen, inventory systems modelling in humanitarian logistics contexts is focused on the preparedness disaster phase. Richer models consider inventory-location or inventory-distribution problems.

2.1.2 Facility Location Problems

Facility location problems aim at determining the number, the position and the mission of required humanitarian aid distribution centers. For instance, Marianov & ReVelle (1995) provide an overview of optimization models, both deterministic and stochastic as well as descriptive. Bell *et al.* (2013) develop a p-median model to locate depots for humanitarian relief chains in a network with damaged links after an earthquake.

Ambulance allocation is addressed by Gong & Batta (2007). They consider the allocation and reallocation of ambulances to casualty clusters which need to be treated. First allocation is done based on makespan minimization of each cluster, and reallocations consider flow time minimization. Surveys on ambulance location are presented by Brotcorne *et al.* (2003) and Li *et al.* (2011).

Lin *et al.* (2012) develop a location model for temporary depots. They propose a heuristic procedure which locates depots and allocates demand points in a first phase, and then explores the operational cost-based logistics performance.

Furuta & Tanaka (2010) propose a maximal covering location problem to locate helipads for helicopters with the objective of maximizing the amount of covered demand.

According to Schilling *et al.* (1993) and Farahani *et al.* (2012), models with coverage-type objectives are especially used when response time is the primary performance criterion.

Most facility location models in emergency logistics combine the process of location (building new facilities or choosing among existing ones) with inventory, evacuation, or relief distribution decisions. For instance, Abounacer *et al.* (2014) present an ϵ -constraint approach

for a location-transportation problem where three objectives are considered: minimize the sum of all transportation durations, minimize the number of agents needed to operate humanitarian aid distribution centers and minimize the total uncovered demand.

Balcik & Beamon (2008) integrate facility location with inventory decisions in a maximal covering model. They also consider probabilistic scenarios, multiple item types, budgetary constraints and capacity restrictions.

An emergency location survey is presented by Başar *et al.* (2012). Prodhon & Prins (2014) provide a survey on location-routing problems which includes a section about problems raised from emergency situations.

2.1.3 Transportation Problems

Most developments in transportation science for disaster logistics can be partitioned in shortest path, network flow and routing models.

One example of shortest path problem is given in Campos *et al.* (2012). They present an algorithm which defines two independent paths from the disaster area to one destination point of an evacuation plan. This algorithm takes into account the capacity of transportation routes and the travel time to select the best set of routes.

A shortest path model to find emergency paths by using two criteria is proposed by Mohaymany *et al.* (2003). The criteria are the minimization of the travel time and the maximization of the rescue teams capabilities to cover the affected population.

Özdamar *et al.* (2004) describe a network flow model that coordinates deliveries of supplies from different depots in the context of relief operations. Their model addresses a dynamic time-dependent transportation problem which needs to be solved repetitively during ongoing aid delivery. The authors propose a mathematical model which integrates the multi-commodity network flow problem and the vehicle routing problem to minimize the sum of unsatisfied demand throughout the planning horizon. This model is applied to small instances up to 9 sites.

Lin *et al.* (2011) compare a genetic algorithm and a decomposition approach in a routing problem in which multiple items, multiple vehicles, multiple periods, soft time windows and split deliveries are considered while three objectives are weighted: minimization of the

unsatisfied demand, the total travel time and the difference in satisfaction rates among sites.

A multimodal transportation routing approach inspired by an immune affinity model is presented by Hu (2011). This model considers a cost-based objective.

Jotshi *et al.* (2009) develop a model of dispatching and routing of emergency vehicles with the support of data fusion. This model collects and interprets information from multiple sources, assign a vehicle to a demand (casualty) point, and determines the path to pickup and deliver casualties.

Lagaros & Karlaftis (2011) propose a vehicle routing problem concerning infrastructure inspections. Its purpose is to find routes and crews to inspect infrastructures in urban areas after a disaster. The solution approach considers a first step where structural blocks to be inspected are grouped in clusters and a second one where clusters are scheduled to be inspected by crews. Five heuristics are developed and compared to solve this problem.

A hybrid fuzzy clustering-optimization approach is proposed by Sheu (2007) for urgent relief demand distribution. This model considers two objective functions: total cost minimization and unsatisfied demand minimization, while fuzzy techniques are used to group required sites in clusters classified by their urgency.

Bakuli & Smith (1996) propose a state-dependent queuing network model for emergency evacuations. Similarly, Stepanov & Smith (2009) deals a multi-objective evacuation routing problem via state-dependent queuing models. Their model include blocking probabilities, clearance time and total travelled distance as objectives.

Vehicle routing applications are presented in more details in Section 2.2.

2.1.4 Infrastructure Recovery

Infrastructure recovery is a topic especially important in the recovery operations phase where, after an earthquake for example, roads can be damaged and some nodes on the network can become inaccessible or costly to reach. In this kind of situations, disruption of the communication and transport infrastructure deserves particular attention because it can obstruct the post-disaster operations (Maya & Sörensen, 2011). A recovery planning must be performed to upgrade the network by making damaged roads operational or upgrading their capacity.

Bryson *et al.* (2002) propose a mathematical model based on a knapsack formulation to optimize the potential recovery measure for the set of selected recovery plans.

Maya & Sörensen (2011) develop a solution approach based on GRASP and VNS metaheuristics to determine how the scarce resources must be assigned in order to optimize the network accessibility after a disaster, by respecting budget and time limitations. This approach is tested on data based on the case of Haiti natural disaster in 2008.

Aksu & Özdamar (2014) present a dynamic path based mathematical model to restore blocked roads with the goal of opening access paths for all locations as early as possible.

2.1.5 Other Applications

This subsection gathers miscellaneous applications which do not justify dedicated sections.

Allocation models can be considered, in disaster relief context, to allocate airtankers in forest-fire management, rescue teams to clusters, or volunteers to different activities.

Islam (1998) addresses the daily airtanker deployment problem for forest-fire management. Their model decides how many airtankers must be used every day and where they should be deployed dynamically.

A resource-constrained project scheduling problem is solved by Yan *et al.* (2009) for scheduling and controlling rescue teams in a maritime disaster. Five heuristics are analyzed for this problem.

A volunteer assignment model in humanitarian organizations is proposed by Falasca & Zobel (2012), where volunteers abilities and preferences must be considered.

Two resource allocation problems are addressed by Chou *et al.* (2014) using evolutionary algorithms. The authors test the algorithm on a refuge site staff allocation planning and a relief supply distribution planning.

Pettit & Beresford (2005) model the relations between participating bodies including military and non-military organizations.

Scott & Rogova (2004) describe how to manage and interpret multiple information sources to conduct disaster relief management with data fusion.

Additional information about operations research applied to disaster relief operations can be found in different surveys on the literature: Altay & Green III (2006) present a literature review where papers on disaster operations management are classified and future research directions are proposed. They classify papers according to different features: the disaster phases concerned (before, during or after disaster operations), research methodology, research contribution, disaster type and problem scenario. A study about emergency evacuations is conducted by Abdelgawad & Abdulhai (2009). Caunhye *et al.* (2012) present a literature review about optimization models focused on facility location, relief distribution and casualty transportation with an analysis about future research directions. Galindo & Batta (2013) follow the same classification proposed in Altay & Green III (2006), complement their literature review covering the period between 2005 and 2010, and propose future research directions.

2.2 Vehicle Routing Optimization Models in Disaster Response Operations

The CVRP involves designing the delivery routes for a set of capacitated vehicles which service a set of geographically-scattered demand points in order to minimize a cost function while taking into account a set of constraints.

Several papers on routing problems applied to disaster response operations contexts have been published, which is a proof of the importance, variety of case studies and complexity of this applications. In this section the vehicle routing models applied to disaster logistics are studied. In Section 2.2.1 special attention is put on models using cumulative objective functions, while Section 2.2.2 is focused on other alternative optimization criteria.

Barbarosoğlu *et al.* (2002), for example, introduce a hierarchical decision support methodology to use helicopters efficiently. They consider two decision levels: in the first level, decisions about fleet size, assignment of pilots to helicopters and number of tours to be performed by each helicopter are elaborated, while the second level is devoted to routing, re-fueling and deliveries, transships and load/unload decisions.

De Angelis *et al.* (2007) consider the airplane routing and scheduling problem to transport food after an emergency in Angola. They investigate the case where aircrafts can make several full-load departs each day to visit required sites and the objective is to maximize the satisfied demand. A mathematical model is built and experiments are run on real data.

The last mile distribution problem (LMDP), defined by Balcik *et al.* (2008) as the final stage of the relief chain, refers to delivery of relief supplies from local and temporary distribution centers to the people in the affected areas (demand locations). It combines location and routing decisions. For instance, Özdamar (2011) presents a planning system to coordinate helicopters in the last mile distribution problem after a disaster.

Selective vehicle routing problem has been also considered in humanitarian context. Allahviranloo *et al.* (2014) explain that this kind of problem is especially useful in humanitarian logistics due to the possibility of damaged routes. The authors propose a classical genetic algorithm and three parallel versions to solve different variants of the problem under uncertainty: reliable, robust and fuzzy selective vehicle routing problem.

In humanitarian relief operations context, as Campbell *et al.* (2008) explain, it is critical that the deliveries to affected sites are both fast and fair. The authors suggest that using service-based objective functions may better reflect the different priorities and strategic goals found in delivering humanitarian aid. One of this service-based objective functions is the sum of arrival times, explored in the next section, while Section 2.2.2 is dedicated to alternative service-based objectives.

2.2.1 Cumulative Objective Functions

The cumulative objective function is used in problems aiming at minimizing the sum of arrival times (or weighted arrival times). This objectives received special attention in some industrial applications like maintenance, or when CO₂ emissions or fuel consumption must be minimized as a function of vehicle load. They can be also applied to humanitarian logistics due to their focus on service and their relation with the survival rate of victims.

In relief context, it is critical that the deliveries should take into account the urgency of the situation. In that sense service-based objective functions, like the minimization of the sum of arrival times, better describes the emergency of humanitarian logistic operations than classical objective functions such as the minimization of total tour length. Note that

the minimization of the sum of arrival times is equivalent to the minimization of the average arrival time.

The sum of arrival times has already been used in single-tour problems. For instance, the minimum latency problem (MLP) consists in finding a tour starting at a depot and visiting each other node only once, in such a way that the total latency is minimized (Blum *et al.*, 1994; Archer & Williamson, 2003). The latency of a node is defined as the total distance or travel time to reach that node. This problem is also known as the delivery man problem (DMP) (Fischetti *et al.*, 1993) or the travelling repairman problem (TRP) (Tsitsiklis, 1992; Jothi & Raghavachari, 2007) because of its possible applications to maintenance operations. Picard & Queyranne (1978) and Simchi-Levi & Berman (1991) study its applications in machine scheduling problems.

Fischetti *et al.* (1993) develop a mathematical integer formulation, lower bounds, and an enumerative algorithm while Blum *et al.* (1994) present a dynamic programming approach for the TRP. Miniéka (1989) propose a pseudo-polynomial algorithm for the special case where nodes belong to a tree network. Two exact algorithms based on dynamic programming and branch and bound methods, as well as a dynamic programming based heuristic, are described in Bianco *et al.* (1993). An online version of the TRP is handled by Krumke *et al.* (2003) with a greedy randomized heuristic. A Greedy Randomized Adaptive Search Procedure with Variable Neighborhood Search (GRASP+VND) is introduced in Salehipour *et al.* (2008).

Many approximation algorithms have been also developed (Blum *et al.*, 1994; Garg, 1996; Goemans & Kleinberg, 1998; Arora & Karakostas, 2000; Archer & Williamson, 2003; Fakcharoenphol *et al.*, 2003).

The multiple travelling repairman problem (k -TRP) is a generalization of the minimum latency problem where k tours must be determined (Jothi & Raghavachari, 2007). Jothi & Raghavachari (2004), Fakcharoenphol *et al.* (2007) and Jothi & Raghavachari (2007) provide approximation algorithms.

The time-dependent travelling salesman problem (TDTSP) is a generalization of the standard TSP and the TRP in which the traversal cost of an arc depends on its position in the tour (Picard & Queyranne, 1978; Lucena, 1990; Gouveia & Voss, 1995). In Lucena (1990) a branch and bound algorithm is developed and tested with TRP instances. A Bender's decomposition methodology is used by Vander Wiel & Sahinidis (1996). Heuristic

solutions are also computed by Vander Wiel & Sahinidis (1995): a local search heuristic based on the Lin-Kernighan tour improvement heuristic and a Benders-decomposition-based heuristic. Gouveia & Voss (1995) compare different mathematical formulations and provide one based on a quadratic assignment formulation. Malandraki & Dial (1996) propose a heuristic method based on a restricted dynamic programming approach.

Picard & Queyranne (1978) apply the TDTSP to solve one-machine scheduling problems via a branch and bound method based on a shortest path formulation and compare different integer formulations.

Among applications requiring several capacitated vehicles, school bus routing problems (SBRP) frequently consider similar objective functions. Li & Fu (2002) present a multi-objective algorithm for the SBRP with four objectives, one of them being the minimization of total student travel time. Bennett & Gazis (1972) design a bi-objective approach minimizing both the total bus travel time and the total student travel time. A survey on SBRP can be found in Park & Kim (2010), where seven objective functions are identified. Three of them are service-based: total student riding distance or time (or total affected individual waiting time in relief distribution), student walking distance and student's time loss.

Kara *et al.* (2007, 2008) tackle another vehicle routing problem that they call cumulative VRP. They explain that the cost of a route is not adequately represented by the distance between sites. Some cost factors such as fuel consumption, vehicle depreciation and maintenance can be captured by a function of the load on the arcs. The contribution to the cost proposed for each arc is the arc length multiplied by the flow traversing the arc (the vehicle load). The objective function is the sum of these products over the set of traversed arcs, which can be translated as the sum of weighted arrival times. This model can be considered as a generalization of the k -TRP. Kara *et al.* (2007) formulate integer linear programs for the collection and delivery cases, and additional extensions for the distance constraint cases. Kara *et al.* (2008) model the problem as an integer linear program and test it on the highway network of Turkey to minimize the energy consumption of a fleet of vehicles.

The cumulative capacitated vehicle routing problem (CCVRP) is a variant of the classical CVRP where the objective function becomes the sum of arrival times at demand sites (Campbell *et al.*, 2008; Ngueveu *et al.*, 2009, 2010). The CCVRP can also be considered as the generalization of the minimum latency problem to several capacitated vehicles.

The CCVRP was first studied by Campbell *et al.* (2008). They perform a comparison between cost minimization, maximal arrival time minimization and average arrival time minimization for the TSP and the CVRP. Their paper presents lower bounds, an insertion heuristic and a local search procedure.

Ngueveu *et al.* (2009, 2010) provide a mathematical model, several lower bounds and two memetic algorithms. Solutions are compared using classical CVRP instances from Christofides *et al.* (1979), replacing the total length of the routes by the sum of arrival times, and on TRP instances from Salehipour *et al.* (2008). Nevertheless their results are not comparable with the results of Campbell *et al.* (2008). The results confirm that the CCVRP and the classical CVRP can have quite different solutions on the same instance, as already observed by Campbell *et al.* (2008). In Euclidean versions for instance, CVRP solutions with edge-crossings are suboptimal, while the elimination of these crossings often brings no improvement for the cumulative objective (Campbell *et al.*, 2008).

More recently, Ribeiro & Laporte (2012) have presented an adaptive large neighborhood search (ALNS) algorithm for the CCVRP, which is compared with the memetic algorithms in Ngueveu *et al.* (2010) using additional instances from Golden *et al.* (1998). Destroy and repair operators are used to generate new solutions, and the best-performing neighborhood is selected with an adaptive probabilistic rule.

Ke & Feng (2013) analyze a two-phase metaheuristic which applies exchange-based and cross-based operators to perturb the solutions in the first phase and a local search procedure in the second phase. This metaheuristic improves some best known solutions from previous works (Ngueveu *et al.*, 2010; Ribeiro & Laporte, 2012).

Chen *et al.* (2012) propose an iterative local search algorithm based on insertion, swap and 2-opt moves. A comparison between three metaheuristics: a tabu search, a genetic algorithm and an evolutionary algorithm involving a particle swarm optimization mechanism with genetic operators, is presented in Ozsoydan & Sipahioglu (2013).

Lysgaard & Wøhlk (2014) develop a branch and cut and price algorithm for the CCVRP able to solve instances up to 69 required sites.

A multiobjective mathematical model is proposed by Huang *et al.* (2012) for the last mile distribution problem (LMDP) where three performance measures are used: efficiency

is measured as the total travel time, efficacy is equivalent to the sum of arrival times, and equity computes the difference between service levels of all required sites.

Note that although vehicles must perform closed-routes, i.e. the vehicles must return to the depot, the travel time of the last arc is not counted since it does not affect any required site. This is a common feature with the Open VRP (OVRP), another CVRP variant where each route ends at its last customer, without returning to the depot (Salari *et al.*, 2010a; Repoussis *et al.*, 2010). Hence, like in the CCVRP, the distance to go back to the depot is ignored in the objective function. Nevertheless, Ngueveu *et al.* (2010) show on an example that optimal OVRP and CCVRP solutions are also quite different.

2.2.2 Alternative Objective Functions

Other authors consider different objective functions to model relief distribution in humanitarian logistics. The main examples are the maximization of the demand satisfaction, the maximal arrival time minimization, and the weighted sum of multiple objective functions.

In the VRP with maximal arrival time minimization, also known as min-max VRP, the goal is to minimize the maximum arrival time among all sites, discarding the arrival times at the intermediate sites of each route, which can vary considerably if many sites are served early or late on their route. The min-max VRP is studied by Campbell *et al.* (2008) and compared with the minimization of the total travel time and the minimization of the sum of arrival times. Other version of the min-max VRP consider the minimization of the longest tour. Applegate *et al.* (2002) also address this objective via a branch-and-cut algorithm, while Hemel *et al.* (1996) solve a practical problem aiming at minimizing the maximal tour length, using the average arrival time to break ties. Dell *et al.* (1996) add a multi-period horizon and equity constraints. Approximation algorithms are proposed by Arkin *et al.* (2006) for the minimum and min-max VRP. The min-max windy rural postman problem with k vehicles is addressed by Benavent *et al.* (2009) and Benavent *et al.* (2010) via a branch-and-cut and a multistart iterated local search (MS-ILS), respectively.

Covering models are also considered for emergency relief. Naji-Azimi *et al.* (2012a) propose a covering tour approach to supply humanitarian aid where satellite distribution centers must be located in such a way that all demand points are covered within a maximal distance and satellites distribution centers must be supplied from a central depot. The same problem is addressed by Salari & Naji-Azimi (2012) using an integer programming-based

local search algorithm. Hà *et al.* (2013) propose an exact approach and an evolutionary local search for the multi-vehicle covering tour problem with a constraint on the number of vertices.

Other covering tour approaches include the formulations and heuristic proposed by Current & Schilling (1994) for the bi-objective median tour and maximal covering tour problem. Jozefowicz *et al.* (2007) also deduce a bi-objective model for the maximal covering tour problem which is solved with a hybrid algorithm combining an evolutionary algorithm and a branch and bound method. Naji-Azimi & Salari (2014) describe the time-constrained maximal covering salesman problem and construct an integer linear programming-based heuristic for this problem.

Chen *et al.* (2006) formulate a real-time time-dependent vehicle routing problem as a sequence of mixed integer programming models in which the objective is a weighted function of travel times, waiting times before service and waiting times before departure at all sites.

In the context of the last mile distribution problem, Balcik *et al.* (2008) present a mathematical formulation where the total travel cost and a penalty cost associated with unsatisfied demand are summed in a unique objective function.

Similarly, Hsueh *et al.* (2008) introduce a dynamic vehicle routing problem for relief logistics (DVRP-RL) which considers that pick-up and deliveries can vary and be known in real time manner. They formulate DVRP-RL as a sequence of mixed integer programming models on a rolling time horizon. The objective of the DVRP-RL is to minimize the total travel time of all routes and the total penalty due to late arrivals.

Holguín-Veras *et al.* (2013) discuss the appropriate objective function to be used in post-disaster humanitarian logistics. In their survey, most of the analyzed papers use logistic cost (or related) objectives, while unmet demands and social cost are the least used objectives.

2.3 Vehicle Routing Solution Approaches

In this section the main solution approaches to vehicle routing problems are presented. Solution approaches are divided in five categories: a) exact methods, which guarantee to find an optimal solution but are computationally costly, b) constructive heuristic methods, which

build a single solutions in a very fast way by using simple rules, c) improvement heuristics, focused on the iterative improvement of an existing solution, d) metaheuristics, which include mechanisms to escape from local optima and get better solutions at the expense of longer running times, and e) matheuristics, methods that have become more popular in recent years by combining exact methods with (meta)heuristic techniques.

2.3.1 Exact Methods

The exact algorithms, as written before, determine an optimal solution. They enumerate implicitly all feasible solutions to find the best one, and discard families of solutions to accelerate the search. However, as they require a large computational effort, only small instances can be solved. Some representative examples are described in the sequel.

Branch and bound algorithms are used to solve discrete and combinatorial optimization problems. They enumerate the candidate solutions on a tree space search. When the algorithm inspects one node of the tree, lower bounds are computed and compared against an upper bound. The node (and its descendents) is discarded if it cannot produce a better solution than the best one found so far by the algorithm. Normally, nodes represent partial solutions. Examples of these techniques for the CVRP are developed by Christofides & Eilon (1969), Christofides (1976), Christofides *et al.* (1981a), Laporte *et al.* (1986), Fisher (1994), Fischetti *et al.* (1994) and Hadjiconstantinou *et al.* (1995). More recently, Turkensteen *et al.* (2008) have dealt with the asymmetric TSP, Zhang *et al.* (2012) the single vehicle routing problem with toll-by-weight, and Almoustafa *et al.* (2013) the asymmetric distance-constrained vehicle routing problem.

The *branch and cut* technique uses valid linear inequalities (cutting planes) to strengthen the linear programming relaxation at each node of a branch and bound tree. This technique is often applied on flow based formulations where the set of decision variables x_{ij} indicates whether a vehicle traverses the arc (i, j) ($x_{ij} = 1$) or not ($x_{ij} = 0$). If decision variables are expressed as the load on the traversed arc (i, j) , different models can be formulated. Some examples can be found in Golden *et al.* (1977), Gavish & Graves (1979), Fisher & Jaikumar (1981), Lysgaard *et al.* (2004), Baldacci *et al.* (2004) and Gounaris *et al.* (2011) to solve the CVRP. Branch and cut has been also used to solve other vehicle routing applications such as the location-routing problem (Belenguer *et al.*, 2011), the location-routing problem with simultaneous pickup and delivery (Karaoglan *et al.*, 2011), the school bus routing (Riera-Ledesma & Salazar-González, 2012), the traveling purchaser problem (Batista-Galván *et al.*, 2013), the hub location and routing problem (Rodríguez-Martín *et al.*, 2014) and the

split delivery VRP (Archetti *et al.*, 2014).

In *column generation*, the decision variables, also called columns, are associated to patterns, for instance a list of customers to visit. As patterns are generally extremely large, a restricted subset of patterns is gradually constructed. This method generates only patterns which have the potential to improve the objective function, i.e. to the ones with negative reduced cost. Two subproblems are solved iteratively: a master problem which finds the best configuration of patterns, and a pricing problem which find new patterns that can reduce the cost. The process terminates when the pricing problem returns a solution with non-negative reduced cost. Examples of vehicle routing applications include the heterogeneous fleet VRP (Choi & Tcha, 2007), the split delivery VRP (Jin *et al.*, 2008), a rich vehicle routing and inventory problem (Oppen *et al.*, 2010), a waste collection routing problem (Hauge *et al.*, 2014) and a ship routing and scheduling problem in oil transportation with split deliveries (Nishi & Izuno, 2014).

Branch and price is a branch and bound method in which at each node of the search tree, columns may be added. Some examples can be found in Gutiérrez-Jarpa *et al.* (2010) for the VRP with deliveries, selective pickups and time windows, Salani & Vacca (2011) for the VRP with discrete split deliveries and time windows, Santos *et al.* (2011) for the VRP with cross-docking, Dayarian *et al.* (2014) for the multi-period VRP and Taş *et al.* (2014) for the VRP with soft time windows and stochastic travel times.

When branch and bound works by adding columns as well as cuts, it is called *branch and cut and price*. This strategy is used by Fukasawa *et al.* (2006) to solve the CVRP, by Martinelli *et al.* (2011) for the CARP, by Stålhane *et al.* (2012) for the ship routing and scheduling problem with split loads, by Luo *et al.* (2014) for the multiple TRP with distance constraints and by Gauvin *et al.* (2014) for the VRP with stochastic demands.

Dynamic programming splits an optimization problem recursively, into nested subproblems. This method is based on the principle that an optimal solution has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal solution with regard to the state resulting from the first decision. Examples are proposed for the CVRP by Eilon *et al.* (1971), Christofides *et al.* (1981b), Laporte & Nobert (1983), Laporte *et al.* (1985), Naddef & Rinaldi (2002) and Lysgaard *et al.* (2004).

The VRP and most of their variant belong to the NP-Hard class (Lenstra & Kan,

1981), so only non-polynomial algorithms are known to solve relatively large instances to optimality. Due to this feature, heuristic approaches must be used in practice to solve larger instances. By opposition to exact methods, heuristics find a relatively good solution by using significantly less computational effort. In the sequel, the more common approaches are described.

2.3.2 Constructive Heuristics

In this section, the simpler algorithms to solve the CVRP are described. On every step of these algorithms, a greedy function is used to make a decision which cannot be reversed, leading to a single solution at the end. The following are some representative methods.

Savings algorithms are among the most popular heuristics in routing problems. The first one was proposed by Clarke & Wright (1964). Although there are several kinds of implementations, they are based on merging routes to get a positive saving, computed as $s_{ij} = w_{0i} + w_{j0} - w_{ij}$ if the first route ends at customer i and the second begins with customer j . Note that this formula has sense for the classical cost-based objective function, but not for others like the sum of arrival times or the satisfaction of demands, which are more common in disaster logistics. Other authors have proposed modifications in order to improve the Clarke and Wright algorithm (Golden *et al.*, 1977; Altinkemer & Gavish, 1991; Wark & Holt, 1994; Paessens, 1988; Nelson *et al.*, 1985).

Petal algorithms, first proposed by Balinski & Quandt (1964), builds a set of routes, called petals, and selects a subset of them by solving a set covering model. A simpler implementation is proposed in Gillett & Miller (1974), called *sweep* algorithm, where a set of non-overlapping petals are generated. Similar algorithms are found in Foster & Ryan (1976), Ryan *et al.* (1993) and Renaud *et al.* (1996b).

In *Cluster-first route-second* algorithms, first proposed by Fisher & Jaikumar (1981), R seeds are located in the first step. The set of sites to visit are clustered by using a generalized assignment model to minimize the sum of nodes-to-seed distances. Then, a route is determined on each cluster by solving a traveling salesman problem. Baker & Sheasby (1999) describe some procedures to select seeds and Bramel & Simchi-Levi (1995) present a capacitated location model to select seeds.

The *route-first cluster-second* approach first creates a giant circuit that visits all customers. Then, the circuit is cut optimally in several feasible routes starting and ending

at the depot (Beasley, 1983; Prins, 2004).

A more descriptive review about vehicle routing heuristics can be found in Laporte (2002).

2.3.3 Improvement Heuristics

Improvement heuristics, also called neighborhood search in local search, are frequently used as post-optimization processes to improve a heuristic solution or an intermediate solution, in a more general approach as metaheuristics or matheuristics.

The neighborhood of a solution is a relatively small subset of solutions, obtained using elementary transformations called moves. Hence, most neighborhoods are implicitly defined by their moves.

Intra-route moves: This category refers to moves which are designed to improve each route in a separate way, which means that any improvement heuristic for the traveling salesman problem belongs to this category.

Lin (1965) described a very general heuristic (λ -opt), which consist in removing λ edges from a route to reconnect the remaining segments in all possible ways. The parameter λ is usually limited to 2 or 3. Lin & Kernighan (1973) propose a dynamic control of this parameter.

The Or-opt method, proposed by Or (1976), relocate a string of consecutive nodes (limited to 3 nodes).

A modified version of 4-opt, called 4-opt*, is proposed by Renaud *et al.* (1996a) where only promising reconnections between a chain with a number of edges limited by a user-parameter and another chain of two edges can be evaluated.

A review of this kind of moves can be found in Laporte (2010).

Inter-route moves: In this category moves which generate changes on several routes simultaneously are considered.

Thompson & Psaraftis (1993) present the b -cyclic k -transfer where b is the number of routes organized in a circular permutation and k is the number of nodes relocated from a

route to the next one.

Several kinds of operations are defined by Breedam (1994): string cross, string exchange, string relocation and string mix. String cross consists in removing one edge from two routes and relinking the resulting strings with different edges. String exchange takes two strings of at most k nodes in different routes and exchange them. String relocations move a string from a route to another route. The string mix move selects the best move between string exchange and string relocation.

Other moves have been proposed depending on the different features considered. For instance, multitrip, multi-depot or split delivery need different moves which cannot be applied in a general way.

The implementations also differ in the move finally selected to modify the current solution. In the best improvement strategy, all possible moves are evaluated and the best improving one is executed. In the first improvement strategy, the first improvement found is applied. An intermediate strategy is proposed by Vidal *et al.* (2013) where the best improvement solution, if it exists, is selected when 5% of the neighborhood is explored.

2.3.4 Metaheuristics

Metaheuristics are general-purpose algorithms that can be applied to solve almost any optimization problem (Talbi, 2009). The success of these methods is based on the balance between intensification and diversification strategies. Intensification strategies are related to the goal of improving quickly a solution by exploiting a promising solution structure, while diversification strategies are oriented to explore different regions of the solution space, escaping from local optima. The most common intensification mechanism is to apply improvement heuristics. Usually, compared with intensification, diversification mechanisms require less computational effort. The most relevant and successful metaheuristic strategies for vehicle routing problems are briefly described in the sequel.

In *tabu search* (TS), improvement heuristics are used as intensification mechanism while diversification takes place with short-term (tabu list) and long-term memories. Examples of this method can be found in Cordeau *et al.* (1997) for a periodic and multi-depot VRP and in Derigs & Kaiser (2007) for the CVRP.

Simulated annealing (SA) explores randomly the neighborhood to search improvements, but accepts non-improvements to escape from local optimal solutions with a probability which depends on the quality of the solution and a timing control parameter. *Threshold-accepting* algorithm (Dueck & Scheuer, 1990) and *record-to-record* travel algorithm (Dueck, 1993) are special cases where the acceptance of a solution is not probabilistic.

Genetic algorithms (GA) is a population-based metaheuristic where offspring, created from selected parents of current population, replace the worst individuals of the population to maintain a high quality set of solutions. Their principal components are selection, crossover (intensification) and mutation (diversification). Some implementations are combined with improvement heuristics to give *memetic algorithms* (Prins, 2004; Mester & Bräysy, 2005; Nagata, 2007; Mester & Bräysy, 2007; Prins, 2009a). Initial population is normally generated randomly. Other implementations generate a set of high-quality solution by using improvement heuristics, recombine that set to create a new population and restart the improvement heuristics (Rochat & Taillard, 1995; Tarantilis & Kiranoudis, 2002; Tarantilis, 2005).

Variable neighborhood search (VNS) makes use of different improvement heuristics as intensification strategy. Several neighborhoods alternate to allow different search directions. An application of VNS to very large scale vehicle routing problems is described in Kytöjoki *et al.* (2007).

The *very large neighborhood search* destroys and reconstructs simultaneously some parts of the solution as a general improvement heuristic. Shaw (1998) make use of constraint propagation in the re-insertion process while Ergun *et al.* (2006) define a multi-label shortest path problem to find good neighbors heuristically.

Several insertion and removal heuristics are used by the *adaptive large neighborhood search* (ALNS) which reproduce the effect of many complex neighborhoods. This strategy has been used by Ropke & Pisinger (2006) and Pisinger & Ropke (2007).

Iterated local search (ILS) (Lourenço *et al.*, 2010) and *evolutionary local search* (ELS) (Wolf & Merz, 2007) are based on the use of local search and mutation procedure. The idea behind ILS is to iteratively apply perturbation and local search procedures to a current solution, and it has been apply to VRPs in Hashimoto *et al.* (2008) and Penna *et al.* (2013). ELS extends ILS by creating a set of children at each iteration instead of only one. ELS has

been used by Prins (2009a), Duhamel *et al.* (2010) and Duhamel *et al.* (2011).

The *granularity principle* is a strategy to concentrate the search in the most promising edge by removing the longest from the network. This principle has been applied by Toth & Vigo (2003) in a tabu search method (granular tabu search) and by Li *et al.* (2005) in conjunction with a record-to-record algorithm.

Penalty functions can also be used in order to intensify or diversify the search process. Intensification strategies consist in penalizing the most frequently performed moves while diversification strategies allow to explore infeasible solutions (Taillard, 1993; Gendreau *et al.*, 1994; Cordeau *et al.*, 1997, 2001).

Frequency based memory also allows to increase the preference for some solution components when their appear on high quality solutions in *ant colony optimization* algorithms (ACO). Reimann *et al.* (2004) describe an ACO for the CVRP.

2.3.5 Matheuristics

Matheuristics are hybrid heuristic algorithms consisting in the interoperation of metaheuristics and mathematical programming techniques (Boschetti *et al.*, 2009; Caserta & Voß, 2010).

One common strategy for VRPs is to create a set of trips by a (meta)heuristic and then build a feasible solution using a set partitioning model. This strategy is used by Salhi *et al.* (2013) for a fleet size and mix VRP, and by Alvarenga *et al.* (2007) and Dhahri *et al.* (2014) for the VRP with time windows (VRPTW). Villegas *et al.* (2013) solve the truck and trailer routing problem by hybridizing an ILS with a set partitioning formulation. Allaoua *et al.* (2013) apply similar techniques for a home health care problem.

Some Very Large-scale Neighborhood Search (VLNS) algorithms (Ahuja *et al.*, 2002) define their neighborhoods based on network flow techniques or dynamic programming. VLNS has been applied to routing problems by in Thompson & Psaraftis (1993) and Glover & Punnen (1997). The corridor method is a dynamic programming-based VLNS (Sniedovich & Voß, 2006) which is applied to TSP.

Yagiura & Ibaraki (1996) use dynamic programming to optimize crossover operators in genetic algorithms applied to the TSP and two additional combinatorial problems.

In Schittekat *et al.* (2013) an exact algorithm assigns student to stops for a given set of routes for the school bus routing problem with stop selection.

The Integer Linear Programming-based heuristic (ILP) is an ALNS algorithm in which insertions or reconstruction phase is performed by an integer programming model. This approach is applied by Franceschi *et al.* (2006) and Toth & Tramontani (2008) for the capacitated vehicle routing problem, by Salari *et al.* (2010a) for the open VRP, by Salari & Naji-Azimi (2012) for the covering salesman problem, by Naji-Azimi *et al.* (2012a) for the m -Ring-Star problem, by Coelho *et al.* (2012) for the inventory-routing problem, and by Guerrero *et al.* (2013) for the inventory location-routing problem.

Finally, splitting procedure or split procedure is a common method in vehicle routing problems. For a given sequence of nodes (or TSP-like solution) it finds an optimal set of trips respecting that sequence. This procedure introduced by Beasley (1983) has been used for the first time in a metaheuristic by Prins (2004). Prins *et al.* (2014) present a survey on this technique and its adaptations. In Section 4.4 a more detailed description of the procedure is presented.

A review of key-references on exact and heuristic methods for the CVRP and its variations can be found in Laporte (2009).

2.4 Concluding Remarks

Most optimization problems raised by humanitarian logistics focuses on a specific phase. For instance, inventory systems researches are mainly concerned by the preparedness phase, where prepositioning becomes challenging because in most cases it is not possible to predict when and where disaster will occur, and how many people will be affected. Location and transportation problems are more frequently applied to response operations, where urgency and an important number of new demands are the main features. Infrastructure restoration takes place when communities start to come back to normalcy in recovery phase.

As noted before, the most important feature in response phase is the emergency of the situation. Nevertheless, many vehicle routing and transportation problems for relief effort still use cost-based objectives. In Section 2.2, various serviced-based objective functions for vehicle routing problems are reviewed. This section shows that few works have treated these alternative objectives which impact deeply the solution methods, for instance the design of

local search procedures.

From these conclusions, the aim of this research is precisely to investigate vehicle routing problems raised by the response phase after a disaster and characterized by cumulative objective functions.

Chapter 3

The Multitrip Cumulative Capacitated Single-Vehicle Routing Problem: An Exact Solution Approach

3.1 Introduction

The aim of this chapter is to propose an exact method based on a resource-constrained shortest path formulation for a new version of the capacitated vehicle routing problem (CVRP). An application case of this problem can be found in the response phase of relief operations, in which the classical objective function (total time or distance traveled) becomes the sum of arrival times at affected sites. In this version, a single vehicle with a limited capacity should satisfy all the demand by several trips. This problem, called the *multitrip cumulative capacitated single-vehicle routing problem* (mt-CCSVRP), is described in Section 3.2.

Angel-Bello *et al.* (2013) propose and compare two mathematical models for the mt-CCSVRP with a maximal number of trips. These models are tested with randomly generated instances and can solve instances with no more than 25 sites.

Azi *et al.* (2007) develop an exact algorithm based on an elementary resource-constrained shortest path (ERCSP) formulation for the single-vehicle routing problem with time windows and multiple trips. On some instances it is not possible to cover all sites because of the time windows. That is why solutions with a maximum number of visited sites are preferred, and minimum total cost is used to break ties. The authors also provide a mathematical program and some dominance rules. The solution method is based on

the elementary shortest path algorithm with resource constraints proposed by Feillet *et al.* (2004), which is a label correcting algorithm. Additional information about shortest path algorithms for transportation applications can be found in the survey of Fu *et al.* (2006).

A general description of the algorithm developed is depicted in Section 3.3. This algorithm is divided in two phases: Auxiliary network generation (Section 3.4) and Shortest path solution approach (Section 3.5).

In Section 3.5, an exact approach inspired by the ERCSP formulation and the algorithms of Feillet *et al.* (2004) and Azi *et al.* (2007) is developed. This algorithm differs from the one proposed by Azi *et al.* (2007) because it always visits all affected sites, the auxiliary network is acyclic (which censors elementary paths) and the objective function is the sum of arrival times. It also involves different dominance rules.

The chapter is structured as follows: in Section 3.2 the problem is formally defined and modelled by two mixed linear programs. A general description of the algorithmic methodology is summarized in Section 3.3. Section 3.4 describes the procedure to transform the mt-CCSVRP in a resource-constrained shortest path problem on an acyclic directed graph. A shortest path algorithm with resource constraints is developed in Section 3.5. Computational results are presented in Section 3.6 while concluding remarks are given in Section 3.7.

3.2 Problem Definition and Mixed Integer Linear Models

The multitrip cumulative capacitated single-vehicle routing problem (mt-CCSVRP) uses the sum of arrival times at required sites as objective function, like in the cumulative CVRP, but involves a single vehicle which can perform more than one trip. This flexibility is necessary when the total demand exceeds the capacity of the vehicle or the range of the vehicle is limited. This problem can also be seen as a generalization of the traveling repairmen problem (TRP) with replenishment arcs.

In the context of humanitarian logistics, the mt-CCSVRP models the distribution of relief supplies to a set of sites affected by a disaster, using for instance one helicopter which can do multiple sorties. After a disaster, the victims are waiting for rescue and multiple crews can be coordinated to use the vehicle full time. Nevertheless, the limited range of the vehicle imposes an upper bound on the flight time of each trip.

Although in a realistic intervention several vehicles can be required, the single-vehicle version presented in this chapter can be useful in a more general solution method where the affected area is divided into elementary zones, each zone being assigned to one vehicle.

The problem can be defined on a complete undirected graph $G = (V, E)$. The node-set $V = \{0, \dots, n\}$ includes one depot-node 0 and a subset $V' = V \setminus \{0\}$ of n locations affected by a disaster, also called *required sites*. The set E is composed by edges (i, j) with travel times w_{ij} satisfying the triangle inequality, also called *flight times*. A single vehicle of capacity Q and range L_{max} (in terms of flight time) is based at the depot. Each required site $i \in V'$ has a known demand q_i and service time s_i . It is assumed without loss of generality that $\sum_{i \in V'} q_i \geq Q$ and $q_i \leq Q \forall i \in V'$.

The objective is to identify an ordered sequence of trips $(1, 2, \dots, v)$, called *multitrip*, such that each required site is visited exactly once and the sum of arrival times at the sites is minimized (in the last trip, the return to the depot is not counted). A trip k is a cycle, starting and ending at the depot, whose total load W_k fits vehicle capacity Q and total flight time L_k does not exceed the range L_{max} . Each trip k has also a setup time (loading time) μ_k before leaving the depot, computed as the sum of service times over all required sites served by the trip, weighted by a given coefficient β . The loading times and service times do not affect vehicle range but delay the arrivals at required sites.

In this version, the number of trips is a decision variable. Contrary to the mtVRP, note that the cumulative objective function is sensitive to the trip ordering: Theorem 3.3 in Section 3.4 defines the optimal ordering for a given set of trips. Figure 3.1 illustrates a small mt-CCSVRP instance with five required sites and one feasible solution with three trips. Each site demand is equal to 10 while the vehicle capacity is set to 20. Null service times are used ($s_i = 0 \forall i \in V'$). In the figure, every node denote a site and arcs represent the roads between sites. Dashed arcs correspond to transitions between two successive trips, called replenishment arcs, which indicate the return to depot and the beginning of a new trip. The arrival time t_i at site i is given near to each site-node while the values on each arc are the flight times. The cost of this solution is $Z = \sum_{i \in V'} t_i = 190$, where t_i indicates the arrival time at site i .

In the following, two mixed integer linear programs (MILP) for the mt-CCSVRP are developed. The first one is a flow-based model whose variables have no trip index, which is not at all trivial. The second one is based on a set partitioning model where columns

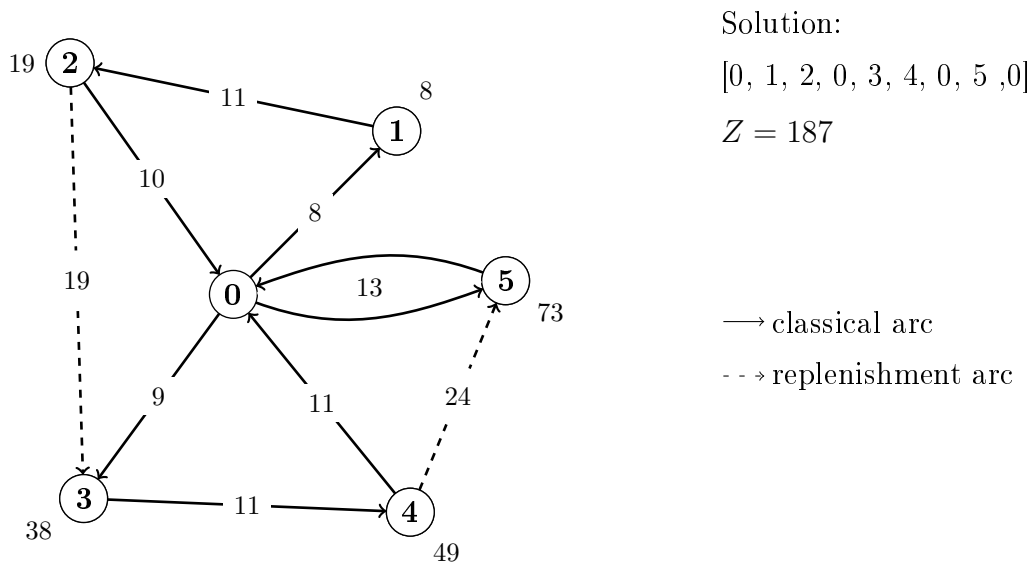


Figure 3.1: Example of an mt-CCSVRP solution for an instance with $n = 5$, all $q_i = 10$ and $Q = 20$.

correspond to the possible routes and each site must be covered (visited) by exactly one route.

3.2.1 Flow-Based Model

The model presented in this subsection is a mixed integer linear program in which the concepts of *replenishment arc* and *arc coefficient* are essential.

The notion of *replenishment arc* has been used by Boland *et al.* (2000) and Mak & Boland (2000) for a multitrip traveling salesman problem in which one vehicle with limited capacity must replenish (reload) at the depot. Replenishment arcs constitute a nice trick to replace the successive trips of a multitrip by a single trip: when a trip with last customer i is followed by a trip with first customer j , the two arcs $(i, 0)$ and $(0, j)$ are replaced by a replenishment arc (i, j) . Here the same technique is used to model the mt-CCSVRP: a multitrip is a sequence of arcs in which replenishment arcs delimit successive trips. The length of a replenishment arc (i, j) is $w'_{ij} = w_{i0} + w_{0j}$.

The concept of *arc coefficient* is defined by Ngueveu *et al.* (2010) for a memetic algorithm for the CCVRP. The sum of arrival times C_k for a trip k defined without loss of generality by a sequence of M_k required nodes $(1, 2, \dots, M_k)$ can be written as:

$$C_k = \sum_{i=1}^{M_k} t_i = t_1 + t_2 + \dots + t_{M_k} \quad (3.1)$$

where $t_i = \mu_k + \sum_{j=0}^{i-1} (s_j + w_{j,j+1})$ is the arrival time to site i , s_i denotes the service time at site i , $s_0 = 0$ and $\mu_k = \beta \cdot \sum_{j=1}^{M_k} s_j$ is the loading time of the trip. C_k can be rewritten as:

$$\begin{aligned} C_k &= M_k \cdot \mu_k + M_k \cdot (s_0 + w_{01}) + (M_k - 1) \cdot (s_1 + w_{12}) + \dots + (s_{M_k-1} + w_{M_k-1, M_k}) \\ C_k &= M_k \cdot \mu_k + \sum_{j=0}^{M_k-1} (M_k - j) \cdot (s_j + w_{j,j+1}) \end{aligned} \quad (3.2)$$

In other words, the traversal time of the first arc in the route is counted M_k times, the cost of the second arc $M_k - 1$ times, and so on. The number of times the cost of arc (i, j) is counted is what we call the *coefficient* of (i, j) . Note that this coefficient is also the number of demand sites visited after site i in the route. The concept can be extended to the replenishment arcs in the multitrip CCVRP.

Similarly, the loading time μ_k is counted M_k times in the objective function. This coefficient is referred as loading time coefficient in the sequel.

In the model, the arc coefficients $(M_k - j)$ from Equation (3.2) are described by variables y_{ij} for arcs (i, j) , while loading time coefficients M_k which multiplies μ_k in the first term of Equation (3.2) is expressed by variables φ_{ij} . Note that two indexes are used to allow modeling the constraints by flow constraints. Variables F_{ij} define the flow on each arc (i, j) , i.e., the load of the vehicle traversing this arc. The binary variables x_{ij} are equal to 1 if and only if arc (i, j) is traversed. The variables x_{0j} are only used at the beginning of the first trip and x_{i0} at the end of the last trip, other depot connections are described by variables x'_{ij} which represent the replenishment arcs. Due to the special nature of replenishment arcs, similar but separate variables x'_{ij} and y'_{ij} are used for these arcs. Finally, the variables f_i are used to respect vehicle range: they represent the cumulated flight time to arrive at site i since the last departure from the depot (without loading and service times), and not the arrival time.

The resulting MILP is defined by Equations (3.3) to (3.34). The objective function (3.3) represents the sum of arrival times to affected sites, extending Equation (3.2) to multitrips and using the variables defined before. The first term computes the loading times for each trip

and the service and flight times for normal arcs multiplied by their corresponding coefficients, while second term computes service and flight times for replenishment arcs multiplied by their arc coefficients. The flight time of replenishment arcs are in fact $w_{i0} + w_{0j}$.

$$\min Z = \sum_{i \in V} \sum_{j \in V} ((s_i + w_{ij}) \cdot y_{ij} + \beta \cdot s_j \cdot \varphi_{ij}) + \sum_{i \in V'} \sum_{j \in V'} (s_i + w_{i0} + w_{0j}) \cdot y'_{ij} \quad (3.3)$$

$$\sum_{j \in V'} x_{0j} = 1, \quad (3.4)$$

$$\sum_{i \in V'} (x_{ij} + x'_{ij}) + x_{0j} = 1, \quad \forall j \in V' \quad (3.5)$$

$$\sum_{i \in V'} (x_{ji} + x'_{ji}) \leq 1, \quad \forall j \in V' \quad (3.6)$$

$$\sum_{j \in V} F_{ji} - \sum_{j \in V'} F_{ij} = q_i, \quad \forall i \in V' \quad (3.7)$$

$$F_{ij} \leq Q \cdot x_{ij}, \quad \forall i \in V', j \in V' \quad (3.8)$$

$$F_{0j} \leq Q \cdot (x_{0j} + \sum_{i \in V'} x'_{ij}), \quad \forall j \in V' \quad (3.9)$$

$$y_{0j} + \sum_{i \in V'} (y_{ij} + y'_{ij}) - \sum_{i \in V'} (y_{ji} - y'_{ji}) = 1, \quad \forall j \in V' \quad (3.10)$$

$$y_{0j} = n \cdot x_{0j}, \quad \forall j \in V' \quad (3.11)$$

$$y_{ij} \leq (n - 1) \cdot x_{ij}, \quad \forall i \in V', j \in V' \quad (3.12)$$

$$y'_{ij} \leq (n - 1) \cdot x'_{ij}, \quad \forall i \in V', j \in V' \quad (3.13)$$

$$y_{ij} \geq x_{ij}, \quad \forall i \in V, j \in V' \quad (3.14)$$

$$y'_{ij} \geq x'_{ij}, \quad \forall i \in V', j \in V' \quad (3.15)$$

$$y_{ij} \geq 2 \cdot x_{ij} - x_{j0}, \quad \forall i \in V, j \in V' \quad (3.16)$$

$$y'_{ij} \geq 2 \cdot x'_{ij} - x_{j0}, \quad \forall i \in V', j \in V' \quad (3.17)$$

$$\sum_{i \in V} \sum_{j \in V'} x_{ij} + \sum_{i \in V'} \sum_{j \in V'} x'_{ij} = n, \quad (3.18)$$

$$\sum_{i \in V} \sum_{j \in V'} y_{ij} + \sum_{i \in V'} \sum_{j \in V'} y'_{ij} = \frac{n \cdot (n + 1)}{2}, \quad (3.19)$$

$$\varphi_{0j} = n \cdot x_{0j}, \quad \forall j \in V' \quad (3.20)$$

$$\varphi_{ij} \geq y'_{ij}, \quad \forall i \in V', j \in V' \quad (3.21)$$

$$\varphi_{ij} \geq y_{ij}, \quad \forall i \in V', j \in V' \quad (3.22)$$

$$\varphi_{ij} \geq \sum_{k \in V} \varphi_{ki} - n \cdot (1 - x_{ij}), \quad \forall i \in V', j \in V' \quad (3.23)$$

$$\varphi_{ij} \leq n \cdot (x_{ij} + x'_{ij}), \quad \forall i \in V', j \in V' \quad (3.24)$$

$$f_j \geq w_{0j}, \quad \forall j \in V' \quad (3.25)$$

$$f_j \geq f_i + w_{ij} - L_{max} \cdot (1 - x_{ij}), \quad \forall i \in V', j \in V' \quad (3.26)$$

$$f_i + w_{i0} \leq L_{max}, \quad \forall i \in V' \quad (3.27)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in V, j \in V', i \neq j \quad (3.28)$$

$$x'_{ij} \in \{0, 1\}, \quad \forall i \in V', j \in V', i \neq j \quad (3.29)$$

$$y_{ij} \geq 0, \quad \forall i \in V, j \in V', i \neq j \quad (3.30)$$

$$y'_{ij} \geq 0, \quad \forall i \in V', j \in V', i \neq j \quad (3.31)$$

$$F_{ij} \geq 0, \quad \forall i \in V, j \in V', i \neq j \quad (3.32)$$

$$\varphi_{ij} \geq 0, \quad \forall i \in V, j \in V', i \neq j \quad (3.33)$$

$$f_i \geq 0, \quad \forall i \in V' \quad (3.34)$$

Constraint (3.4) means that only one vehicle is used. Equations (3.5) and (3.6) respectively indicate that exactly one arc is traversed to arrive at each site j and leave it. The last arc of the multitrip is ignored since it has no impact on the objective function. Constraints (3.7) to (3.9) concern flow variables: Equations (3.7) ensure that each demand is satisfied and express the load variation after visit to site i , Equations (3.8) and (3.9) guarantee that vehicle capacity is respected and no flow can traverse an unused arc, and Equations (3.9) limit the maximum flow at the beginning of each trip to the capacity Q .

Constraints (3.10) to (3.13) concern arc coefficients. Constraints (3.10) imply that the coefficients of successive arcs decrease along a trip. Equations (3.11) to (3.13) are related to the maximum arc coefficients, for instance after the first arc of the multitrip n sites are visited and at most $n - 1$ after the remaining arcs. No traversed arc have arc coefficients equal to zero. Constraints (3.12) concern normal arcs and constraints (3.13) concern replenishment arcs. Constraints (3.14) to (3.19) are simple valid inequalities which are useful to reduce running time when the model is solved by a commercial solver. Constraints (3.14) and (3.15) mean that all traversed arcs, except the last one, have a positive arc coefficient, which are greater than 1. Constraints (3.16) and (3.17) imply that traversed arcs (i, j) have coefficients greater or equal to 2, but becomes 1 if the next arc is the last one, $x_{j0} = 1$. Equation (3.18) indicates that exactly n sites must be visited and consequently the arc coefficient of the first arc is n . Equation (3.19) expresses the sum of the arc coefficients

$$\sum_{i \in V} \sum_{j \in V'} y_{ij} + \sum_{i \in V'} \sum_{j \in V'} y'_{ij} = \sum_{i=1}^n i = \frac{n \cdot (n + 1)}{2}.$$

The coefficients used for loading times, given by the variables φ_{ij} , are constrained by Equations (3.20) to (3.24). Constraints (3.20) set the loading time coefficient of the first

arc, similar to Equations (3.11) for the arc coefficients. Constraints (3.21) and (3.22) guarantee that loading time coefficients are greater than the corresponding arc coefficients. Equations (3.23) imply that loading time coefficients only decrease when a replenishment arc is traversed. Constraints (3.24) indicate that loading time coefficients are null for unused arcs.

The range constraint is implemented via Equations (3.25) to (3.27), using the flight time variables f_i . Constraints (3.25) impose a lower bound to the flight time to reach a site. Constraints (3.26) compute the flight time to reach site j when arc (i, j) is traversed. Constraints (3.27) limit the range to L_{max} .

Finally, constraints (3.28) to (3.34) define the seven groups of variables. Note that variables y_{ij} , y'_{ij} , F_{ij} and φ_{ij} are flow variables, and they are not required to be binary nor integer. The flight time variables f_i are positive real by nature. Only the variables x_{ij} and x'_{ij} are binary.

The model has been tested on small instances and computational results are reported in Subsection 3.6.3.

3.2.2 Set Partitioning Model

The mt-CCSVRP can also be modeled as a set partitioning problem. In this model, it is assumed that the set K of all feasible and non-dominated trips is built using a precomputation procedure (see Section 3.4 for details), with the following attributes for each trip $k \in K$: the cost C_k , the duration D_k , the number of required sites visited M_k , and binary indicators $r_k^i = 1$ if site i is visited by trip k .

This model takes advantage of Equation (3.35) to compute the cost of a multitrip, based on the cost, duration and number of sites of each trip. The sum of arrival times Z_p for a multitrip $p = (1, 2, \dots, v)$ composed of v trips can be computed as:

$$\begin{aligned}
 Z_p &= C_1 + D_1 \cdot (M_2 + \dots + M_v) + C_2 + D_2 \cdot (M_3 + \dots + M_v) + \dots \\
 &\quad + C_{v-1} + D_{v-1} \cdot M_v + C_v \\
 Z_p &= C_1 + D_1 \cdot \sum_{k'=2}^v M_{k'} + C_2 + D_2 \cdot \sum_{k'=3}^v M_{k'} + \dots + C_{v-1} + D_{v-1} \cdot M_v + C_v \quad (3.35) \\
 Z_p &= \sum_{k=1}^v \left(C_k + D_k \cdot \sum_{k'=k+1}^v M_{k'} \right)
 \end{aligned}$$

Two types of binary variables are used. The variable χ_k is equal to 1 if trip k is selected in the solution. Variables $\gamma_{kk'}$ define the relative order of the trips in the multitrip: $\gamma_{kk'} = 1$ if trip k is performed before trip k' .

The set partitioning formulation for the mt-CCSVRP is given by Equations (3.36) to (3.42). The objective function (3.36) which represents the sum of arrival times is derived from Equation (3.35), by adding the decision variables.

$$\min Z = \sum_{k \in K} \left(C_k \cdot \chi_k + \sum_{k' \in K} (D_k \cdot M_{k'} \cdot \gamma_{kk'}) \right) \quad (3.36)$$

$$\sum_{k \in K} r_k^i \cdot \chi_k = 1, \quad \forall i \in V' \quad (3.37)$$

$$\gamma_{kk'} \leq \chi_k, \quad \forall k \in K, k' \in K \quad (3.38)$$

$$\gamma_{kk'} \leq \chi_{k'}, \quad \forall k \in K, k' \in K \quad (3.39)$$

$$\gamma_{kk'} + \gamma_{k'k} \geq \chi_k + \chi_{k'} - 1, \quad \forall k \in K, k' \in K \quad (3.40)$$

$$\chi_k \in \{0, 1\}, \quad \forall k \in K \quad (3.41)$$

$$\gamma_{kk'} \in \{0, 1\}, \quad \forall k \in K, k' \in K, k \neq k' \quad (3.42)$$

Constraints (3.37) imply that each required site is visited by exactly one trip. Constraints (3.38) to (3.40) control the order of the trips. Equation (3.38) (resp. (3.39)) inhibits the ordering variable $\gamma_{kk'}$ when trip k (resp. k') is not used in the solution, i.e., when $\chi_k = 0$ (resp. $\chi_{k'} = 0$). Equations (3.40) impose an order between trips k and k' when both belong to the solution.

Finally, Equations (3.41) and (3.42) define the variables. In Subsection 3.6.3 this model is tested and compared with the flow model and the exact method of Section 3.5 on small instances.

3.3 General Description of the Algorithm

The resource-constrained shortest path algorithm developed in this chapter is split in two sequential phases. The first phase translates the single-vehicle routing problem to a shortest path problem where each node is a trip, called trip-node in the sequel, and required sites becomes resources. The generated graph is acyclic due to the property in Theorem 3.3. The

second phase adapts the resource-constrained shortest path algorithm to manage the sum of arrival times objective.

An example of a solution is depicted in Figure 3.2. This figure shows the auxiliary network for the instance of Figure 3.1. In that network, inside each trip-node k (represented by squares), the visited required sites of the trip are recorded. The cost trip C_k for a departure at time 0 and duration trip D_k are given above and under each node, respectively. The solution cost can be computed as a function of the trip costs, trip durations and number of visited sites. The boldfaced arcs represent the feasible solution (multitrip) in Figure 3.1 where the cost can be computed as $Z = (27 + 29 + 13) + 29 \cdot (2 + 1) + 31 \cdot 1 = 187$, following Equation (3.35).

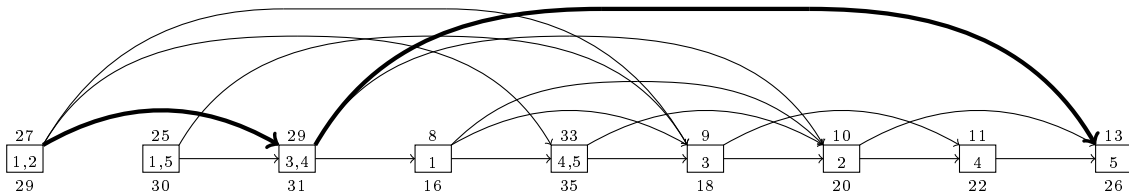


Figure 3.2: Example of a solution on the shortest path network.

3.4 Auxiliary Network for the Shortest Path Problem

This section describes how to transform the mt-CCSVRP into a resource-constrained shortest path problem. A directed acyclic graph (DAG) $G' = (K, E')$, called *auxiliary network*, is built, where each node in K (called *trip-node* to distinguish them from the nodes of the real network G) represents one trip, E' contains one arc (k, k') if trip k can be followed by k' in a multitrip, and the required sites become consumable resources. The precomputation of feasible trips looks very heavy but many trips can be discarded in practice, using two dominance rules stated in Theorems 3.1 and 3.2. The process is also sensitive to the vehicle range L_{max} , as shown in the tests of Section 3.6.

One arc links two trip-nodes k and k' if they have no required site in common and if a crucial dominance property stated in Theorem 3.3 holds. It is this dominance rule which leads to an acyclic graph. A feasible solution is a path, called *complete path*, whose trip-nodes cover all required sites.

3.4.1 Generation of the Set of Trip-Nodes

Each trip k is defined by a cost C_k , a duration D_k , a set of visited required sites B_k , $M_k = |B_k|$, a total load W_k , a last site visited θ_k and a total flight time L_k . Recall that the latter differs from the trip duration D_k which includes service and setup times: $D_k = L_k + \sum_{i \in B_k} (1 + \beta) \cdot s_i$.

The trip set K is initialized with one direct trip per required site. Then, each trip $k \in K$ is extended by adding one required site i at the end, provided i is not already visited ($i \notin B_k$) and capacity and range constraints are satisfied ($W_k + q_i \leq Q$ and $L_k + w_{\theta_k, i} + w_{i, 0} - w_{\theta_k, 0} \leq L_{max}$). The number of trips with the same required sites is strongly reduced using Theorem 3.1 and Theorem 3.2.

Theorem 3.1. *Let k and k' be two trips with sum of arrival times C_k and $C_{k'}$, durations D_k and $D_{k'}$, and visiting the same subset of required sites ($B_k = B_{k'}$). Trip k dominates trip k' if $C_k \leq C_{k'}$ and $D_k \leq D_{k'}$.*

Proof. Let p' be the best complete path in G' containing k' . Its cost (sum of arrival times at sites) can be computed as (recall Equation 3.35):

$$Z_{p'} = \sum_{v=1}^{|p'|} \left(C_v + D_v \cdot \sum_{v'=v+1}^{|p'|} M_{v'} \right)$$

Now, suppose that trip k' is replaced by trip k in path p' , giving path p . The difference between Z_p and $Z_{p'}$ can be computed as:

$$Z_{p'} - Z_p = C_{k'} - C_k + (D_{k'} - D_k) \cdot \sum_{v'=k'+1}^{|p'|} M_{v'}$$

Since it is assumed that $C_k \leq C_{k'}$ and $D_k \leq D_{k'}$, it can be concluded that $Z_{p'} \geq Z_p$, which means that trip k' is dominated by trip k . \square

Theorem 3.2. *Let two trips k and k' with the same required sites ($B_k = B_{k'}$) and the same last site ($\theta_k = \theta_{k'}$). If $C_k \leq C_{k'}$ and $D_k \leq D_{k'}$, then any trip obtained by adding a sequence v of required sites at the end of trip k dominates the trip obtained by adding the same sequence at the end of k' .*

Proof. Let u and u' denote trips k and k' without the depot at the end and \oplus the concatenation operator. If a sequence of required sites $v = (v_1, v_2, \dots, v_{|v|})$ is added to k and k' after the last required site θ_k , the sums of arrival times for the resulting trips $(u \oplus v \oplus 0)$ and $(u' \oplus v \oplus 0)$ are:

$$\begin{aligned} Z_{u \oplus v \oplus 0} &= C_k + \sum_{i=1}^{|v|} \left(D_k - w_{\theta_k, 0} + w_{\theta_k, v_1} + \sum_{j=1}^{i-1} (s_{v_{j-1}} + w_{v_{j-1}, v_j}) \right) \\ &= C_k + |v| \cdot (D_k - w_{\theta_k, 0} + w_{\theta_k, v_1}) + C_v \end{aligned}$$

and:

$$\begin{aligned} Z_{u' \oplus v \oplus 0} &= C_{k'} + \sum_{i=1}^{|v|} \left(D_{k'} - w_{\theta_{k'}, 0} + w_{\theta_{k'}, v_1} + \sum_{j=1}^{i-1} (s_{v_{j-1}} + w_{v_{j-1}, v_j}) \right) \\ &= C_{k'} + |v| \cdot (D_{k'} - w_{\theta_{k'}, 0} + w_{\theta_{k'}, v_1}) + C_v \end{aligned}$$

where $C_v = \sum_{i=1}^{|v|} \left(\sum_{j=1}^{i-1} (s_{v_{j-1}} + w_{v_{j-1}, v_j}) \right)$ is the cost of sequence v .

The following expression indicates the difference between $Z_{u' \oplus v \oplus 0}$ and $Z_{u \oplus v \oplus 0}$:

$$Z_{u' \oplus v \oplus 0} - Z_{u \oplus v \oplus 0} = C_{k'} - C_k + |v| \cdot (D_{k'} - D_k)$$

Since $C_k \leq C_{k'}$ and $D_k \leq D_{k'}$, it can be concluded that $Z_{u' \oplus v \oplus 0} \geq Z_{u \oplus v \oplus 0}$. This proves that the trip obtained by adding any sequence of required sites v at the end of trip k dominates the trip built by adding the same sequence at the end of trip k' . \square

The two theorems concern two trips k and k' with the same required sites. If $C_k \leq C_{k'}$ and $D_k \leq D_{k'}$ then Theorem 3.1 holds, and Theorem 3.2 too if k and k' share the same last site. If Theorem 3.2 holds for $k = (1, 2, 3)$ and $k' = (2, 1, 3)$ (for instance), $k \oplus v$ will dominate $k' \oplus v$ for any sequence v of sites, e.g., $(1, 2, 3, 4, 5)$ will dominate $(2, 1, 3, 4, 5)$. Hence, k' can be deleted immediately. However, if k and k' satisfy Theorem 3.1 but have distinct last sites, e.g. $(1, 2, 3)$ and $(3, 1, 2)$, k dominates k' but this is not necessarily the case for extensions like $(1, 2, 3, 4)$ and $(3, 1, 2, 4)$. So, Theorem 3.1 is applied only at the end, once all possible extensions of each trip have been generated.

Note that after the precomputation of K , it is no longer necessary to verify capacity and range constraints. The same trip generation procedure is also used in the set partitioning model described in Subsection 3.2.2.

3.4.2 Generation of the Arc-Set

The arc-set E' contains one arc (k, k') if trip-node k can be followed by trip-node k' in a complete path, e.g., if trip k can be followed by trip k' in a multitrip. A first condition is that k and k' visit distinct sites, i.e., $B_k \cap B_{k'} = \emptyset$. Theorem 3.3 demonstrates that it is not necessary to consider all possible trip orderings in a multitrip.

Theorem 3.3. *The cost of a multitrip is minimized by ordering its trips in non-decreasing order of mean trip duration, where the mean duration of a trip k is defined as its total duration D_k divided by its number of required nodes M_k .*

Proof. Consider a multitrip p which does not follow this rule. It contains two consecutive trips k and k' such that the mean duration of k is larger than the one of k' . Let Z_k^p be the sum of arrival times of trip k in multitrip p , t_0^k its starting time, and t_i the arrival time at site i (if trip k starts at t_0^k). The two trip costs can be written as follows:

$$Z_k^p = \sum_{i \in k} (t_0^k + t_i) = M_k \cdot t_0^k + \sum_{i \in k} t_i$$

and

$$Z_{k'}^p = \sum_{i \in k'} (t_0^{k'} + t_i) = M_{k'} \cdot t_0^{k'} + \sum_{i \in k'} t_i$$

As k is ordered before k' , we can take without loss of generality $t_0^k = 0$ and $t_0^{k'} = D_k$. Then the total cost is:

$$Z_k^p + Z_{k'}^p = \sum_{i \in k} t_i + M_{k'} \cdot D_k + \sum_{i \in k'} t_i$$

If k and k' are swapped to obtain a new multitrip p' which does not affect the trips before k or after k' , then $t_0^{k'} = 0$, $t_0^k = D_{k'}$, and the sum of arrival times for the two trips becomes:

$$Z_k^{p'} + Z_{k'}^{p'} = M_k \cdot D_{k'} + \sum_{i \in k} t_i + \sum_{i \in k'} t_i$$

The cost difference is:

$$(Z_k^p + Z_{k'}^p) - (Z_k^{p'} + Z_{k'}^{p'}) = M_{k'} \cdot D_k - M_k \cdot D_{k'}$$

But the initial assumption states that the mean duration of k' is less than the one of k :

$$\frac{D_k}{M_k} > \frac{D_{k'}}{M_{k'}}$$

Therefore:

$$(Z_k^p + Z_{k'}^p) - (Z_k^{p'} + Z_{k'}^{p'}) > 0$$

which shows that the total cost is reduced by swapping two consecutive trips that are not ordered in non-decreasing order of mean trip duration. \square

According to this theorem, E' can be limited to the arcs (k, k') such that $B_k \cap B_{k'} = \emptyset$ and $\frac{D_k}{M_k} < \frac{D_{k'}}{M_{k'}}$, giving a directed acyclic graph. If $\frac{D_k}{M_k} = \frac{D_{k'}}{M_{k'}}$ either (k, k') or (k', k) can be selected, provided no cycle is created. Moreover, by ordering the set K of trip-nodes in non-decreasing order of mean duration, a *topological sort* of G' is obtained, which simplifies the shortest path algorithm. Theorem 3.3 can be also applied to reduce solution space in the set partitioning model of Subsection 3.2.2, by adding the constraints:

$$\left(\frac{D_{k'}}{M_{k'}} - \frac{D_k}{M_k} \right) \cdot y_{kk'} \geq 0, \quad \forall k \in K, k' \in K \quad (3.43)$$

During the construction of G' , the following values are precomputed to ease calculations of lower bounds and dominance rules in the shortest path algorithm described in the next section: the maximum number of required sites per trip $M_{max} = \max\{M_k \mid k \in K\}$ and, in the original graph, the minimum cost of the arcs incident to the depot $w_{min}^0 = \min\{w_{0j} \mid j \in V'\}$, the minimum service time $s_{min} = \min\{s_i \mid i \in V'\}$, and the minimum cost of the arcs between two required sites $w_{min} = \min\{w_{ij} \mid i, j \in V'\}$.

3.5 Shortest Path Algorithm

Like in Feillet *et al.* (2004) and Azi *et al.* (2007), an extension of Bellman-Ford algorithm with multiple labels per node is used to solve the resource-constrained shortest path problem. In this version, shown in Algorithm 3.1, all paths are implicitly elementary since the graph

is directed and acyclic: each partial path undergoes a single extension towards its feasible successors.

The algorithm numbers the trip-nodes from 1 to $|K|$ in topological order. It defines for each trip-node $k \in K$ a set Ω_k of non-dominated partial paths ending at this trip-node and updates the best solution p^* when it is improved. This solution is initialized in line 1 by the best result of two heuristics, called in the procedure *Initial_solution* described in the sequel.

Each path p is identified by a label $R_p = (C_p, D_p, B_p, M_p)$. For the sake of simplicity, the same notation is used as for the trips, but with a path index p instead of a trip index k : C_p is the total cost of the path (sum of arrival times), D_p its duration, B_p the set of visited required sites and $M_p = |B_p| = \sum_{k \in p} M_k$ its cardinality.

In most shortest path problems on DAGs, feasible paths go from the first node (without predecessors) to the last (without successors). Here, other nodes may lack successors or predecessors (see Figure 3.2) and a complete path may begin and end at any node. So, for each trip-node k , the algorithm initializes Ω_k with the partial path reduced to trip k , with a label $R_k = (C_k, D_k, B_k, M_k)$ (lines 2 to 4). Then, it inspects each trip k in topological order (line 5) and each partial path $p \in \Omega_k$ (line 6) and tries to extend it by adding one trip k' at the end. The candidates correspond to the arcs $(k, k') \in E'$. The existence of arc (k, k') ensures by construction that trips k and k' have no common sites but some sites of k can be visited by earlier trips on the path: this case is avoided by checking that $B_p \cap B_k = \emptyset$. The set of feasible successors for path p , Θ_p , is computed in line 7 via the procedure *Successors*, which can be accelerated using the condition $M_p + M_k \leq n$.

For each successor $k' \in \Theta_p$, a tentative label $R_{p \oplus k'}$ can be computed for the new path $p \oplus k'$, using the system of equations (3.44). Silva *et al.* (2012) have defined similar equations to concatenate sequences of customers in a local search for the Cumulative TSP. The equations are adapted here to concatenate one trip-node k' at the end of a path p . In that case the last site of the path p and the first site of the trip-node k' is the depot, which allows to simplified some equations.

$$\begin{aligned}
 C_{p \oplus k} &= C_p + D_p \cdot M_k + C_k \\
 D_{p \oplus k} &= D_p + D_k \\
 B_{p \oplus k} &= B_p \cup B_k \\
 M_{p \oplus k} &= M_p + M_k
 \end{aligned}
 \tag{3.44}$$

Algorithm 3.1 – Shortest path algorithm for the mt-CCSVRP

```

1: Initial_solution ( $p^*$ )
2: for  $k \leftarrow 1$  to  $|K|$  do
3:    $\Omega_k \leftarrow \{k\}$ 
4: end for
5: for  $k \leftarrow 1$  to  $|K|$  do
6:   for each partial path  $p$  in  $\Omega_k$  do
7:      $\Theta_p \leftarrow \text{Successors}(K, p)$ 
8:     for each  $k' \in \Theta_p$  such that  $LB_{p \oplus k'} < Z_{p^*}$  do
9:       if ( $M_{p \oplus k'} = n$ ) and ( $C_{p \oplus k'} < Z_{p^*}$ ) then
10:         $p^* \leftarrow p \oplus k'$ 
11:       end if
12:        $u \leftarrow k + 1$ 
13:       while ( $u \leq |K|$ ) and (no path in  $\Omega_u$  dominates  $p \oplus k'$ ) do
14:         remove from  $\Omega_u$  the paths dominated by  $p \oplus k'$ 
15:          $u \leftarrow u + 1$ 
16:       end while
17:       if  $u > |K|$  then  $\Omega_{k'} \leftarrow \Omega_{k'} \cup \{p \oplus k'\}$  endif
18:     end for
19:   end for
20: end for
    
```

$LB_{p \oplus k'}$ is the lower bound of the best complete path which extends $p \oplus k'$. It is computed as the maximum of the five bounds presented in the next subsection. The new partial path is discarded if this bound is not smaller than the best cost Z_{p^*} . Otherwise, if $p \oplus k'$ is complete and outperforms the current best solution (line 9), p^* is updated (lines 9–11).

Most shortest path algorithms on DAGs compare paths with the same extremities and apply dominance tests only to $\Omega_{k'}$ when one arc (k, k') is added to a partial path ending at node k . A singularity of the mt-CCSVRP problem is that dominated or dominating paths may exist in sets $\Omega_{k+1}, \Omega_{k+2}, \dots, \Omega_{|K|}$. For instance, the simplest dominance rule presented in the next subsection states that path p dominates p' if $C_p \leq C_{p'}$, $D_p \leq D_{p'}$ and $B_{p'} \subseteq B_p$. Consider a partial path p ending at trip-node k , with $B_p = \{2, 3, 4\}$, and the three next nodes such that $B_{k+1} = \{2, 3, 7\}$, $B_{k+2} = B_{k'} = \{7, 8\}$ and $B_{k+3} = \{3, 8\}$. In that case, B_{k+1} and B_{k+3} are both included in $B_{p \oplus k'} = \{2, 3, 4, 7, 8\}$. Dominated labels may exist in Ω_{k+1} and Ω_{k+3} , not only in $\Omega_{k'}$, and paths $k+1$ and $k+3$ may be dominated by path $p \oplus k'$, even

if they do not have the same extremities.

This is why the *while* loop of lines 13–16 browses each trip-node u after k , even if the new partial path is complete, to eliminate the existing paths which are dominated. This loop stops when all trip-nodes have been inspected or as soon as an existing path dominates the new one. If the latter is not dominated (line 17), it is added to $\Omega_{k'}$.

3.5.1 Lower Bounds

Given a partial path p , five lower bounds of the best complete path which extends p are described.

Recall that the following values have been computed during the construction of G' : the maximum number of required sites per trip $M_{max} = \max\{M_k \mid k \in K\}$, the minimum cost of the arcs incident to the depot $w_{min}^0 = \min\{w_{0j} \mid j \in V'\}$, the minimum service time $s_{min} = \min\{s_i \mid i \in V'\}$, and the minimum cost of the arcs between two required sites $w_{min} = \min\{w_{ij} \mid i, j \in V'\}$.

Lower Bound 3.1. Consider a partial path p and the best complete path $p \oplus \bar{p}$ that extends it, i.e., \bar{p} is the optimal way of visiting the remaining required sites after p . According to Equation (3.35), the cost of the complete path is $C_{p \oplus \bar{p}} = C_p + D_p \cdot M_{\bar{p}} + C_{\bar{p}}$. As \bar{p} is unknown and $C_{\bar{p}} \geq 0$, $C_{\bar{p}}$ can be set to 0 and, since $M_{\bar{p}} = n - M_p$, get a first lower bound $LB_p^1 = C_p + D_p \cdot (n - M_p)$ for the cost of $p \oplus \bar{p}$. This simple bound can be computed in $O(1)$.

Lower Bound 3.2. LB_p^1 can be improved by serving the remaining sites in a single trip k . Equation (3.2) defined the cost of a trip $k = (0, 1, 2, \dots, M_k, 0)$ as $C_k = M_k \cdot \beta \cdot \sum_{i=1}^{M_k} s_i + \sum_{i=0}^{M_k-1} (M_k - i) \cdot (s_i + w_{i,i+1})$. Using the minima defined at the end of Section 3.4, the cost of the first arc w_{01} can be set to w_{min}^0 , the cost of each arc linking two required sites $w_{i,i+1}$ to w_{min} , and each service time s_i to s_{min} . After simplification, a second lower bound is obtained: $LB_p^2 = LB_p^1 + (w_{min}^0 + \beta \cdot s_{min} \cdot (n - M_p)) \cdot (n - M_p) + \frac{1}{2} \cdot (w_{min} + s_{min}) \cdot (n - M_p) \cdot (n - M_p - 1)$. This bound dominates LB_p^1 and, as w_{min} , w_{min}^0 and s_{min} are precomputed, it is also computable in constant time.

Lower Bound 3.3. The previous bound can be refined by using M_{max} to approximate the minimum number of remaining trips m and a lower bound for their cost and duration.

First, the minimum number of remaining trips can be computed as $m = \left\lceil \frac{(n - M_p)}{M_{max}} \right\rceil$.

Then, lower bounds for the cost and the duration of the first $(m - 1)$ trips, which visit at most M_{max} required sites, can be written as $LB_c = (w_{min}^0 + \beta \cdot s_{min} \cdot M_{max}) \cdot M_{max} + \frac{1}{2} \cdot (w_{min} + s_{min}) \cdot (M_{max} - 1) \cdot M_{max}$ and $LB_d = 2 \cdot w_{min}^0 + w_{min} \cdot (M_{max} - 1) + (1 + \beta) \cdot s_{min} \cdot M_{max}$, respectively.

As the last trip visits the $(n - M_p - M_{max} \cdot (m - 1))$ remaining sites, its cost can be lower bounded by $LB_c^{last} = (w_{min}^0 + \beta \cdot s_{min} \cdot (n - M_p - M_{max} \cdot (m - 1))) \cdot (n - M_p - M_{max} \cdot (m - 1)) + \frac{1}{2} \cdot (w_{min} + s_{min}) \cdot (n - M_p - M_{max} \cdot (m - 1)) \cdot (n - M_p - M_{max} \cdot (m - 1) - 1)$.

Finally, the lower bound can be computed as $LB_p^3 = LB_p^1 + (m - 1) \cdot (LB_c + LB_d \cdot (n - M_p - M_{max} \cdot (m - 1))) + \frac{1}{2} \cdot LB_d \cdot M_{max} \cdot (m - 1) \cdot (m - 2) + LB_c^{last}$. It dominates LB_p^1 and LB_p^2 and can be computed in $O(1)$ too. It can be seen that when the number of remaining nodes $n - M_p$ is lower than M_{max} , the minimum number of remaining trips becomes $m = 1$ and then $LB_p^3 = LB_p^2$.

Lower Bound 3.4. This lower bound approximates the arrival times to the remaining sites. Consider a list π_p containing these sites i in non-decreasing order of distance to the depot w_{0i} . The arrival time at the i -th site in π_p is approximated as $c'_i = (i \cdot \beta + i - 1) \cdot s_{min} + w_{0, \pi_p(i)} \cdot \rho_{i-1} + d_i$, where $\pi_p(i)$ is the i -th site in π_p , $d_i = (w_{\pi_p(i-1), 0} + w_{0, \pi_p(i)}) \cdot \rho_{i-1} + d_{i-1}$, $d_0 = D_p$, $\pi_p(0) = 0$, $\rho_0 = 0$, and ρ_i a binary indicator equal to 1 if i is multiple of M_{max} which means that a trip would be terminated at node i . Finally, the lower bound is computed as $LB_p^4 = C_p + \sum_{i=1}^{n-M_p} c'_i$.

Note that c'_i is always smaller than the real arrival time at the i -th remaining site if the triangle inequality is satisfied, because a) $w_{0, \pi_p(i)} \leq \sum_{j=1}^i w_{\pi_p(j-1), \pi_p(j)}$ for any order π'_p is a lower bound of the flight time to reach the i -th site in π_p , and b) d_i is a lower bound of the duration of trips before that site.

Lower Bound 3.5. In this modification of LB_p^4 , the arrival time at the i -th site in π_p is approximated by $c''_i = (i \cdot \beta + i - 1) \cdot s_{min} + (i - \tau_i) \cdot w_{min} + d''_i$, where $d''_i = (w_{\pi_p(2 \cdot \tau_i), 0} + w_{0, \pi_p(2 \cdot \tau_i + 1)}) \cdot \rho_{i-1} + d''_{i-1}$, $d''_0 = D_p + w_{0, \pi_p(1)}$, and $\tau_i = \tau_{i-1} + \rho_i$ is the minimum number of trips until i -th site ($\tau_0 = 1$). The binary indicator ρ_i is equal to 1 if, given the last position j such that $\rho_j = 1$, we have $w_{\pi_p(j), 0} + (j - i) \cdot w_{min} + w_{0, \pi_p(i)} > L_{max}$ or $(i - j + 1) = M_{max}$. Compared with LB_p^4 , $w_{0, \pi_p(i)}$ in c' is replaced by $\sum_{j=1}^i w_{min} = i \cdot w_{min}$,

which is smaller than $\sum_{j=1}^i w_{\pi'_p(j-1), \pi'_p(j)}$ for any order π'_p . The bound is computed as $LB_p^5 = C_p + \sum_{i=1}^{n-M_p} c_i''$.

The two last lower bounds can be computed in $O(n)$. The following strategy is applied in line 8 of Algorithm 3.1. As the bounds $LB_{p \oplus k}^1$ and $LB_{p \oplus k}^2$ are dominated by $LB_{p \oplus k}^3$, they are not used. $LB_{p \oplus k}^3$ is first computed in $O(1)$. If $LB_{p \oplus k}^3 \geq Z_{p^*}$, the new partial path $p \oplus k$ can be discarded. Otherwise, $LB_{p \oplus k}^4$ and $LB_{p \oplus k}^5$ are computed in $O(n)$ and the best one is compared with Z_{p^*} .

3.5.2 Dominance Rules

Section 3.4 describes two dominance rules to eliminate trip-nodes when building the auxiliary network. Four other rules are now introduced to discard partial paths in the shortest path algorithm. The first one (Theorem 3.4) extends Theorem 3.1 to compare two partial paths p and p' instead of two trips k and k' . It simply says that path p dominates path p' if it is not more costly, it does not last longer, and all sites visited by path p' are also visited by path p .

Theorem 3.4. *A partial path p dominates a partial path p' if $C_p \leq C_{p'}$, $D_p \leq D_{p'}$, and $B_{p'} \subseteq B_p$.*

Proof. Let p and p' be two partial paths with labels $R_p = (C_p, D_p, B_p, M_p)$ and $R_{p'} = (C_{p'}, D_{p'}, B_{p'}, M_{p'})$, respectively. Let $p \oplus \bar{p}$ and $p' \oplus \bar{p}'$ denote the best complete paths extending p and p' , $C_{\bar{p}}$ and $C_{\bar{p}'}$ the costs of \bar{p} and \bar{p}' , and $M_{\bar{p}} = n - M_p$ and $M_{\bar{p}'} = n - M_{p'}$ the number of sites that they visit. From Equation (3.35), the costs of the two complete paths are:

$$Z_{p \oplus \bar{p}} = C_p + C_{\bar{p}} + D_p \cdot (n - M_p)$$

and

$$Z_{p' \oplus \bar{p}'} = C_{p'} + C_{\bar{p}'} + D_{p'} \cdot (n - M_{p'})$$

It can be proved that $Z_{p \oplus \bar{p}} \leq Z_{p' \oplus \bar{p}'}$ by showing that each term in the first cost does not exceed the corresponding term in the second one. From the hypotheses, it is known that $C_p \leq C_{p'}$ and $D_p \leq D_{p'}$. Moreover, as $B_{p'} \subseteq B_p$, then $M_{p'} = |B_{p'}| \leq M_p = B_p \iff$

$n - M_p \leq n - M_{p'}$. Finally, as $B_{\bar{p}} \subseteq B_{\bar{p}'}$ and travel times satisfy the triangle inequality, it can be confirmed that $C_{\bar{p}} \leq C_{\bar{p}'}$. \square

The remaining rules are deduced from Theorem 3.4 by computing lower bounds for the costs of the two complete paths.

Theorem 3.5. *A partial path p dominates a partial path p' if $B_{p'} \subseteq B_p$ and $C_p + D_p \cdot (n - M_{p'}) \leq C_{p'} + D_{p'} \cdot (n - M_{p'})$.*

Proof. Consider again two partial paths p and p' and the best complete paths $p \oplus \bar{p}$ and $p' \oplus \bar{p}'$ which extend them. According to the hypotheses:

$$C_p + D_p \cdot (n - M_{p'}) \leq C_{p'} + D_{p'} \cdot (n - M_{p'})$$

In the previous proof it can be seen that $n - M_p \leq n - M_{p'}$, which implies:

$$C_p + D_p \cdot (n - M_p) \leq C_{p'} + D_{p'} \cdot (n - M_{p'})$$

As $C_{\bar{p}} \leq C_{\bar{p}'}$, we can write:

$$C_p + D_p \cdot (n - M_p) + C_{\bar{p}} \leq C_{p'} + D_{p'} \cdot (n - M_{p'}) + C_{\bar{p}'}$$

As the two sides of this inequality correspond to $Z_{p \oplus \bar{p}}$ and $Z_{p' \oplus \bar{p}'}$ according to Equation (3.35), it can be concluded that path p dominates path p' . \square

Theorem 3.6 below can be deduced from the proof of Theorem 3.5. It has the same statement, except that $M_{p'}$ is replaced by M_p in the left-hand side of the inequality. This theorem dominates the two previous ones.

Theorem 3.6. *A partial path p dominates a partial path p' if $B_{p'} \subseteq B_p$ and $C_p + D_p \cdot (n - M_p) \leq C_{p'} + D_{p'} \cdot (n - M_{p'})$.*

It can be verified that if Theorem 3.4 holds, Theorems 3.5 and 3.6 too. And, if Theorem 3.5 holds, from the previous proofs $n - M_p \leq n - M_{p'}$ is satisfied, which implies that $C_p + D_p \cdot (n - M_p) \leq C_p + D_p \cdot (n - M_{p'})$. Then, from the statements of Theorem 3.6

$C_p + D_p \cdot (n - M_p) \leq C_p + D_p \cdot (n - M_{p'}) \leq C_{p'} + D_{p'} \cdot (n - M_{p'})$, which implies that if Theorem 3.5 holds, Theorem 3.6 too.

The last dominance rule, specified by Theorem 3.7, improves and generalizes the previous ones by including a lower bound on the cost to visit the sites visited by path p but not by path p' , i.e., the sites in $B_p \setminus B_{p'}$.

Theorem 3.7. *A partial path p dominates a partial path p' if $B_{p'} \subseteq B_p$ and $C_p + D_p \cdot (n - M_p) \leq C_{p'} + D_{p'} \cdot (n - M_{p'}) + w_{min} \cdot (M_p - M_{p'}) \cdot (n - M_p) + \frac{1}{2} \cdot w_{min} \cdot (M_p - M_{p'}) \cdot (M_p - M_{p'} + 1)$.*

Proof. Like for Theorem 3.4, $p \oplus \bar{p}$ and $p' \oplus \bar{p}'$ are the best complete paths that extend p and p' , with their costs defined by Equation (3.35):

$$Z_{p \oplus \bar{p}} = C_p + C_{\bar{p}} + D_p \cdot (n - M_p)$$

and

$$Z_{p' \oplus \bar{p}'} = C_{p'} + C_{\bar{p}'} + D_{p'} \cdot (n - M_{p'})$$

As $M_{p'} \leq M_p$, path \bar{p}' has to visit $M_p - M_{p'}$ more sites than \bar{p} and:

$$C_{\bar{p}'} \geq C_{\bar{p}} + \frac{1}{2} \cdot w_{min} \cdot (M_p - M_{p'}) \cdot (M_p - M_{p'} + 1) + w_{min} \cdot (M_p - M_{p'}) \cdot (n - M_p)$$

where $\frac{1}{2} \cdot w_{min} \cdot (M_p - M_{p'}) \cdot (M_p - M_{p'} + 1)$ is a lower bound for the cost of visiting the $M_p - M_{p'}$ additional sites, and $w_{min} \cdot (M_p - M_{p'}) \cdot (n - M_p)$ is a lower bound for the sum of arrival time shifts at the last $(n - M_p)$ sites of \bar{p}' , due to the travel times to the first $M_p - M_{p'}$ sites visited by this path.

According to the hypotheses:

$$\begin{aligned} C_p + D_p \cdot (n - M_p) &\leq C_{p'} + D_{p'} \cdot (n - M_{p'}) + w_{min} \cdot (M_p - M_{p'}) \cdot (n - M_p) \\ &\quad + \frac{1}{2} \cdot w_{min} \cdot (M_p - M_{p'}) \cdot (M_p - M_{p'} + 1) \end{aligned}$$

If $C_{\bar{p}}$ is added to both sides, the following inequality is obtained:

$$\begin{aligned} C_p + D_p \cdot (n - M_p) + C_{\bar{p}} &\leq C_{p'} + D_{p'} \cdot (n - M_{p'}) + \\ C_{\bar{p}} + w_{min} \cdot (M_p - M_{p'}) \cdot (n - M_p) &+ \frac{1}{2} \cdot w_{min} \cdot (M_p - M_{p'}) \cdot (M_p - M_{p'} + 1) \end{aligned}$$

As shown before, the second line is not greater than $C_{\bar{p}'}$. So:

$$C_p + D_p \cdot (n - M_p) + C_{\bar{p}} \leq C_{p'} + C_{\bar{p}'} + D_{p'} \cdot (n - M_{p'})$$

which means that $Z_{p \oplus \bar{p}} \leq Z_{p' \oplus \bar{p}'}$ and p dominates p' . □

Note that when $w_{min} = 0$, Theorem 3.7 reduces to Theorem 3.6 and, when $B_p = B_{p'}$, Theorems 3.5, 3.6 and 3.7 give the same result. Since the last dominance rule dominates the others, it is the only one actually used in Algorithm 3.1.

3.5.3 Initial Solutions

Two constructive heuristics are executed at the beginning of the algorithm. The best solution obtained is used to initialize p^* .

First heuristic. Starting with an empty path p , the first heuristic sweeps the trip-nodes in topological order (non-decreasing mean duration) and adds the incumbent trip-node k at the end of p if its required sites are not already visited by the path ($B_p \cap B_k = \emptyset$). The heuristic ends when path p is complete (all required sites are covered).

Second heuristic. According to the system of equations (3.44) and the first lower bound, the expression $C_p + C_k + D_p \cdot M_k + (D_k + D_p) \cdot (n - M_k - M_p)$ gives a lower bound for the cost of a shortest path beginning with the path $p \oplus k$. Starting again from an empty path, the second heuristic adds at each iteration the feasible trip-node k with the smallest lower bound value.

3.6 Computational Experiments

Computational experiments have been conducted to evaluate and compare the performances of the two mathematical models and the shortest path approach. The instances used are presented in Subsection 3.6.1. Subsection 3.6.2 discusses the impact of the dominance rules of Theorem 3.1 and Theorem 3.2 on the auxiliary network generation phase. In Subsection 3.6.3, the two mathematical problems are solved using a MILP solver (CPLEX) and compared with the shortest path algorithm on instances with $n = 20$ required sites. The exact approach is tested on larger instances with up to $n = 40$ in Subsection 3.6.4.

3.6.1 Implementation and Instances

The shortest path algorithm is coded in Microsoft Visual Studio C++ 2010 Professional while the mathematical models are solved using IBM ILOG CPLEX Optimization Studio version 12.5. The tests are run on an Intel Core i5 PC at 2.50 GHz with 4 GB of RAM and Windows 7 Professional.

The 14 classical CVRP instances from Christofides *et al.* (1979) (CMT instances) are taken as a basis. These benchmark problems with 50 to 199 customers include 7 instances with service times (instances 6 to 10, plus 13 and 14). Instances 6 to 10 with 50, 75, 100, 150 and 199 customers are selected to derive 138 mt-CCSVRP instances, partitioned in four groups containing $n = 20, 25, 30$ and 40 sites, respectively. To build a set of instances with n sites, each CMT instance is taken as basis, a single vehicle is considered, and different values of L_{max} are chosen. The depot and a subset of n required sites is kept.

The values for L_{max} are multiples of 10, between a minimum feasible value (which depends on the CMT instance) and 120. The nodes selected are the depot and the n first required sites i which can be reached by the vehicle, i.e., the ones satisfying $2 \cdot w_{0i} \leq L_{max}$. Note that the selected nodes are not necessarily consecutive in the original CMT instance. The coordinates of these sites, their demands and service times, and the vehicle capacity are the same as in the original CMT file. The traveling time w_{ij} on each edge (i, j) is equal to the Euclidean distance, computed as a double-precision real number.

Finally, the loading time for each trip k in the auxiliary network is computed as $\mu_k = \beta \cdot \sum_{i \in k} s_i$, where s_i is the service time at site i and $\beta = 0.2$.

Instances 13 and 14 (100 and 120 customers) are discarded. Contrary to the other CMT instances, their clustered nodes induce a huge number of trips. For example, instance 13 with $L_{max} = 70$ and a selection of $n = 20$ required sites has already more than 20 millions feasible trips and the auxiliary network requires more than 24 hours to be generated.

3.6.2 Impact of Dominance Rules on Auxiliary Network Construction

As explained in Section 3.4, Theorem 3.2 can be applied to delete dominated trips as soon as they are generated, thus avoiding useless attempts to extend them, while Theorem 3.1 is used only at the end, once all extensions have been done. Two strategies are compared

to evaluate the impact of these rules. Strategy 1 consists in using Theorem 3.1 only, while Strategy 2 applies both theorems. The two strategies end by sorting the resulting set K of non-dominated trips in non-decreasing order of mean trip duration, following Theorem 3.3, and building the arc-set E' . Experiments are conducted on 20 instances with $n = 20$ sites and four L_{max} values: the minimum feasible multiple of 10, plus 70, 100 and 120.

Table 3.1 summarizes the results. The two first columns indicate the original CMT instance and the vehicle range L_{max} , which defines the instance data. Column $|K|$ reports the number of non-dominated trips obtained at the end. This number, identical for both strategies, is the number of trip-nodes in the auxiliary graph G' for the shortest path algorithm. Column M_{max} mentions the maximum number of required sites per trip, to see the impact of the parameter L_{max} and have an idea of the size of the solution space. The last four columns provide for each strategy the number of generated trips (columns $|K_g|$) and the total time in seconds to build the auxiliary graph (columns Time_g).

The table shows that Strategy 1 (Theorem 3.1 alone) leads to a strong reduction between the $|K_g|$ trips generated and the final $|K|$ non-dominated trips. This reduction ranges from 57% (for CMT₀₆) to 95% (for CMT₀₉), with an average of 77%. Clearly, this reduction gets stronger as L_{max} increases but, in parallel, the running time, the number of generated trips and the number of non-dominated trips grow very quickly. The maximum number of sites per trip (M_{max}) is roughly proportional to L_{max} .

The application of Theorem 3.2 in the second strategy brings additional savings. Compared with Strategy 1, the number of generated trips $|K_g|$ is reduced by up to 64% (for CMT₀₉) and by 27% on average. A decrease in running time of up to 81% (CMT₀₉ again) and 74% on average can also be observed: the extra time spent in applying Theorem 3.2 is compensated by a much smaller number of trip extensions.

Among the instances derived from CMT₀₈, the number of non-dominated trip-nodes $|K|$ is larger for $L_{max} = 40$ than for $L_{max} = 70$. This result is not abnormal since the n sites i extracted, which are the first to satisfy $2 \cdot w_{0i} \leq L_{max}$, are not necessarily the same for different L_{max} values.

3.6.3 Comparing the Mathematical Models and the Exact Method

Table 3.2 compares the two mathematical models and the resource-constrained shortest path algorithm, reusing the 20 instances with $n = 20$ of the previous section. The three first

Table 3.1: Impact of dominance rules on auxiliary network generation for instances with $n = 20$ sites.

Instance	L_{max}	$ K $	M_{max}	Strategy 1		Strategy 2	
				$ K_g $	Time _g	$ K_g $	Time _g
CMT ₀₆	50	160	4	374	0.00	371	0.00
CMT ₀₆	70	504	6	1 568	0.00	1 473	0.00
CMT ₀₆	100	9 023	8	67 746	6.41	43 011	4.84
CMT ₀₆	120	45 599	10	736 122	1 284.73	301 130	285.83
CMT ₀₇	40	173	4	424	0.00	422	0.00
CMT ₀₇	70	596	6	1 956	0.01	1 774	0.02
CMT ₀₇	100	9 071	8	71 704	8.21	42 929	5.03
CMT ₀₇	120	38 047	8	621 252	1 129.94	246 662	230.99
CMT ₀₈	40	263	5	726	0.00	698	0.00
CMT ₀₈	70	192	5	472	0.00	461	0.00
CMT ₀₈	100	3 321	8	16 778	0.32	12 743	0.21
CMT ₀₈	120	17 720	9	158 492	43.43	87 016	20.90
CMT ₀₉	30	209	5	524	0.00	510	0.00
CMT ₀₉	70	636	6	2 408	0.01	2 134	0.01
CMT ₀₉	100	10 782	9	97 362	13.32	54 665	6.94
CMT ₀₉	120	56 866	11	1 030 402	2 199.37	369 435	416.75
CMT ₁₀	30	199	6	568	0.00	529	0.00
CMT ₁₀	70	587	6	2 192	0.01	1 933	0.01
CMT ₁₀	100	9 220	9	72 940	7.67	43 143	4.76
CMT ₁₀	120	51 047	11	774 642	1 360.43	307 339	309.06

columns recall the CMT instance of origin, the range L_{max} and the number of non-dominated trip-nodes $|K|$. For the exact method are reported the optimal solution value Z_p^* and the total running time in seconds (including the network construction using Strategy 2). For each mathematical model are given the time in seconds to find an optimal solution (limited to two hours), the percentage gap between the best integer solution found and the best lower bound reported by CPLEX (Gap_c) and the percentage gap between the best integer solution found and the optimal solution (Gap). Null gaps are highlighted in boldface while dashes indicate memory overflows in CPLEX.

The results show that the shortest path approach is much more efficient than a direct

Table 3.2: Results of the mt-CCSVRP solution approaches on small instances with $n = 20$ sites.

Instance	L_{max}	$ K $	Shortest Path		Flow-based model			Set partitioning model		
			Z_p^*	Time	Time	Gap _c	Gap	Time	Gap _c	Gap
CMT ₀₆	50	160	5 474.61	0.12	7 200.00	5.86	0.00	2 092.72	0.00	0.00
CMT ₀₆	70	504	5 529.37	0.12	7 200.00	8.91	0.00	7 200.00	74.44	0.00
CMT ₀₆	100	9 023	5 037.30	7.76	7 200.00	8.52	0.81	–	–	–
CMT ₀₆	120	45 599	4 942.38	305.42	7 200.00	6.43	1.09	–	–	–
CMT ₀₇	40	173	4 562.43	0.03	7 200.00	2.36	0.00	2 683.07	0.00	0.00
CMT ₀₇	70	596	5 536.46	0.03	7 200.00	10.63	0.51	7 200.00	75.19	3.76
CMT ₀₇	100	9 071	5 065.22	8.74	7 200.00	7.56	0.36	–	–	–
CMT ₀₇	120	38 047	5 031.30	252.98	7 200.00	6.90	0.48	–	–	–
CMT ₀₈	40	263	4 616.32	0.10	7 200.00	8.00	0.00	7 200.00	27.13	1.51
CMT ₀₈	70	192	7 141.29	0.10	7 200.00	10.07	0.00	2 650.01	0.00	0.00
CMT ₀₈	100	3 321	5 272.93	1.07	7 200.00	8.53	0.56	–	–	–
CMT ₀₈	120	17 720	5 143.83	25.07	7 200.00	5.43	0.00	–	–	–
CMT ₀₉	30	209	4 112.95	0.16	7 200.00	6.37	0.08	6 612.31	0.00	0.00
CMT ₀₉	70	636	5 571.45	0.16	7 200.00	9.68	0.87	7 200.00	64.54	5.77
CMT ₀₉	100	10 782	4 958.99	11.43	7 200.00	7.69	1.36	–	–	–
CMT ₀₉	120	56 866	4 821.66	441.59	7 200.00	3.66	0.00	–	–	–
CMT ₁₀	30	199	3 953.31	0.03	7 200.00	1.18	0.00	2 341.46	0.00	0.00
CMT ₁₀	70	587	5 627.16	0.03	7 200.00	7.73	0.00	7 200.00	65.33	6.15
CMT ₁₀	100	9 220	5 035.16	8.60	7 200.00	8.75	1.45	–	–	–
CMT ₁₀	120	51 047	4 926.24	334.80	7 200.00	6.40	0.85	–	–	–

resolution of the two mixed integer linear programs: it solves all instances to optimality in 69 seconds on average and in 441 seconds (7.4 minutes) in the worst case. Nevertheless, the number of trip-nodes in the auxiliary network and the total running time grow quickly with L_{max} . A comparison with the previous table reveals that the fraction of time devoted to the auxiliary network construction increases in parallel, reaching 92.3% for the last instance (309.06 seconds out of 334.80).

Although the instances considered are small (20 required sites), the flow-based model retrieves only nine of the optima found by the shortest path algorithm and CPLEX has no proof of optimality since its branch-and-cut is always interrupted after two hours. The

CPLEX gaps (Gap_c) varies between 1 and 10% but is not sensitive to the parameter L_{max} .

The set partitioning formulation behaves differently due to the large number of columns: $|K|$ binary variables are used to select the trips, plus $O(|K|^2)$ to enforce the dominance relation for the order of these trips in the multitrip. The five smaller instances with at most 209 trip-nodes are solved to optimality. The time limit is reached and poor integer solutions are returned for 5 other instances with up to 636 trip-nodes. The 10 remaining instances from 3 321 trip-nodes onwards raise an “out of memory” error.

Contrary to the other instances, on the instances derived from CMT₀₈, the set partitioning formulation obtains an optimal solution when $L_{max} = 70$ but not for $L_{max} = 40$. These results can be explained by the number of non-dominated trip-nodes, which is larger in the instance with $L_{max} = 40$.

3.6.4 Test of the Shortest Path Algorithm on Larger Instances

The previous subsections have shown that the shortest path approach is fast enough for $n = 20$ (one minute on average) to solve larger instances. Tables 3.3, 3.4 and 3.5 respectively gather the results for 25, 30 and 40 required sites. Most column headers have been already defined. The number of trips generated $|K_g|$ and the final number of non-dominated trips $|K|$ come again from Strategy 2. The time to generate the auxiliary network ($Time_g$) is now distinguished from the time to determine the shortest path ($Time_s$), and it is provided the percentage gap between the initial solution and the optimum (Gap_0). Concerning vehicle range L_{max} , all multiples of 10 between the minimum feasible value and 120 are considered for $n \in \{25, 30\}$. For $n = 40$, L_{max} has been limited to 100 otherwise the network is too large.

The results in Tables 3.3, 3.4 and 3.5 confirm the impact of the dominance rule of Theorem 3.1. If $|K_g|$ and $|K|$ are compared, the number of trip-nodes in the auxiliary network is reduced by 59% to 82% for instances with $n = 25$ sites, 60% to 85% for $n = 30$, and 66% to 82% for $n = 40$. These percentages are not significantly affected by the number of sites but tend to increase with L_{max} . For most instances, a greater value of $|K_g|$ or $|K|$ implies a greater value of the maximum number of sites per trip M_{max} , but the latter can stagnate because of capacity constraints, see for example instance CMT₀₇ in Table 3.4.

The two heuristics used at the beginning yield very good solutions on average, since Gap_0 varies between 0% and 12.7%, with an average value of 4.31%. Nevertheless, a better initial solution does not always imply a faster resolution. Consider in Table 3.5 instance CMT₀₇

Table 3.3: Results of the shortest path approach for the mt-CCSVRP for instances with $n = 25$ sites.

Instance	L_{max}	$ K_g $	$ K $	M_{max}	Time $_g$	Gap $_0$	Z_{p^*}	Time $_s$
CMT ₀₆	50	1 863	657	6	0.01	3.87	7 791.63	14.20
CMT ₀₆	60	932	377	5	0.00	0.12	9 000.46	3.77
CMT ₀₆	70	2 170	766	6	0.01	5.93	8 521.20	2.57
CMT ₀₆	80	8 220	2 502	7	0.08	2.26	7 869.37	1.71
CMT ₀₆	90	29 318	7 796	8	0.87	0.00	7 753.51	12.21
CMT ₀₆	100	97 792	22 301	8	2.49	0.50	7 678.77	74.65
CMT ₀₆	110	311 092	62 545	9	212.27	0.78	7 617.17	348.08
CMT ₀₆	120	930 583	162 971	10	2 041.07	0.90	7 578.53	1 455.52
CMT ₀₇	40	1 532	537	6	0.00	4.70	6 874.75	1.45
CMT ₀₇	50	1 348	511	6	0.00	4.47	7 606.16	5.10
CMT ₀₇	60	2 350	784	6	0.01	5.89	8 371.39	23.01
CMT ₀₇	70	4 057	1 155	6	0.03	7.04	8 929.86	39.09
CMT ₀₇	80	6 264	1 886	6	0.05	4.30	8 750.41	31.00
CMT ₀₇	90	19 887	5 227	7	0.43	5.83	8 144.82	12.27
CMT ₀₇	100	59 519	13 661	8	3.78	2.16	7 956.53	26.76
CMT ₀₇	110	162 762	32 566	8	50.68	2.77	7 909.23	114.44
CMT ₀₇	120	410 258	72 477	9	409.20	2.55	7 909.23	467.83
CMT ₀₈	40	924	355	5	0.00	3.55	7 367.81	4.21
CMT ₀₈	50	3 500	1 020	7	0.02	3.46	7 461.80	15.16
CMT ₀₈	60	2 477	784	6	0.01	11.43	8 481.42	18.32
CMT ₀₈	70	997	380	5	0.00	8.98	10 507.82	16.16
CMT ₀₈	80	2 125	729	6	0.01	4.56	9 042.59	0.28
CMT ₀₈	90	6 905	2 120	6	0.06	3.48	8 565.82	0.90
CMT ₀₈	100	21 058	5 656	8	0.48	1.19	8 287.96	3.25
CMT ₀₈	110	60 715	14 389	9	3.84	2.25	8 087.34	14.81
CMT ₀₈	120	169 345	35 786	9	51.37	1.42	8 043.47	80.84
CMT ₀₉	30	821	334	5	0.00	2.59	6 141.02	1.32
CMT ₀₉	40	1 026	385	6	0.00	0.22	6 837.04	1.32
CMT ₀₉	50	2 443	808	6	0.01	6.96	7 308.46	5.09
CMT ₀₉	60	1 811	643	6	0.01	12.70	8 424.55	7.87
CMT ₀₉	70	2 777	863	6	0.01	6.40	8 968.68	29.95
CMT ₀₉	80	9 879	2 752	7	0.10	4.45	7 896.67	3.79
CMT ₀₉	90	34 108	8 378	8	1.04	4.45	7 725.48	15.43
CMT ₀₉	100	113 569	24 960	9	15.04	3.51	7 498.90	53.18
CMT ₀₉	110	362 612	73 286	10	234.05	1.45	7 357.69	297.57
CMT ₀₉	120	1 115 535	206 372	11	2 323.26	0.00	7 340.54	1 663.67
CMT ₁₀	30	1 279	458	6	0.00	0.00	5 890.93	0.67
CMT ₁₀	40	474	191	5	0.00	1.52	7 573.60	0.28
CMT ₁₀	50	992	360	5	0.00	3.84	8 624.68	14.68
CMT ₁₀	60	2 382	802	6	0.01	7.94	8 423.82	19.88
CMT ₁₀	70	4 494	1 316	7	0.03	4.87	8 846.70	52.17
CMT ₁₀	80	7 141	2 022	7	0.06	8.76	8 721.17	22.07
CMT ₁₀	90	22 535	5 630	8	0.49	2.08	8 031.10	6.88
CMT ₁₀	100	67 993	15 123	9	4.52	2.47	7 877.49	27.63
CMT ₁₀	110	195 408	38 449	10	66.09	3.13	7 740.00	109.58
CMT ₁₀	120	540 922	95 872	11	602.23	0.19	7 667.16	482.87

Table 3.4: Results of the shortest path approach for the mt-CCSVRP for instances with
 $n = 30$ sites.

Instance	L_{max}	$ K_g $	$ K $	M_{max}	Time _g	Gap ₀	Z_p^*	Time _s
CMT ₀₆	60	2 660	930	7	0.00	10.29	11 972.47	0.22
CMT ₀₆	70	4 188	1 454	6	0.02	2.99	11 884.52	0.28
CMT ₀₆	80	18 205	5 392	8	0.47	6.03	11 137.31	3.70
CMT ₀₆	90	74 422	19 478	9	10.45	4.18	10 701.01	41.36
CMT ₀₆	100	282 931	64 931	10	229.76	3.25	10 428.35	375.68
CMT ₀₆	110	1 019 319	207 685	10	2 951.29	2.53	10 376.84	3 232.78
CMT ₀₆	120	3 356 052	1 310 047	11	30 959.25	2.51	10 422.68	80 226.02
<hr style="border-top: 1px dashed black;"/>								
CMT ₀₇	50	4 875	1 531	7	0.03	3.17	10 041.72	0.32
CMT ₀₇	60	5 687	1 753	7	0.05	3.86	11 357.59	0.53
CMT ₀₇	70	10 454	3 006	7	0.14	5.61	11 625.83	0.97
CMT ₀₇	80	40 196	9 965	8	2.11	7.06	11 293.42	10.37
CMT ₀₇	90	149 514	31 797	8	53.32	3.57	10 832.63	80.63
CMT ₀₇	100	485 501	88 918	9	753.47	3.61	10 738.75	595.76
CMT ₀₇	110	1 376 717	216 623	9	6 505.47	1.91	10 697.95	3 439.42
CMT ₀₇	120	3 325 977	476 430	9	38 343.50	1.66	10 697.95	15 774.61
<hr style="border-top: 1px dashed black;"/>								
CMT ₀₈	40	2 225	837	6	0.01	5.04	9 841.19	0.12
CMT ₀₈	50	4 583	1 408	7	0.03	4.80	10 540.81	0.26
CMT ₀₈	60	3 117	1 002	6	0.02	12.41	12 805.36	0.98
CMT ₀₈	70	5 228	1 555	7	0.04	4.21	13 053.24	0.45
CMT ₀₈	80	16 642	4 396	8	0.43	6.41	11 745.29	1.97
CMT ₀₈	90	57 085	13 434	9	6.25	1.55	11 262.15	15.41
CMT ₀₈	100	194 220	41 537	9	120.87	2.43	10 936.76	138.49
CMT ₀₈	110	619 710	117 912	10	1 219.40	3.68	10 761.69	992.23
CMT ₀₈	120	1 915 491	328 233	11	12 512.69	2.03	10 549.61	5 973.68
<hr style="border-top: 1px dashed black;"/>								
CMT ₀₉	40	3 335	1 077	6	0.02	3.06	9 518.48	0.21
CMT ₀₉	50	3 556	1 177	6	0.02	1.67	10 309.85	0.22
CMT ₀₉	60	4 561	1 466	6	0.03	11.94	11 695.71	0.33
CMT ₀₉	70	4 639	1 503	6	0.03	8.87	12 523.25	0.38
CMT ₀₉	80	19 261	5 512	7	0.45	7.51	11 362.75	4.09
CMT ₀₉	90	77 839	19 836	9	8.38	5.58	10 709.45	39.11
CMT ₀₉	100	298 288	67 979	10	183.63	1.88	10 546.39	386.07
CMT ₀₉	110	1 084 926	226 277	11	2 506.97	3.36	10 396.25	3 511.24
CMT ₀₉	120	3 812 339	732 867	12	33 255.76	3.57	10 239.78	26 629.38
<hr style="border-top: 1px dashed black;"/>								
CMT ₁₀	40	1 328	459	6	0.00	2.29	9 863.88	0.05
CMT ₁₀	50	1 753	634	6	0.00	4.05	11 715.45	0.65
CMT ₁₀	60	6 038	1 855	7	0.05	3.22	10 914.47	0.43
CMT ₁₀	70	15 136	4 105	7	0.32	8.39	11 260.09	1.94
CMT ₁₀	80	40 665	10 197	9	2.15	3.07	11 206.42	8.09
CMT ₁₀	90	160 153	35 250	10	56.59	5.85	10 662.64	87.47
CMT ₁₀	100	583 174	112 859	11	930.87	3.51	10 463.74	734.42
CMT ₁₀	110	1 981 764	336 220	12	10 573.35	2.14	10 301.81	5 338.83
CMT ₁₀	120	6 307 617	944 922	12	105 856.07	2.11	10 269.46	36 075.79

Table 3.5: Results of the shortest path approach for the mt-CCSVRP for instances with $n = 40$ sites.

Instance	L_{max}	$ K_g $	$ K $	M_{max}	Time _g	Gap ₀	Z_{p^*}	Time _s
CMT ₀₆	70	14 274	4 397	7	0.28	3.50	20 867.69	2.94
CMT ₀₆	80	67 730	18 032	9	7.91	5.84	19 361.03	49.16
CMT ₀₆	90	318 535	76 182	10	288.26	3.50	18 835.61	720.29
CMT ₀₆	100	1 435 280	306 674	11	7 568.10	4.15	18 203.66	9 493.35
CMT ₀₇	50	29 750	7 435	7	1.65	8.95	17 516.44	6.51
CMT ₀₇	60	37 199	9 568	7	1.88	5.04	18 703.93	14.47
CMT ₀₇	70	44 396	11 204	8	2.39	8.00	19 762.98	27.88
CMT ₀₇	80	181 916	41 388	8	83.44	4.80	18 866.02	246.15
CMT ₀₇	90	755 087	149 248	9	1 954.73	2.49	18 398.30	3 258.86
CMT ₀₇	100	2 719 984	475 916	9	26 683.87	2.64	18 223.89	39 534.99
CMT ₀₈	50	7 176	2 385	7	0.06	3.12	18 399.02	1.04
CMT ₀₈	60	22 509	5 820	8	0.63	8.35	20 897.66	7.25
CMT ₀₈	70	11 287	3 199	7	0.18	5.46	22 532.44	2.31
CMT ₀₈	80	52 230	13 564	8	3.80	2.22	20 443.41	22.80
CMT ₀₈	90	181 591	41 935	9	91.08	4.73	19 913.11	199.83
CMT ₀₈	100	722 857	152 764	10	1 927.96	5.27	19 231.59	2 662.42
CMT ₀₉	40	12 293	3 705	7	0.22	7.18	16 130.70	26.82
CMT ₀₉	50	37 699	9 437	9	2.38	4.16	16 569.72	13.60
CMT ₀₉	60	28 283	7 914	8	0.91	5.47	18 280.33	9.36
CMT ₀₉	70	22 859	6 329	7	0.73	5.77	21 818.16	5.15
CMT ₀₉	80	98 199	24 222	9	20.65	7.32	19 149.11	69.57
CMT ₀₉	90	441 127	97 765	10	621.76	6.82	18 574.58	903.67
CMT ₀₉	100	1 926 620	386 437	11	15 106.90	6.60	18 142.65	14 513.45
CMT ₁₀	40	6 882	2 058	7	0.07	3.14	16 073.16	1.35
CMT ₁₀	50	13 216	3 763	7	0.30	2.89	18 061.52	33.07
CMT ₁₀	60	17 778	5 282	8	0.34	6.27	18 369.49	4.36
CMT ₁₀	70	35 263	9 490	8	1.52	3.34	19 551.22	14.08
CMT ₁₀	80	171 076	40 270	10	54.32	6.18	18 595.38	185.93
CMT ₁₀	90	835 757	177 535	11	1 870.86	3.77	18 120.34	3 146.68
CMT ₁₀	100	3 870 005	746 480	11	34 203.30	2.32	17 719.54	45 118.58

with $L_{max} = 90$, and CMT₀₈ with $L_{max} = 100$. They have comparable networks (149 248 and 152 764 non-dominated trips). However, the first one which has a better initial gap (2.49% vs. 5.27%) takes longer to be solved (3 258 seconds vs. 2 662 seconds).

The time to generate the auxiliary network and compute the shortest path increases quickly with n and L_{max} . For instances with the largest L_{max} values, constructing the

network can be more time-consuming than determining the min-cost path, like for CMT_{10} with $L_{max} = 120$ in Table 3.4. Figures 3.3 and 3.4 illustrate the number of non-dominated trip-nodes ($|K|$) for the different values considered for n and L_{max} . Figure 3.3 uses a linear scale for the Y-axis while Figure 3.4 uses a logarithmic one.

Similarly, Figures 3.5 and 3.6 illustrate the total running time ($Time_g + Time_s$) for the different values considered for n and L_{max} with linear and logarithmic scale respectively. In the logarithmic plots, the curves are almost linear when L_{max} is large enough, indicating an exponential growth.

3.7 Concluding Remarks

This chapter has presented a new version of the CCVRP, in which only one vehicle is available and the range of the vehicle represents a very restrictive constraint. This combinatorial optimization problem constitutes a good way to model the delivery of relief supplies after a humanitarian disaster, since the objective function takes into account the urgency of the situation.

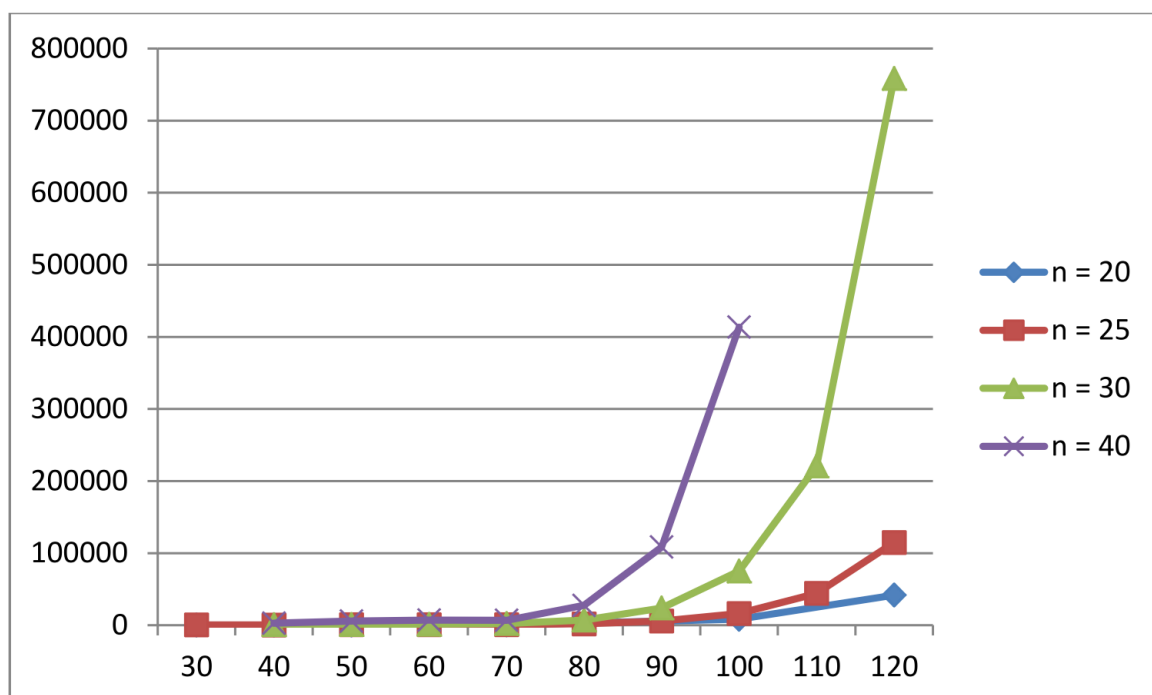


Figure 3.3: Impact of n and L_{max} on $|K|$ for the mt-CCSVRP (linear scale).

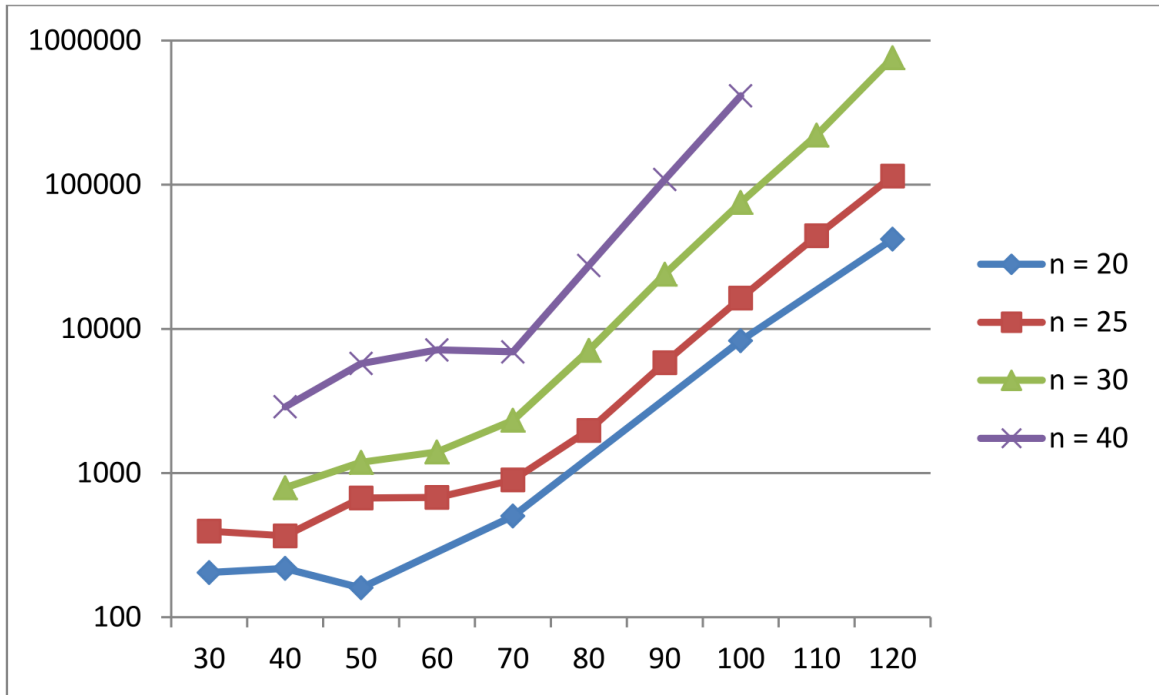


Figure 3.4: Impact of n and L_{max} on $|K|$ for the mt-CCSVRP (logarithmic scale).

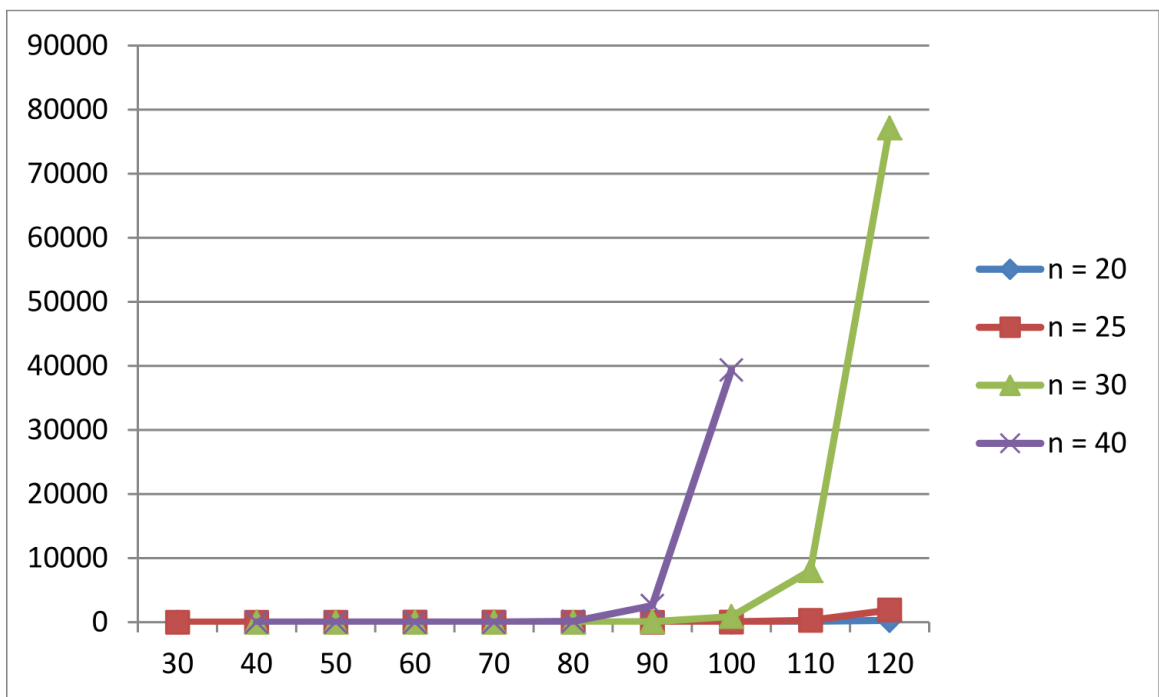


Figure 3.5: Impact of n and L_{max} on the total running time for the mt-CCSVRP (linear scale).

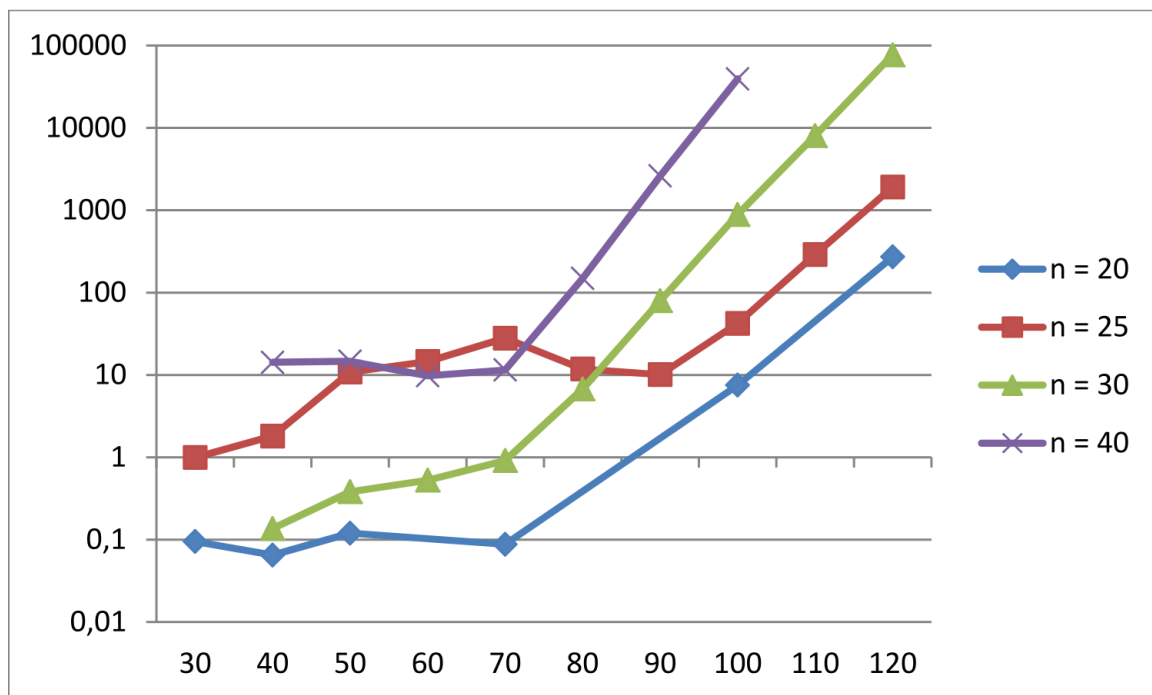


Figure 3.6: Impact of n and L_{max} on the total running time for the mt-CCSVRP (logarithmic scale).

Two mathematical models have been formulated to solve the mt-CCSVRP. Both are compared with an exact approach based on a reformulation of the mt-CCSVRP as a resource-constrained shortest path problem in an ad hoc auxiliary graph. While a commercial MIP software solves only a minority of instances for 20 required sites, the proposed shortest path model can tackle instances with up to 40 required sites, thanks to good initial solutions, dominance rules and lower bounds which accelerate the solution procedure. Like many dynamic programming approaches, the proposed exact method reaches its limits for large values of the range L_{max} , which lead to huge state-graphs.

Although a realistic intervention requires several vehicles, our single-vehicle version can be useful in a solution method where the territory concerned is divided into elementary zones, each zone being assigned to one vehicle.

Conferences and Publications

Preliminary results of the exact procedure developed in this chapter were presented in the 15-th annual conference of the French society of operations research and decision making (15ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la

Décision), ROADEF 2014:

- J. C. Rivera, H. M. Afsar and C. Prins. Une méthode exacte pour le problème de multiple-tournées cumulatifs avec un véhicule. 15ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, ROADEF 2014. Bordeaux, France. February 2014.

A full-paper version based on this chapter has been submitted to European Journal of Operations Research:

- J. C. Rivera, H. M. Afsar and C. Prins. Mathematical formulations and exact algorithm for the multitrip cumulative capacitated single-vehicle routing problem. European Journal of Operations Research. *Submitted*.

Chapter 4

The Multitrip Cumulative Capacitated Vehicle Routing Problem: Mathematical Formulations and Solution Algorithms

4.1 Introduction

In this chapter the problem presented in the previous chapter is generalized by allowing a fleet of several identical vehicles. As in the previous chapter, the objective function is the sum of arrival times and multiple trips are allowed. In the sequel, this problem will be referred as the *multitrip cumulative capacitated vehicle routing problem* (mt-CCVRP).

Three metaheuristic strategies are proposed. The components of these metaheuristics must be tailored for the multiple trips and the cumulative objective function. The considered strategies are Multistart Iterated Local Search (MS-ILS), Memetic Algorithm with Population Management (MA|PM) and Multistart Evolutionary Local Search (MS-ELS). All of them make use of a Variable Neighborhood Descent (VND) metaheuristic to improve solution. In MA|PM and MS-ELS, an adapted splitting procedure with two levels is proposed, which takes a sequence of required sites, splits it optimally into a set of multitrips and deduces an optimal set of ordered trips from each multitrip.

The mathematical formulation presented for the mt-CCVRP is an extension of the best mathematical model among the four proposed formulations and the model proposed by Ngueveu *et al.* (2010) for the CCVRP. The five formulations are mixed integer linear programs and they are compared by using small instances with a commercial solver.

The mt-CCVRP generalizes the CCVRP introduced by Ngueveu *et al.* (2010) to multiple trips. In most studies, a fundamental assumption is that each vehicle performs a single trip. Clearly, in many cases this assumption does not hold. In the last decades, some publications have studied the case where vehicles are allowed to make multiple trips. The multitrip extension of the classical CVRP is known as the multitrip vehicle routing problem (mt-VRP) and was first studied by Fleischmann (1990).

Taillard *et al.* (1996) design a hybrid algorithm for the mt-VRP by combining tabu search, a population-based approach and a bin-packing heuristic. A penalty is incurred when a trip exceeds a given time limit. Brandão & Mercer (1997) present an approach for a real life problem with multiple trips, time windows, unloading times and access restrictions at customers, and a maximum length for each trip. The same authors (Brandão & Mercer, 1998) compare a tabu search metaheuristic with the algorithm presented in Taillard *et al.* (1996).

Petch & Salhi (2004) design an algorithm combining the approaches proposed by Brandão & Mercer (1998) and by Taillard *et al.* (1996) and evaluate the savings induced by multiples trips. Olivera & Viera (2007) describe an adaptive memory procedure while Salhi & Petch (2007) develop a genetic algorithm. Gribkovskaia *et al.* (2006) present a real application of the mt-VRP. Prins (2002) investigate the mt-VRP with heterogeneous vehicles and propose constructive heuristics and a tabu search.

A hybrid memetic algorithm in which a splitting procedure is developed is proposed by Cattaruzza *et al.* (2014).

To the best of our knowledge, no published article considers the multi-trip cumulative capacitated vehicle routing problem. The special objective already complicates the moves in local search procedures for the cumulative CVRP, although each vehicle is limited to one trip in this problem (Nguèveu *et al.*, 2010). For instance, the cost of a sequence of customers changes when it is inverted by a 2-opt move, contrary to the classical CVRP. As we shall see, the calculations are even more complicated if multiple trips are allowed, since the cost of a multi-trip depends on the order of its trips.

The chapter is structured as follows: In Section 4.2 the problem is formally defined and modelled as a mixed linear program. A metaheuristic approach derived from iterated local search is developed in Section 4.3. The splitting procedure is described in Section 4.4, and

called by the memetic algorithm of Section 4.5 and the evolutionary local search-based metaheuristic of Section 4.6. Computational results are presented in Section 4.7 while concluding remarks are given in Section 4.8.

4.2 Problem Definition and Mixed Integer Linear Models

Compared to the CVRP, the mt-CCVRP uses the sum of arrival times at required nodes as objective function, like in the cumulative CVRP, but it allows more than one trip per vehicle. This flexibility is necessary when the total demand exceeds the total capacity of the fleet of vehicles. Moreover, while the fleet size is often a decision variable in CVRP, it must be fixed for the CCVRP and the mt-CCVRP, otherwise the least-cost solution consists in doing a direct trip for each required node, as noted by Nogueu *et al.* (2010).

In the context of humanitarian logistics, the mt-CCVRP models the distribution of relief supplies to a set of sites affected by a disaster, using for instance helicopters which can do multiple sorties. A planning horizon limited to one day is considered, without time windows on affected sites nor maximum working times for vehicles. Indeed, time windows are in general imposed by customers in commercial logistics, while maximum working times result from regulations about driving time. After a disaster the victims are waiting for rescue and successive crews can be used to use each vehicle full time.

The problem can be defined on an undirected complete graph $G = (V, E)$. The node-set $V = \{0, \dots, n\}$ includes a depot-node 0 and a subset $V' = V \setminus \{0\}$ of n affected sites, also called *demand nodes* or *required sites*. In the sequel, it is assumed that G is encoded as a symmetric directed graph, with travel times $w_{ij} = w_{ji}$ for each arc (i, j) in E . A fleet of R identical vehicles of capacity Q is based at the depot and each node $i \in V'$ has a known demand q_i . It is assumed without loss of generality that $\sum_{i \in V'} q_i \geq R \cdot Q$, $n \geq R$, and $q_i \leq Q \forall i \in V'$.

The objective is to identify a set of trips such that each site is visited exactly once and the sum of arrival times at sites is minimized. The most important factor in the emergency relief operations is the arrival time at sites and not the total distance traveled by vehicles. A trip is defined as a circuit, starting and ending at the depot, in which the total demand serviced does not exceed the vehicle capacity Q . Every trip must be assigned to exactly one vehicle and, if it is necessary, vehicles can perform more than one trip. Moreover, the trips assigned to each vehicle must be ordered because this affects the objective function. As it is

defined in Chapter 3 for one vehicle, the set of successive trips performed by one vehicle is called a *multitrip*.

In the following subsections, mathematical models are described. In Subsection 4.2.1 five mixed integer linear formulations for the CCVRP are presented. A comparison of these models allows to compare different approaches to formulate cumulative routing problems. In Subsection 4.2.2 the mt-CCVRP is modeled as an mixed integer linear program which extends the best model of Subsection 4.2.1.

4.2.1 Mathematical Formulations for the Cumulative Capacitated Vehicle Routing Problem

In this subsection, five mathematical formulations for the CCVRP are described. The aim of this presentation is to compare different decision variables and modeling approaches, to choose the best way to model the CCVRP and extend it in the next subsection to the multitrip case.

CCVRP mathematical model 1

The first model, referred as CCVRP1 in the sequel, is presented by Ngueveu *et al.* (2010), and it is the only one found in the literature review. This model is based on two sets of decision variables: x_{ij}^k is a binary variable equal to 1 if vehicle k traverses edge (i, j) from i to j , and t_i is the arrival time at site i . Originally the model uses t_i^k , the arrival time of vehicle k at site i , but index k is not necessary. This model is defined by Equations (4.1) to (4.9).

The objective function (4.1) models the sum of arrival times at sites. Equations (4.2) express that a vehicle arriving at site i must leave it. Equations (4.3) specify that every site must be served by exactly one vehicle. The capacity constraints are given by Equations (4.4), where the sum of demands of the nodes serviced by a vehicle cannot exceed the vehicle capacity Q . Equations (4.5) and (4.6) ensure that the depot is at the beginning and at the end of each route, respectively. Arrival times are defined via Equations (4.7): the arrival time at site j must be greater than arrival time at site i plus the travel time between the two sites if arc (i, j) is traversed by vehicle k . These equations are also useful to prevent sub-tours. Finally, Equations (4.8) and (4.9) define the domain of the decision variables. The parameter M has a big enough value satisfying $M \gg t_i^k + w_{ij}$.

$$\min Z = \sum_{i \in V'} t_i \quad (4.1)$$

$$\sum_{i \in V} x_{ij}^k = \sum_{i \in V} x_{ji}^k, \quad \forall j \in V', k \in \{1, \dots, R\} \quad (4.2)$$

$$\sum_{k=1}^R \sum_{j \in V} x_{ij}^k = 1, \quad \forall i \in V' \quad (4.3)$$

$$\sum_{i \in V'} \sum_{j \in V} x_{ij}^k \cdot q_i \leq Q, \quad \forall k \in \{1, \dots, R\} \quad (4.4)$$

$$\sum_{j \in V'} x_{0j}^k = 1, \quad \forall k \in \{1, \dots, R\} \quad (4.5)$$

$$\sum_{i \in V'} x_{i0}^k = 1, \quad \forall k \in \{1, \dots, R\} \quad (4.6)$$

$$t_i + w_{ij} - (1 - x_{ij}^k) \cdot M \leq t_j, \quad \forall i, j \in V', k \in \{1, \dots, R\} \quad (4.7)$$

$$t_i \geq w_{0i}, \quad \forall i \in V' \quad (4.8)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in V, i \neq j, k \in \{1, \dots, R\} \quad (4.9)$$

When the triangle inequality is satisfied, the arrival time at site i , given by the variable t_i , can be constrained to be greater than its distance to the depot, as shown in Equations (4.8).

CCVRP mathematical model 2

This second model, referred as CCVRP2 in the sequel, is a flow-based formulation. The index k is removed from variable x_{ij}^k , but it is noted x_{ij} as in previous model for simplicity. The set of decision variables F_{ij} measures the load on arc (i, j) and allows to limit the vehicle load. CCVRP2 model is defined by Equations (4.10) to (4.19).

The objective function (4.10) uses the same Equation (4.1) as in the model CCVRP1. Equations (4.11) limit the number of available vehicles to R . The Equations (4.12) and (4.13) specify that only one arc can arrive at each site and leave it. The Constraints (4.14) indicate that the load is reduced at every site by a quantity equal to the site demand. The capacity constraints are given by Equations (4.15): the load on every arc can exceed the vehicle capacity Q and the load on no traversed arcs is zero. Arrival times are handled by Equations (4.16) like in the model CCVRP1. These equations also prevent sub-tours in

this model. Finally, Equations (4.17), (4.18) and (4.19) define the domain of the decision variables.

$$\min Z = \sum_{i \in V'} t_i \quad (4.10)$$

$$\sum_{j \in V'} x_{0j} = R, \quad (4.11)$$

$$\sum_{i \in V} x_{ij} = 1, \quad \forall j \in V' \quad (4.12)$$

$$\sum_{j \in V} x_{ij} = 1, \quad \forall i \in V' \quad (4.13)$$

$$\sum_{j \in V} F_{ji} - \sum_{j \in V'} F_{ij} = q_i, \quad \forall i \in V' \quad (4.14)$$

$$F_{ij} \leq Q \cdot x_{ij}, \quad \forall i \in V, j \in V' \quad (4.15)$$

$$t_i + w_{ij} - (1 - x_{ij}) \cdot M \leq t_j, \quad \forall i, j \in V' \quad (4.16)$$

$$t_i \geq w_{0i}, \quad \forall i \in V' \quad (4.17)$$

$$F_{ij} \geq 0, \quad \forall i \in V, j \in V', i \neq j \quad (4.18)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V, i \neq j \quad (4.19)$$

CCVRP mathematical model 3

The third model, referred as CCVRP3 in the sequel, considers the set of decision variables y_{ij} to compute the objective function instead of variables t_i . The variable y_{ij} expresses the *arc coefficient* of the arc (i, j) , which means that the traversal time w_{ij} is summed y_{ij} times in the objective function. The concept of *arc coefficient*, introduced by Ngueveu *et al.* (2010), is detailed in Subsection 3.2.1. The proposed model is defined by Equations (4.20) to (4.30).

The objective function (4.20) expresses the sum of arrival times at sites. This equation adapts Equation (3.3) from the Flow-Based model for the mt-CCSVRP in Subsection 3.2.1. Equation (4.20) differs from Equation (3.3) because the CCVRP3 model has null service times ($s_i = 0$) and does not consider multiple trips neither loading times.

Equations (4.21) to (4.25) correspond to Equations (4.11) to (4.15) in the model CCVRP2. Equations (4.26) limit the maximal arc coefficient values. As demonstrated by Ngueveu *et al.* (2010), when $n \geq R$, all available vehicles are used in optimal solutions, which means that at least one site is visited by each vehicle. Then, the maximum number

of sites visited by a vehicle is $n - R + 1$.

Constraints (4.27) indicate that arc coefficients are decremented 1 at each visited site. Finally, Equations (4.28), (4.29) and (4.30) define the domain of the decision variables.

$$\min Z = \sum_{i \in V} \sum_{j \in V} w_{ij} \cdot y_{ij} \quad (4.20)$$

$$\sum_{j \in V'} x_{0j} = R, \quad (4.21)$$

$$\sum_{i \in V} x_{ij} = 1, \quad \forall j \in V' \quad (4.22)$$

$$\sum_{j \in V} x_{ij} = 1, \quad \forall i \in V' \quad (4.23)$$

$$\sum_{j \in V} F_{ji} - \sum_{j \in V'} F_{ij} = q_i, \quad \forall i \in V' \quad (4.24)$$

$$F_{ij} \leq Q \cdot x_{ij}, \quad \forall i \in V, j \in V' \quad (4.25)$$

$$y_{ij} \leq (n - R + 1) \cdot x_{ij}, \quad \forall i \in V, j \in V' \quad (4.26)$$

$$\sum_{j \in V} (y_{ji} - y_{ij}) = 1, \quad \forall i \in V' \quad (4.27)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V, i \neq j \quad (4.28)$$

$$y_{ij} \geq 0, \quad \forall i, j \in V, i \neq j \quad (4.29)$$

$$F_{ij} \geq 0 \quad \forall i, j \in V, i \neq j \quad (4.30)$$

Note that this model does not have big- M constraints, contrary to the two previous ones. Sub-tour elimination is modelled by Equations (4.24) and (4.27) in this formulation.

CCVRP mathematical model 4

The previous three models use binary variables x_{ij} (or x_{ij}^k) to model routes. The fourth model, CCVRP4, defines the routes by using alternative decision variables xp_{il}^k which indicate whether site i is the l -th visited site by the route k . This model is defined by Equations (4.31) to (4.38).

The Equation (4.31) describes the objective function, the sum of arrival times, like in the first two models CCVRP1 and CCVRP2. Equations (4.32) indicate that all sites must be visited exactly once while Equations (4.33) describe that at most one site can be assigned to

one position in a route. Constraints (4.34) express that a position can be occupied only if the previous position is occupied. Vehicle capacity is respected by Equations (4.35). Arrival times are limited by Constraints (4.36) where the arrival time at site j must be greater than arrival time at site i , plus the traversal time w_{ij} if both site are visited by the same vehicle and site j is visited immediately after site i . Finally, Equations (4.37) and (4.38) define the decision variable domains. This model does not require sub-tour elimination constraints.

$$\min Z = \sum_{i \in V'} t_i \quad (4.31)$$

$$\sum_{l=1}^n \sum_{k=1}^R xp_{il}^k = 1, \quad \forall i \in V' \quad (4.32)$$

$$\sum_{i \in V'} xp_{il}^k \leq 1, \quad \forall k \in \{1, \dots, R\}, l \in \{1, \dots, n\} \quad (4.33)$$

$$\sum_{i \in V'} xp_{il}^k \leq \sum_{i \in V'} xp_{i,l-1}^k, \quad \forall k \in \{1, \dots, R\}, l \in \{2, \dots, n\} \quad (4.34)$$

$$\sum_{i \in V'} \sum_{l=1}^n xp_{il}^k \cdot q_i \leq Q, \quad \forall k \in \{1, \dots, R\} \quad (4.35)$$

$$t_i + w_{ij} - (2 - xp_{il}^k - xp_{j,l+1}^k) \cdot M \leq t_j, \quad \forall i, j \in V', l \in \{1, \dots, n\}, k \in \{1, \dots, R\} \quad (4.36)$$

$$t_i \geq w_{0i}, \quad \forall i \in V' \quad (4.37)$$

$$xp_{il}^k \in \{0, 1\}, \quad \forall i \in V', l \in \{1, \dots, n\}, k \in \{1, \dots, R\} \quad (4.38)$$

CCVRP mathematical model 5

The last proposed model, referred as CCVRP5 in the sequel, uses alternative decision variables xb_{ij}^k which are equal to 1 if site i is visited before site j by route k . These decision variables also define an order on the sites visited by a vehicle. Note that these decision variables do not require that site j be visited immediately after site i . This model is defined by Equations (4.39) to (4.48).

Equation (4.39) expresses the sum of arrival time as objective function. Equations (4.40) indicate that every site must be visited by exactly one vehicle. Equations (4.41) and (4.42) mean that the variables related to a pair of sites i and j can only be used (equal to one) if sites i and j are assigned to the same vehicle. Equations (4.43) indicate that when sites i and j are visited by the same vehicle, only one order is possible. Constraints (4.44) force to assign a visiting order to sites i and j when both are assigned to the same vehicle.

The vehicle capacity cannot be exceeded thanks to Equations (4.45). Arrival times are guaranteed by Equations (4.46). Finally, Equations (4.47) and (4.48) define the domain of decision variables. No sub-tour elimination constraints are needed with this set of decision variables.

$$\min Z = \sum_{i \in V'} t_i \quad (4.39)$$

$$\sum_{k=1}^R x b_{0j}^k = 1, \quad \forall j \in V' \quad (4.40)$$

$$x b_{ij}^k \leq x b_{0i}^k, \quad \forall i, j \in V', k \in \{1, \dots, R\} \quad (4.41)$$

$$x b_{ij}^k \leq x b_{0j}^k, \quad \forall i, j \in V', k \in \{1, \dots, R\} \quad (4.42)$$

$$x b_{ij}^k + x b_{ji}^k \leq 1, \quad \forall i, j \in V', k \in \{1, \dots, R\} \quad (4.43)$$

$$x b_{ij}^k + x b_{ji}^k \geq x b_{0i}^k + x b_{0j}^k - 1, \quad \forall i, j \in V', k \in \{1, \dots, R\} \quad (4.44)$$

$$\sum_{j \in V'} x b_{0j}^k \cdot q_j \leq Q, \quad \forall k \in \{1, \dots, R\} \quad (4.45)$$

$$t_i + w_{ij} - (1 - x b_{ij}^k) \cdot M \leq t_j, \quad \forall i, j \in V', k \in \{1, \dots, R\} \quad (4.46)$$

$$t_i \geq w_{0i}, \quad \forall i \in V' \quad (4.47)$$

$$x b_{ij}^k \in \{0, 1\}, \quad \forall i \in V, j \in V', i \neq j, k \in \{1, \dots, R\} \quad (4.48)$$

The five models are compared in Section 4.7 in order to select the best model to formulate the mt-CCVRP. This comparison allows to select the best model, CCVRP3, to be extended to the multitrip case, as explained in the following subsection.

4.2.2 Mathematical Formulations for the Multitrip Cumulative Capacitated Vehicle Routing Problem

In this subsection a 0–1 mixed integer linear program (MILP) for the mt-CCVRP is described. This model extends the model CCVRP3 from previous subsection to the multitrip case, and the flow-based model from Subsection 3.2.1 to several vehicles. Nevertheless, this model considers null service times. Contrary to the model proposed by Ngueveu *et al.* (2010), variables are indexed only by arcs and no trip nor vehicle index are required. The concepts of *replenishment arc* and *arc coefficient* respectively are introduced by Boland *et al.* (2000) and Ngueveu *et al.* (2010), and they are recalled in Subsection 3.2.1. They are central in the mt-CCVRP formulation.

Replenishment arcs constitute a trick to replace the successive trips of a multitrip by a single trip: a replenishment arc (i, j) implies the traversal of arcs $(i, 0)$ and $(0, j)$ when i is the last site of a trip and j the first site of the next trip.

Arc coefficients allow to compute the sum of arrival time as a weighted sum of the traversal times. The coefficient of the arc (i, j) means the number of times the traversal time w_{ij} is counted in the objective function.

This model is based on five types of variables. Like in all flow-based vehicle routing models, variables F_{ij} define the flow on each arc (i, j) , i.e., the load of the vehicle traversing this arc. The binary variables x_{ij} are equal to 1 if and only if arc (i, j) is traversed by a vehicle. The arc coefficients explained before are expressed by variables y_{ij} which are very useful to compute the objective function and prevent subtours. Due to the special nature of replenishment arcs, similar but separate variables x'_{ij} and y'_{ij} are used for these arcs.

To better understand the model, note that four basic actions are possible for a vehicle: 1) it leaves the depot to reach the first required node j of its multitrip ($x_{0j} = 1$); 2) it traverses a normal arc linking two required nodes i and j ($x_{ij} = 1$); 3) it traverses a replenishment arc connecting two required nodes i and j , i.e., it ends a trip at i , reloads at the depot and initiates a new trip with j ($x'_{ij} = 1$); 4) it returns to the depot after the last required node of its multitrip ($x_{j0} = 1$). Normal arcs $(0, j)$ leaving the depot are used only at the beginning of the first trip of a multitrip. The fourth action is not important because the return arcs $(j, 0)$ have no influence on the objective function considered. It is assumed that variables x_{j0} exist, but with a null cost since their arc coefficients are zero. For instance, if a trip visits nodes 1, 2 and 3, the sum of arrival times is $(w_{01}) + (w_{01} + w_{12}) + (w_{01} + w_{12} + w_{23})$ and the arc coefficients are $y_{01} = 3$, $y_{12} = 2$, $y_{23} = 1$ and $y_{30} = 0$.

The resulting 0–1 mixed integer linear program, a flow-based formulation, is given by Equations (4.49) to (4.67).

$$\min Z = \sum_{i \in V} \sum_{j \in V} w_{ij} y_{ij} + \sum_{i \in V'} \sum_{j \in V'} (w_{i0} + w_{0j}) y'_{ij} \quad (4.49)$$

$$\sum_{i \in V'} x_{0i} = R, \quad (4.50)$$

$$\sum_{i \in V'} (x_{ij} + x'_{ij}) + x_{0j} = 1, \quad \forall j \in V' \quad (4.51)$$

$$\sum_{i \in V'} (x_{ji} + x'_{ji}) + x_{j0} = 1, \quad \forall j \in V' \quad (4.52)$$

$$\sum_{j \in V} F_{ji} - \sum_{j \in V} F_{ij} = q_i, \quad \forall i \in V' \quad (4.53)$$

$$F_{ij} \leq Q \cdot x_{ij}, \quad \forall i \in V', j \in V \quad (4.54)$$

$$F_{0j} \leq Q \cdot (x_{0j} + \sum_{i \in V'} x'_{ij}), \quad \forall j \in V' \quad (4.55)$$

$$y_{ij} \leq (n - R + 1) \cdot x_{ij}, \quad \forall i \in V, j \in V \quad (4.56)$$

$$y'_{ij} \leq (n - R) \cdot x'_{ij}, \quad \forall i \in V', j \in V' \quad (4.57)$$

$$\sum_{j \in V} (y_{ij} - y_{ji}) + \sum_{j \in V'} (y'_{ij} - y'_{ji}) = 1, \quad \forall i \in V' \quad (4.58)$$

$$y_{ij} \geq x_{ij}, \quad \forall i \in V, j \in V' \quad (4.59)$$

$$y'_{ij} \geq x'_{ij}, \quad \forall i \in V', j \in V' \quad (4.60)$$

$$y_{ij} \geq 2 \cdot x_{ij} - x_{j0}, \quad \forall i \in V, j \in V' \quad (4.61)$$

$$y'_{ij} \geq 2 \cdot x'_{ij} - x_{j0}, \quad \forall i \in V', j \in V' \quad (4.62)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in V, j \in V, i \neq j \quad (4.63)$$

$$x'_{ij} \in \{0, 1\}, \quad \forall i \in V', j \in V', i \neq j \quad (4.64)$$

$$y_{ij} \geq 0, \quad \forall i \in V, j \in V, i \neq j \quad (4.65)$$

$$y'_{ij} \geq 0, \quad \forall i \in V', j \in V', i \neq j \quad (4.66)$$

$$F_{ij} \geq 0, \quad \forall i \in V, j \in V, i \neq j \quad (4.67)$$

The objective function (4.49) represents the sum of arrival times to affected sites, expressed using the arc coefficient variables y_{ij} . The second sum with the y'_{ij} concern replenishment arcs (i, j) , as their traversal times are in fact $w_{i0} + w_{0j}$. Constraints (4.50) mean that only R vehicles (R multitrips) can be used (recall that variables x_{0i} are used only at the beginning of the first trip of a vehicle).

The Equations (4.51) and (4.52) respectively indicate that exactly one arc is traversed to arrive at site j and leave it. These equations are based on the three first actions explained before.

Constraints (4.53) to (4.55) concern flow variables and ensure that each demand is satisfied. Equations (4.53) express the vehicle load variation after a visit to site i and ensure that each demand is satisfied. Using Equations (4.54), no flow can traverse an unused arc and each flow is limited by vehicle capacity. Equations (4.55) limit the maximum flow at the beginning of each trip to the capacity Q of vehicles: $x_{0j} = 1$ concerns the flow for the

first trip of a vehicle while $x'_{ij} = 1$ concerns its other trips, after a replenishment arc.

Constraints (4.56) to (4.58) concern arc coefficients. Equations (4.56) (resp. (4.57)) limit the coefficients y_{ij} (resp. y'_{ij}) to the maximum number of normal or replenishment arcs that can be used in a multitrip. Ngueveu *et al.* (2010) have shown that all the vehicles are used in an optimal CCVRP solution (each vehicle visits at least one demand node) and this is still true for the mt-CCVRP. This implies that no multitrip can visit more than $(n - R + 1)$ demand nodes, which is also the maximum arc coefficient. In constraints (4.57) the limit is reduced by one, because there is at least one required node before a replenishment arc.

Constraints (4.58) imply that the coefficients of successive arcs decrease along a trip. This property prevents subtours, like in the Miller-Tucker-Zemlin subtour elimination constraints for the TSP (Miller *et al.*, 1960). Note that it is not necessary to use integer variables for y_{ij} and y'_{ij} . Constraints (4.59) to (4.62) are valid inequalities, since they are not necessary to guarantee feasibility but allow to find better linear relaxations and reduce on average the time to get optimal solutions. Equations (4.59) (resp. (4.60)) assign an arc coefficient greater than or equal to one to a normal arc (resp. replenishment arc) when this arc is traversed. Similarly, Equations (4.61) (resp. (4.62)) are valid inequalities, which express that traversed arcs with not depot destination have an arc coefficient greater or equal than two, but becomes one if the destination of its immediate successor arc is the depot ($x_{j0} = 0$).

Finally, constraints (4.63) to (4.67) define the five groups of variables. According to constraints (4.56), (4.57) and (4.58), note that the last arc traversed by each vehicle may have a rank equal to zero, which means that this arc is not considered in the objective function. The model is tested on small instances and computational results are reported in Subsection 4.7.5.

4.3 Multi-Start Iterative Local Search

Iterated Local Search (ILS) is a local search-based metaheuristic which has proved its efficiency on various optimization problems (Lourenço *et al.*, 2010). This metaheuristic is composed of three components: a constructive heuristic, an improvement heuristic, and a perturbation procedure also called mutation because of its similarity to the mutation operator of genetic algorithms.

The constructive heuristic is used to create an initial solution, which is improved by the

improvement heuristic and marked as the current solution. Then, for each iteration of the ILS, a child-solution is generated by applying a perturbation and the improvement heuristic on the current solution. If the resulting solution improves the current solution, the last one is replaced. The process is repeated until a stop criteria is found.

The ILS has been used to solve several VRP variants. For instance, Hashimoto *et al.* (2008) propose an ILS for the time-dependent vehicle routing problem with time windows, with an improvement heuristic based on 2-opt*, cross exchange and Or-opt moves. Laurent & Hao (2009) develop an ILS using ejection chains-based improvement, also called block moves, for the multiple depot vehicle scheduling problem. Cordeau & Maischberger (2012) develop a parallel metaheuristic combining ILS with tabu search to solve four VRP variants: classical VRP, periodic VRP, multi-depot VRP, and site-dependent VRP, and its time windows versions. Walker *et al.* (2012) present a flexible hyper-heuristic with a multiple-neighborhood iterated local search. Nguyen *et al.* (2012a) describe a multi-start version (MS-ILS) with short-term memory, path relinking and split procedure for the two-echelon location-routing problem. A periodic VRP with time windows and time spread constraints on services is solved by Michallet *et al.* (2014) with an MS-ILS algorithm. Vansteenwegen & Mateo (2014) address the single-vehicle cyclic inventory routing problem with an ILS algorithm. Palhazi Cuervo *et al.* (2014) elaborate an ILS for the VRP with backhauls.

In the following subsections the proposed MS-ILS and its components are described.

4.3.1 Principle and General Structure

The proposed metaheuristic is a *multi-start iterated local search* (MS-ILS) calling a *variable neighborhood descent* (VND). The MS-ILS variant, consisting in restarting an ILS from several initial solutions to diversify the search, has given excellent solutions on the CVRP (Prins, 2009a). VND is a kind of local search in which wider and wider neighborhoods are successively explored (Mladenović & Hansen, 1997). The resulting hybrid metaheuristic is sketched in Algorithm 4.1 while its internal components are described in the sequel.

MaxStart successive ILS are executed and update a global best solution BS . Each ILS calls a greedy randomized heuristic to get one initial solution, improves it using the VND, and then performs *MaxIter* iterations. Each iteration of the incumbent ILS takes a copy S' of S , applies a perturbation procedure (*Perturb*) and calls the VND to improve the perturbed solution. In most implementations, S is replaced by S' in case of improvement. As solutions with equal costs are not rare, S is also replaced by S' when $Z_S = Z_{S'}$ (line 11) to better

diversify the search.

Note that each ILS generates a sequence of local optima with decreasing costs. The procedures *Precomputations* and *Update_Precomputations*, explained later, are used to speed up the VND.

Algorithm 4.1 – MS-ILS for the mt-CCVRP

```

1:  $Z_{BS} \leftarrow \infty$ 
2: for  $Start \leftarrow 1$  to  $MaxStart$  do
3:   Greedy_Randomized_Heuristic ( $S$ )
4:   Precomputations ( $S$ )
5:   VND ( $S$ )
6:   for  $Iter \leftarrow 1$  to  $MaxIter$  do
7:      $S' \leftarrow S$ 
8:     Perturb ( $S'$ )
9:     Update_Precomputations ( $S'$ )
10:    VND ( $S'$ )
11:    if  $Z_{S'} \leq Z_S$  then
12:       $S \leftarrow S'$ 
13:    end if
14:  end for
15:  if  $Z_S < Z_{BS}$  then
16:     $BS \leftarrow S$ 
17:  end if
18: end for
    
```

4.3.2 Solution Representation

A solution is encoded as R lists of nodes, one per vehicle or multitrip. Each list indicates the order in which the required nodes are visited by the corresponding multitrip, using the depot (node 0) as delimiter. Figure 4.1 gives an example for seven required nodes with respective demands $\{10, 20, 10, 20, 10, 10, 10\}$ and two vehicles with capacity 35. The arcs used are plotted with their traversal times w_{ij} . Vehicle 1 performs one multitrip of two trips while vehicle 2 performs one trip only. Arc (2, 3) is an example of replenishment arc. The arrival times are written near each node. The cost of this solution (sum of arrival times to required nodes) is $Z = 226$.

4.3.3 Precomputations

Contrary to the CVRP, it is not obvious to evaluate a move in constant time in local search procedures for the mt-CCVRP, due to several subtle differences. Inserting or removing a node in a route shifts the arrival times at the following nodes. Several moves in the proposed VND involve reversals of sequences of nodes: unlike the CVRP, the cost of a sequence is modified after a reversal and, in particular, the cost of a trip changes when performed backwards. A move in a route changes also the costs of the following routes in the same multitrip. Finally, the cost of a multitrip depends on the ordering of its trips.

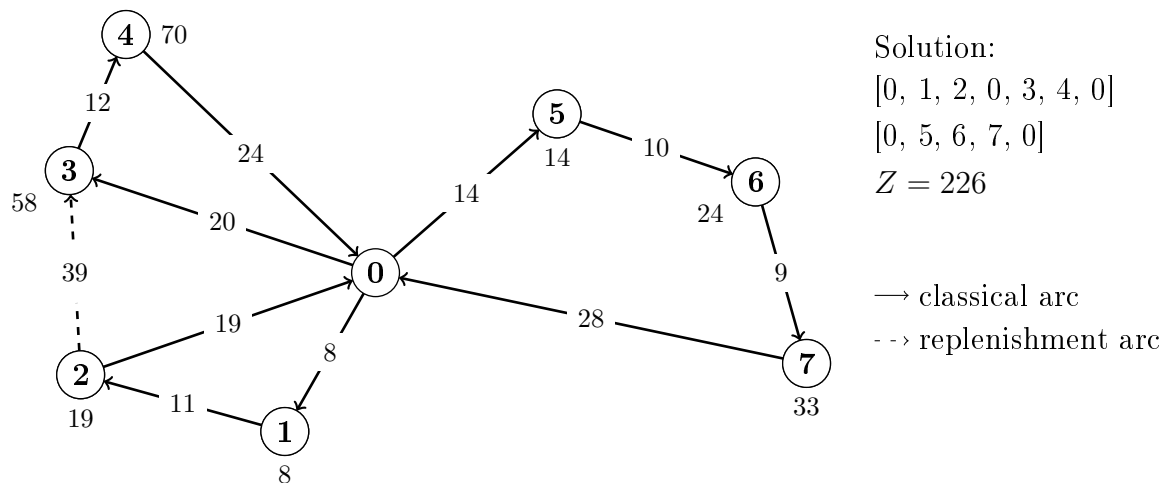


Figure 4.1: Example of a solution for an instance with $n = 7$, $R = 2$ and $Q = 35$.

Ngueveu *et al.* (2010) have succeeded in evaluating each move in $O(1)$ in their heuristics for the CCVRP, but using heavy cost variation formulas. Silva *et al.* (2012) have proposed for the cumulative TSP simpler equations, based on the fact that any move can be expressed by concatenations of node sequences. For instance, if node 5 is inserted in route $(1, 2, 3, 4)$ after node 2, the result can be written as $(1, 2) \oplus (5) \oplus (3, 4)$, \oplus denoting the concatenation. Silva *et al.* (2012) define the attributes listed in Table 4.1 for a sequence of nodes $\sigma = (\sigma_i, \sigma_{i+1}, \dots, \sigma_j)$, denoted in the sequel as $(i, i + 1, \dots, j)$ to save one level of indexing. Recall that t_i denotes the arrival time at node i .

For a sequence σ reduced to a single node, we have $M_\sigma = 1$ and $C_\sigma = D_\sigma = 0$. If sequence $\sigma' = (i', i' + 1, \dots, j')$ is concatenated after σ , the following equations can be used to deduce the attributes for $\sigma \oplus \sigma'$:

Table 4.1: Notations from Silva *et al.* (2012) for a sequence σ of nodes.

Symbol	Definition
C_σ	Cost to perform the sequence σ when leaving i at time 0, $C_\sigma = \sum_{l=i}^j t_l$
D_σ	Duration of the sequence σ , i.e., $D_\sigma = \sum_{l=i}^{j-1} w_{l,l+1}$
M_σ	Number of sites in the sequence σ

$$M_{\sigma \oplus \sigma'} = M_\sigma + M_{\sigma'} \quad (4.68)$$

$$D_{\sigma \oplus \sigma'} = D_\sigma + w_{j,i'} + D_{\sigma'} \quad (4.69)$$

$$C_{\sigma \oplus \sigma'} = C_\sigma + M_{\sigma'} \cdot (D_\sigma + w_{j,i'}) + C_{\sigma'} \quad (4.70)$$

This formalism for the cumulative TSP can be extended to the mt-CCVRP, where a sequence σ may span several trips in a multitrip and include copies of the depot. Three modifications are required. The first one is to check vehicle capacity by introducing the total demand W_σ for a sequence σ : a trip obtained by concatenating two sequences σ and σ' is feasible if $W_\sigma + W_{\sigma'} \leq Q$.

The second modification is to ignore depot copies when computing C_σ and M_σ , i.e., $u \in \sigma$ in the sum for C_σ must be such that $\sigma_u \neq 0$ and M_σ counts only the required nodes in σ . However, D_σ does not change and still includes the cost of arcs incident to depot copies. As several moves in the proposed VND are based on sequence reversals, the third change consists in defining for a sequence σ its reversal $\overleftarrow{\sigma}$. Note that $M_{\overleftarrow{\sigma}} = M_\sigma$ and $D_{\overleftarrow{\sigma}} = D_\sigma$. The way of computing $C_{\overleftarrow{\sigma}}$ is indicated at the end of this subsection.

As an example, imagine that in Figure 4.1 a new demand node 8 is inserted after node 1 in the first multitrip, with $q_8 = 5$, $w_{18} = 7$ and $w_{82} = 8$. Vehicle capacity is respected but how to compute the cost of the resulting multitrip? If $\sigma = (0, 1)$, $\sigma' = (8)$ and $\sigma'' = (2, 0, 3, 4, 0)$, the goal is to get $C_{\sigma \oplus \sigma' \oplus \sigma''}$. Using Equation (4.70), we compute first $C_{\sigma \oplus \sigma'} = C_\sigma + M_{\sigma'} \cdot (D_\sigma + w_{18}) + C_{\sigma'} = w_{01} + 1 \cdot (w_{01} + w_{18}) + 0 = 2w_{01} + w_{18}$. Then, if $\sigma \oplus \sigma' = \bar{\sigma}$, we calculate $C_{\sigma''} = t_2 + t_3 + t_4 = 0 + (w_{20} + w_{03}) + (w_{20} + w_{03} + w_{34}) = 2w_{20} + 2w_{03} + w_{34}$, and finally $C_{\bar{\sigma} \oplus \sigma''} = C_{\bar{\sigma}} + M_{\sigma''} \cdot (D_{\bar{\sigma}} + w_{82}) + C_{\sigma''} = (2w_{01} + w_{18}) + 3((w_{01} + w_{18}) + w_{82}) + (2w_{20} + 2w_{03} + w_{34})$.

We obtain $5w_{01}+4w_{18}+3w_{82}+2w_{20}+2w_{03}+w_{34} = 5 \times 8 + 4 \times 7 + 3 \times 8 + 2 \times 19 + 2 \times 20 + 12 = 182$. Note that $2 \cdot (w_{20} + w_{03})$ corresponds to the contribution of replenishment arc $(2, 3)$ in the objective function of the mathematical model (Equation (4.50)).

In the previous example, the computations can be done in constant time provided the attributes of involved sequences are already known. This is the role of the *Precomputations* procedure in Algorithm 4.1, which calculates the attributes for each sequence $\sigma = (i, i + 1, \dots, j)$ contained in the multitrips of the incumbent solution, and for its inverse $\overleftarrow{\sigma}$. This procedure contains a main loop which scans each multitrip and each node i in this multitrip, while a nested loop inspects each node j from i onward in the same multitrip.

For $i = j$, we have $M_\sigma = M_{\overleftarrow{\sigma}} = 1$ and $D_\sigma = D_{\overleftarrow{\sigma}} = C_\sigma = C_{\overleftarrow{\sigma}} = 0$. When j is incremented, the attributes for the new sequence are derived in $O(1)$ from the ones for σ , using Equations (4.68) to (4.70) with $\sigma' = (j + 1)$. For instance, $C_{\sigma \oplus \sigma'}$ is computed via Equation (4.70): $C_{\sigma \oplus \sigma'} = C_\sigma + M_{\sigma'} \cdot (D_\sigma + w_{j,j+1}) + C_{\sigma'} = C_\sigma + D_\sigma + w_{j,j+1}$. For the reverse sequence, the same formula is used but adding $j + 1$ at the beginning, i.e., $C_{\overleftarrow{\sigma \oplus \sigma'}} = C_{\overleftarrow{\sigma'} \oplus \overleftarrow{\sigma}} = C_{\overleftarrow{\sigma'}} + M_{\overleftarrow{\sigma}} \cdot (D_{\overleftarrow{\sigma'}} + w_{j+1,i}) + C_{\overleftarrow{\sigma}} = M_\sigma \cdot w_{j+1,i} + C_{\overleftarrow{\sigma}}$.

In practice, the attributes are stored in matrices, using the first node and last node of each sequence as subscripts. For instance, the value of C_σ for $\sigma = (i, i + 1, \dots, j)$ is stored in a matrix \tilde{C} , in element \tilde{C}_{ij} . As each attribute is calculated in $O(1)$, it is clear that all precomputations can be done in $O(n^2)$. After a call to *Precomputations*, each perturbation and each move in the VND can be evaluated in $O(1)$. As accepted perturbations and moves change the incumbent solution, the attributes must be updated. This task is carried out by the *Update_Precomputations* procedure, similar to *Precomputations* but restricted to the sequences contained in modified multitrips.

Note that when sequences to concatenate are trips which start and finish at the depot, for instance $\sigma_1 = k_1$ and $\sigma_2 = k_2$, Equations (4.69) and (4.70) can be simplified:

$$D_{k_1 \oplus k_2} = D_{k_1} + w_{0,0} + D_{k_2} = D_{k_1} + D_{k_2}$$

and

$$C_{k_1 \oplus k_2} = C_{k_1} + M_{k_2} \cdot (D_{k_1} + w_{0,0}) + C_{k_2} = C_{k_1} + M_{k_2} \cdot D_{k_1} + C_{k_2}$$

which are equivalent to the Equations (3.44) presented in Section 3.5 of previous chapter.

Similarly, the dominance rule given by Theorem 3.3 in Subsection 3.5.1 shows that it is not necessary to consider all possible trip orderings in a given multitrip. That theorem remains valid for the mt-CCVRP. To recall, Theorem 3.3 states that the cost of a multitrip is minimized by ordering its trips in non-decreasing order of mean trip duration, where the mean duration of a trip k is defined as its total duration D_k divided by its number of required sites M_k .

Preliminary tests have shown that it is too time-consuming to enforce the dominance rule at each step of the metaheuristic. In practice it is applied to the initial solution of each ILS, and periodically in the VND.

4.3.4 Initial Solutions

Each ILS in the proposed MS-ILS starts with an initial solution computed by a greedy randomized heuristic that builds one route at a time. Each new route is initialized with the farthest unserved site. Then, all feasible insertions of remaining sites in the incumbent routes are evaluated. A restricted candidate list (RCL) gathers the sites satisfying Equation (4.71), where V'' is the set of unserved sites, $z(i)$ is the insertion cost of site i in the emerging trip, and z_{max} and z_{min} are the largest and smallest insertion costs. One site is randomly selected from the RCL for a real insertion. The incumbent trip is finalized when all demand nodes are serviced or when all possible insertions of remaining demand nodes violate vehicle capacity.

$$RCL = \{i \in V'' \mid z(i) \leq z_{min} + 0.1 \cdot (z_{max} - z_{min})\} \quad (4.71)$$

Finally, R empty multitrips are prepared, the trips are sorted in non-decreasing order of mean duration D_k/M_k according to the Theorem 3.3, and added one by one at the end of the multitrip with minimum duration.

4.3.5 Variable Neighborhood Descent

The improvement procedure used in the proposed MS-ILS is a variable neighborhood descent (VND). VND is in itself a simple metaheuristic, based on d neighborhoods N_b , $b = 1, 2, \dots, d$. Each neighborhood is implicitly defined by a type of move. Starting from $b = 1$ and one input solution S' , the basic iteration of VND consists in exploring the neighborhood N_b of S' ($N_b(S')$). The first improvement mode is used: as soon as a better solution is discovered, it becomes the incumbent solution and b is reset to 1, otherwise b is

incremented. The procedure stops when the exploration of the last neighborhood N_d brings no improvement. The method is sketched in Algorithm 4.2.

The proposed VND for the mt-CCVRP is based upon $d = 5$ neighborhoods, each of them being implicitly defined by the union of several elementary moves. N_1 involves 2-opt moves and λ -interchanges, adapted to the cumulative objective function and applied to individual trips. N_2 considers the same moves, but applied to two trips in the same multitrip. In N_3 the same moves are attempted on two trips of different vehicles. N_4 and N_5 respectively consist in splitting a trip and reordering the trips of a multitrip. Only feasible solutions are considered, i.e., all moves must respect vehicle capacity and fleet size.

As the reversal of each moved sequence in the reinsertions are also tested, each classical CVRP move like 2-opt is here declined into several cases. Since it would take too much room to give the evaluation formula for each move and each of its cases, the equation is given only for two moves in the sequel, knowing that the transposition to the other cases is easy using the pre-computed attributes. Without the pre-processing, the cost variation formulas would be much more complicated and time consuming, especially for moves operating on two multitrips.

Algorithm 4.2 – VND for a solution S'

```

1:  $b \leftarrow 1$ 
2: while  $b \leq d$  do
3:   search for the first solution  $S'' \in N_b(S')$  such that  $Z_{S''} < Z_{S'}$ 
4:   if  $S''$  exists then
5:      $S' \leftarrow S''$ 
6:      $b \leftarrow 1$ 
7:     Update_Precomputations ( $S'$ )
8:   else
9:      $b \leftarrow b + 1$ 
10:  end if
11: end while

```

N_1 : 2-OPT moves and λ -interchanges on one trip

The 2-opt move on one trip was already used by Ngueveu *et al.* (2010) for the CCVRP. In this case, two arcs $(i, i + 1)$ and $(j, j + 1)$ are deleted in one trip k of one multitrip p

and the fragments are reconnected via two new arcs. Equivalently, it can be defined as the reversal of sequence $(i + 1, \dots, j)$. Let σ_1 be the beginning of the multitrip p up to node i included, $\sigma_2 = (i + 1, \dots, j)$, and σ_3 the rest of p . The cost variation can be computed using Equation (4.72). This move is depicted in Figure 4.2. This figure also shows the legends used to describe the changes in arc coefficients: a) unchanged arcs keep the same coefficients, b) coefficient modified indicates arcs which are changed their position in the solution, and c) inserted arcs correspond to those no belonging to original solution.

$$\Delta Z = C_{\sigma_1 \oplus \overleftarrow{\sigma_2} \oplus \sigma_3} - C_{\sigma_1 \oplus \sigma_2 \oplus \sigma_3} \quad (4.72)$$

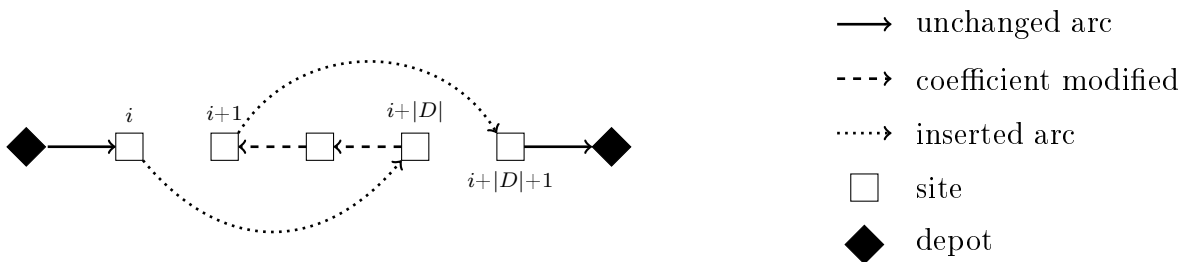


Figure 4.2: Example of 2-opt move within a route.

As the cost of a trip changes when inverted, it is considered also the new variant of Figure 4.3. Let σ_1 be the sequence containing the trips before trip k , σ_2 the beginning of k up to node i included, $\sigma_3 = (i + 1, \dots, j)$, σ_4 the end of k , and σ_5 the rest of multitrip p . Contrary to the standard 2-opt move, σ_2 and σ_4 are reversed, but not σ_3 . The cost variation is deduced by Equation (4.73).

$$\Delta Z = C_{\sigma_1 \oplus \overleftarrow{\sigma_4} \oplus \sigma_3 \oplus \overleftarrow{\sigma_2} \oplus \sigma_5} - C_{\sigma_1 \oplus \sigma_2 \oplus \sigma_3 \oplus \sigma_4 \oplus \sigma_5} \quad (4.73)$$

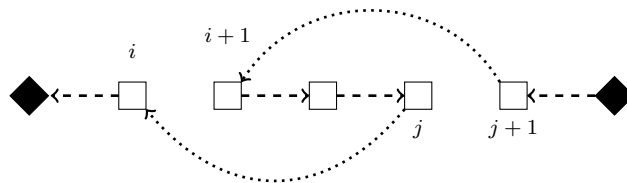


Figure 4.3: Example of 2-opt move within a reversal route.

The other moves of N_1 are λ -interchanges. They consist in exchanging a sequence with one to λ consecutive nodes with another (non-overlapping) sequence containing zero to λ consecutive nodes. The two sequences must belong to the same trip k but may have different

lengths. Each sequence with more than one node can be reversed in the reinsertion, giving four cases. A null length is allowed for the second sequence, to include relocations of the first sequence as particular case.

Figure 4.4 shows the case where no sequence is reversed. Let σ_1 be the beginning of multitrip p up to node i , $\sigma_2 = (i + 1, \dots, i')$, $\sigma_3 = (i' + 1, \dots, j)$, $\sigma_4 = (j + 1, j')$ and σ_5 the rest of the multitrip. The cost variation if σ_2 and σ_4 are exchanged is given by Equation (4.74).

$$\Delta Z = C_{\sigma_1 \oplus \sigma_4 \oplus \sigma_3 \oplus \sigma_2 \oplus \sigma_5} - C_{\sigma_1 \oplus \sigma_2 \oplus \sigma_3 \oplus \sigma_4 \oplus \sigma_5} \quad (4.74)$$

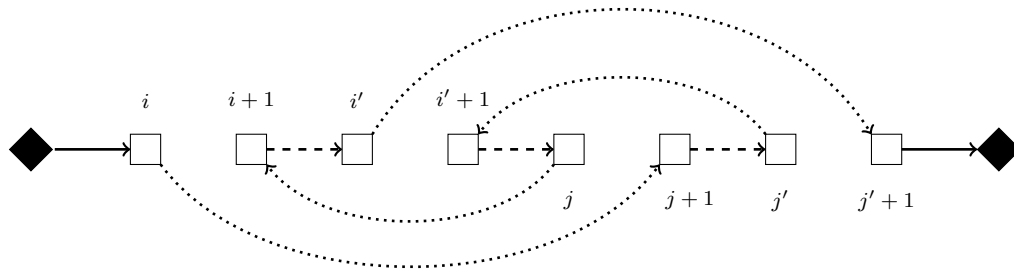


Figure 4.4: Example of λ -interchange within a route.

In all cases, the cost variation can be computed in constant time using the operator \oplus . It is easy to see that N_1 can be explored in $O(n^2/\tau)$, where τ is the number of trips in the incumbent solution.

N_2 : 2-OPT moves and λ -interchanges involving two trips in a multitrip

Figure 4.5 depicts the two kinds of 2-OPT moves for two routes k and k' contained in the same multitrip. The λ -interchanges now involve one sequence in each trip, like in Figure 4.6. This detail is important to interpret the two figures, as the line styles used (coefficients changed or not, see Figure 4.2) depend on the relative order of the two trips. As reversals of k and k' are allowed in these transformations, eight cases must be evaluated for 2-OPT moves and four for λ -interchanges. The evaluation of each case is still possible in $O(1)$ using the pre-computations, and N_2 can be searched in $O(n^2/R)$.

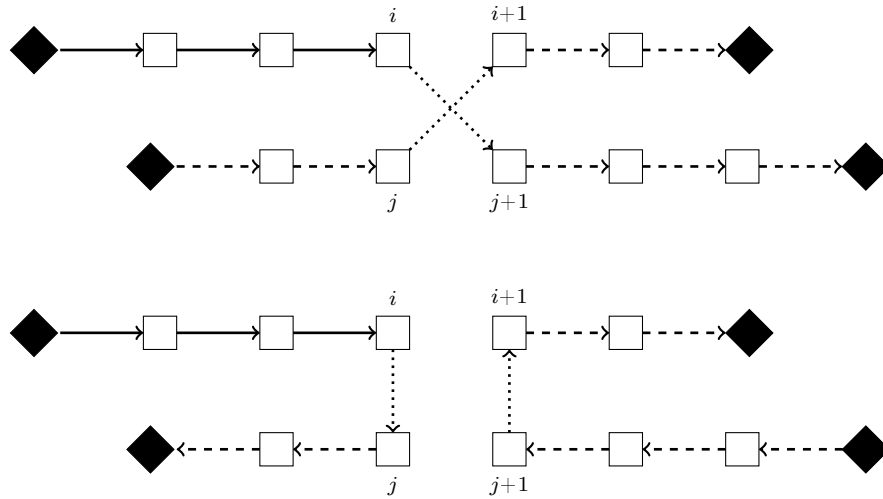


Figure 4.5: Example of 2-opt move on two trips done by the same vehicle.

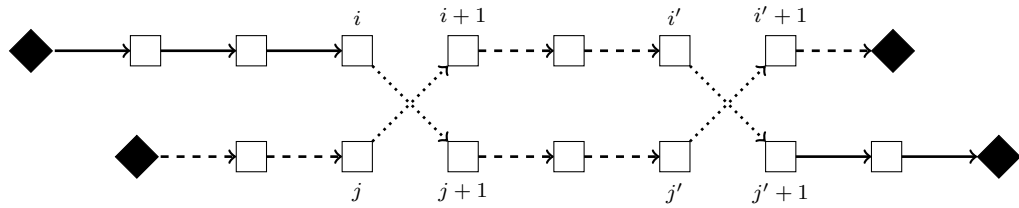


Figure 4.6: Example of λ -interchange on two trips done by the same vehicle.

N_3 : 2-OPT moves and λ -interchanges on two trips done by distinct vehicles

Moves affecting two trips of two different multitrips are similar to the ones browsed in N_1 and N_2 but require more complicated cost variation formulas since two multitrips are modified. The moves in N_3 bring deeper modifications to the incumbent solution, which explain why this neighborhood is searched after N_1 and N_2 by the VND. N_3 can be browsed in $O(n^2)$.

An example of λ -interchange move between different vehicles is shown in Figure 4.7. Note that this figure looks like Figure 4.6, but as two multitrips are concerned the arc coefficient changes are not the same. For instance, the arc coefficients of the first sequence can change in both multitrips if the interchanged sequences have different number of nodes. But interchanges never affect the arc coefficients of the last sequence in the multitrip.

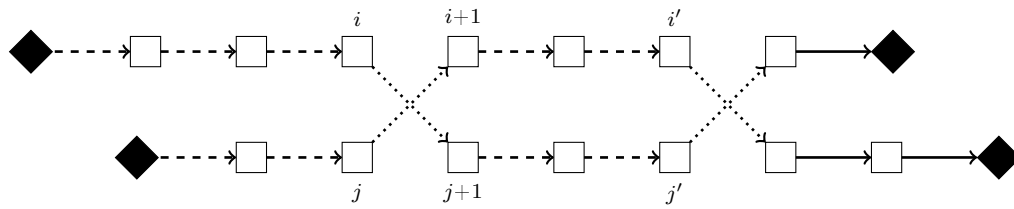


Figure 4.7: Example of λ -interchange on two trips done by distinct vehicles.

N_4 : Trip reordering

According to the dominance rule presented in Subsection 3.5.2, a solution can be improved by sorting the trips of each multitrip in non-decreasing order of mean duration. During preliminary tests we tried to evaluate the moves in N_1 , N_2 and N_3 while reordering simultaneously the affected trips, but the metaheuristic became too time-consuming. In fact, the quality of final solutions is practically the same if the trips are reordered only when the three previous neighborhoods bring no improvement.

Although this reordering phase is called N_4 in the VND, no real neighborhood is searched since the trips are sorted systematically in each multitrip. This process can be executed in $O(R m \log m)$, where R is the number of multitrips and $m = \left(\sum_{i=1}^n q_i \right) / (Q \cdot R)$ is the average number of trips per multitrip.

N_5 : Trip splitting

This neighborhood is based on an adaptation of a route-splitting procedure called *meiosis* and designed by Petch & Salhi (2004) for the multitrip VRP. This procedure determines the shortest multitrip p , inspects each multitrip $p' \neq p$ and evaluates the cost variation when a) the last trip of p' is cut after the first, second, \dots , or last but one customer, then b) the rest of the trip is moved at the end of multitrip p . This process which can be implemented in $O(R n)$ is interrupted as soon as a saving is obtained. In case of improvement, the VND goes back to neighborhood N_1 , otherwise it terminates.

4.3.6 Perturbation

The perturbation is applied before calling the VND and consists in swapping three pairs of nodes randomly selected. The nodes in each pair may belong to the same trip, to two trips of the same multitrip, or to two multitrips, but vehicle capacity must be respected. It is interesting to observe that this kind of move is tested in the neighborhoods N_1 , N_2

and N_3 , so the effect of the perturbation could be theoretically cancelled in the VND that follows. In fact, the perturbation is sufficient to change the trajectory of the VND which uses a first-improvement implementation instead of a best improvement strategy. In the perturbation different moves not used in the VND are tried, like random 4-OPT moves, but without getting better results on average.

4.4 Split Procedure for the mt-CCVRP

This section describes a split procedure applied to the solution of the mt-CCVRP. The developed method has two levels: the first level splits a giant-tour in multitrips, while the second one splits each multitrip in feasible trips.

In the following, Subsection 4.4.1 recalls the general splitting procedure and its applications, and Subsection 4.4.2 introduces the two-level split procedure developed for the mt-CCVRP.

4.4.1 The Split Procedure and The Route-First Cluster-Second Principle

The *Split* procedure is an optimization technique used in VRPs which takes a *giant-tour* solution, and optimally splits it in several routes. A giant-tour is a TSP-like tour which starts at the depot and covers all sites with unlimited vehicle capacity and without depot delimiters. When used in metaheuristics, the advantages of this technique are the reduced solution space, its flexibility to manage different features, and its efficiency (Prins *et al.*, 2014).

Beasley (1983) is the first to propose a method to decompose a giant-tour solution into several capacitated routes, by solving a shortest path problem on an auxiliary graph for the CVRP. Unfortunately, he does not provide numerical results.

Prins (2004) has the idea of using the split procedure to evaluate chromosomes encoded as giant-tours, in a memetic algorithm for the CVRP.

Split procedures have been applied and adapted later to several vehicle routing variants. Some of these adaptations consider the application of Split to the CVRP (Prins, 2004, 2009a), the VRP with heterogeneous fleet (Prins, 2009b), the multi-compartment VRP (Fallahi *et al.*,

2008), the stochastic multi-compartment VRP (Mendoza *et al.*, 2010), the VRP with time windows (Labadi *et al.*, 2008c), the capacitated arc routing problem (CARP) (Lacomme *et al.*, 2001, 2004), the periodic CARP (Lacomme *et al.*, 2005; Chu *et al.*, 2006), the CARP with time windows (Reghioui *et al.*, 2007; Labadi *et al.*, 2008b), the split delivery CARP (Labadi *et al.*, 2008a; Belenguer *et al.*, 2010), the bi-objective CARP (Lacomme *et al.*, 2006), the bi-objective stochastic CARP (Fleury *et al.*, 2008), the location-routing problem (Duhamel *et al.*, 2010), the truck and trailer routing problem (TTRP) (Villegas *et al.*, 2010), and others.

A state of the art on Split-like methods has been presented by Prins *et al.* (2014), where more than 70 references are presented, showing the flexibility of this procedure.

Ngueveu *et al.* (2010) develop a split procedure for the cumulative VRP, while Vidal *et al.* (2014) apply Split to the multitrip VRP. Nevertheless, no publication has adapted split simultaneously to cumulative objective and multiple trips. In the following subsection, a two-level Split is developed for the mt-CCVRP.

4.4.2 Two-Level Split Procedure for the mt-CCVRP

The Two-Level Split Procedure allows to obtain an mt-CCVRP solution S from a giant-tour solution T by reformulating the problem as a set of shortest path problems in an auxiliary digraph where the order of sites in T is conserved.

In the first phase, the auxiliary digraph $H = (X, A)$ is defined. $X = \{0, 1, \dots, n\}$ is the set of nodes. The arc set A contains all arcs (i, j) with $i < j$, where arc (i, j) means that the sequence of sites $\sigma = (T_{i+1}, \dots, T_j)$ is visited by a vehicle and e_{ij} is the multitrip cost (sum of arrival times assuming a beginning at time zero) of these arcs. Note that multitrips do not have capacity constraints, and the number of sites in multitrips is not limited. Nevertheless, as shown by Ngueveu *et al.* (2010), in an optimal solution for the sum of arrival times objective, all available vehicles are used, which implies in practice that the p -th multitrip can start at site T_p and finish at site T_{n-R+p} , but the number of arcs A is limited to R which increase the algorithm complexity.

Figure 4.8 illustrates an example of the graph H of the example depicted in Figure 4.1 with the giant-tour solution $T = (1, 2, 3, 4, 5, 6, 7)$. Only a subset of arcs in A is included to facilitate the comprehension of the figure. Above every node, the demand of each site is presented. Above every arc, the sequence of nodes in the corresponding multitrip is

recorded, following by the multitrip cost. The boldfaced path corresponds to the multitrip depicted in Figure 4.1. The solution cost is computed as the sum of the multitrip costs: $Z = C_{04} + C_{47} = 155 + 71 = 226$.

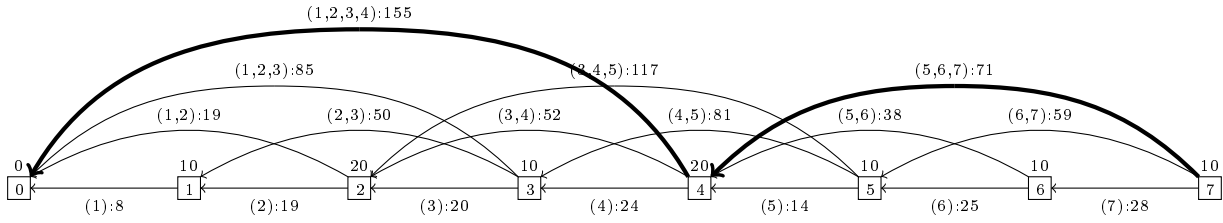


Figure 4.8: Example of the first phase split graph with $n = 7$.

Note that arc cost (or multitrip cost) e_{ij} is not unique in general since sequences can be split into different sets of trips. The optimal arc cost e_{ij} of arc $(i, j) \in A$ is computed by solving a supplementary shortest path problem on a new auxiliary digraph $H' = (X', A')$, where the set of nodes X' contains a dummy node 0, and the $j - i$ nodes of the sequence σ . The arc set A' contains arc (i', j') with $i' < j'$ if the sequence $\sigma' = (T_{i'+1}, \dots, T_{j'})$ can be serviced by a feasible trip, respecting the capacity constraint given by the Equation (4.75). The cost $e'_{i',j'}$ of arc $(i', j') \in A'$ is the sum of arrival times of the corresponding trip.

$$\sum_{v \in \sigma'} q_v \leq Q \quad (4.75)$$

Figure 4.9 illustrates the relationship between both phases of the Two Phase Split Procedure proposed. The graph above corresponds with the graph H in Figure 4.8, while the graph H' below represents the shortest path graph corresponding to the arc $(0, 4)$ in the first phase. In the graph H' each node is equivalent to a node in H . Over each arc (i', j') the best path which finish in the arc (i', j') , following by its cost, is recorded.

The trip cost $e'_{i',j'}$ of arc $(i', j') \in A'$ is the sum of arrival time of the sequence $(0, T_{i'+1}, \dots, T_{j'})$ when starting time is zero ($t_0 = 0$). In practice, the trip cost $e'_{i',j'}$ can be deduced from the trip cost $e'_{i'+1,j'}$ by following the Equation (4.76), where $e'_{j',j'} = 0$.

$$e'_{i',j'} = (j' - i') \cdot w_{0,T_{i'}} + (j' - i' - 1) \cdot (w_{T_{i'},T_{i'+1}} - w_{0,T_{i'+1}}) + e'_{i'+1,j'} \quad (4.76)$$

Similarly, the graph $H' = (X', A')$ for $X' = (0, i', \dots, j')$ can be deduced by adding the node i' and its corresponding feasible arcs to the graph $H'_2 = (X'_2, A'_2)$ with $X'_2 =$

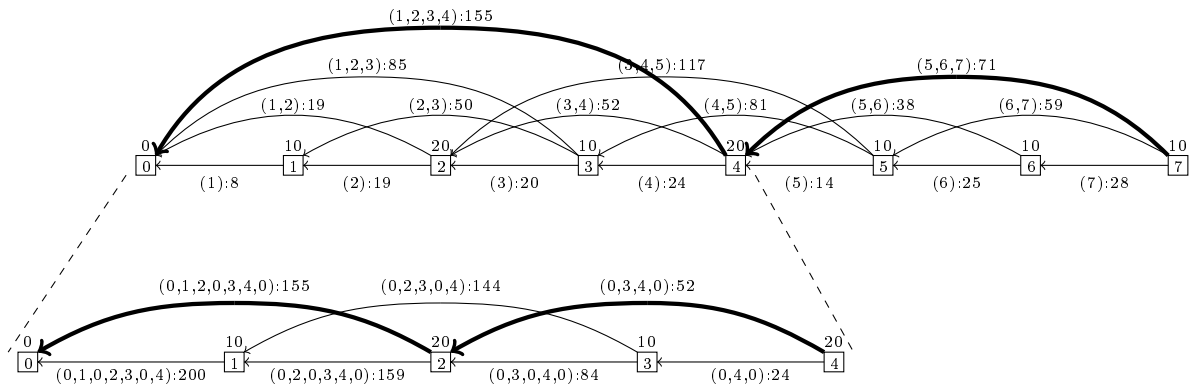


Figure 4.9: Example of the second phase split graph.

$(0, i' + 1, \dots, j')$, and their labels can be reused.

Figure 4.10 shows how the shortest path solution is interpreted. In the figure, the multitrips $(1, 2, 3, 4)$ and $(5, 6, 7)$ from Figure 4.9 are represented by dashed arrows, and the trips $(1, 2)$, $(3, 4)$ and $(5, 6, 7)$ are represented by continuous arrows.

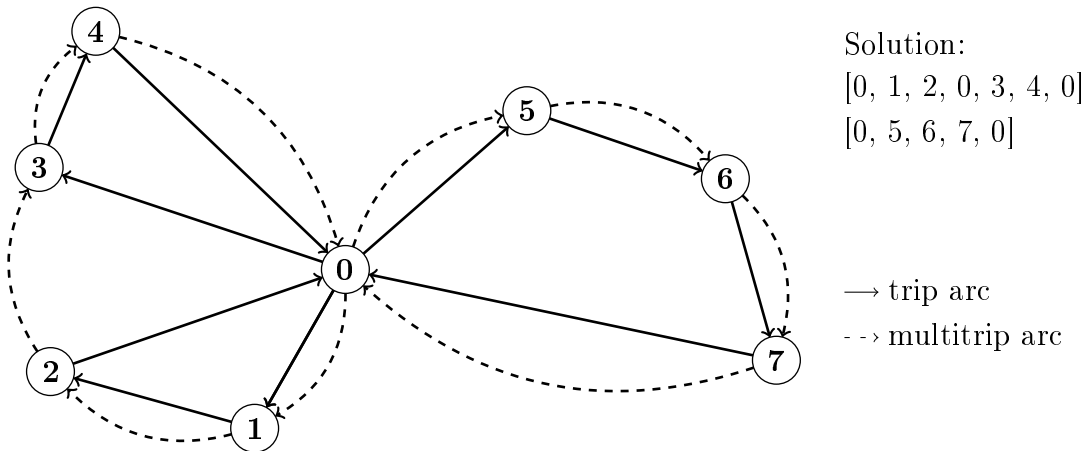


Figure 4.10: Interpretation of the shortest path given by Figures 4.8 and 4.9.

From Equation (4.70), where the cost of two concatenated sequences $C_{\sigma \oplus \sigma'}$ is computed using the concatenation operator \oplus , it can be seen that different path costs and path durations can produce several labels when paths are extended forwards. In order to reduce the complexity of the shortest path algorithm, paths are extended backwards, starting from the last node j' to the first one i' . When paths are extended backwards, only one label must be saved on each node, because the number of nodes in a path starting in node i' is fixed. In this way, shortest path can be computed using an adapted Bellman's algorithm for directed

acyclic graphs, with a complexity $O(R n^3)$, where R is the number of vehicles.

To summarize, the split algorithm has been adapted for the mt-CCVRP where a) the objective function is the sum of arrival times instead classical total travel time, b) multiple trips are allowed by managing two phases, c) the number of vehicles is limited to R , d) the generated labels are memorized to speed up the computations, and e) the paths are extended backwards which generates one label per node in the graph H' .

Algorithm 4.3 describes the first level of the proposed split method, while Algorithm 4.4 presents the second level. Algorithm 4.5 describes how to extract a mt-CCVRP solution S like introduced in Subsection 4.3.2. These algorithms use the node-labels V_i for similarity with the algorithm described by Prins *et al.* (2014).

Algorithm 4.3 performs an iteration for every available vehicle. On each iteration p , the labels V_i , W_i and Suc_{pi} of node i are updated. Label W_i indicates the cumulative cost of a shortest path from node i to node n for the iteration p based on the cumulative cost of shortest paths on iteration $p - 1$. Label Suc_{pi} stores the successor of node i on the shortest path at iteration p .

Lines 1-6 initialize the variables V_i and W_i . The value UB corresponds to an upper bound of the objective function, which is computed by assuming $\lceil \frac{n}{R} \rceil$ sites on the first $R - 1$ multitrips and the remaining sites on the last multitrip, and trips are built by adding sites while capacity constraint is satisfied.

A sequence is ignored when the number of nodes exceeds the maximum $n - R + 1$ or when the cost of the resulting path is upper then UB .

Algorithm 4.4 uses two functions to compute the cost of a multitrip. When all nodes of the sequence can be visited by a single trip, the Equation (4.76) is used in line 3, otherwise Equation (4.70) is adapted in line 15. The label Suc'_{ij} expresses the successor of site i in the graph H' for the multitrip finished at site j .

The values $load_i$ and $length_i$ respectively, are recursively computed by Equations (4.77) and (4.78), and accumulate the demand and the length along the sequence without depot visits, with $load_0 = 0$ and $length_0 = 0$.

Algorithm 4.3 – First level Split for the mt-CCVRP

```

1:  $V_n \leftarrow 0$ 
2:  $W_n \leftarrow 0$ 
3: for  $i \leftarrow 0$  to  $n - 1$  do
4:    $V_i \leftarrow UB + 1$ 
5:    $W_i \leftarrow UB + 1$ 
6: end for
7: for  $p \leftarrow 1$  to  $R$  do
8:    $j \leftarrow n$ 
9:   while  $V_j \leq UB$  and  $j \geq R - p$  do
10:     $i \leftarrow j - 1$ 
11:    if  $e_{ij}$  is not already computed then
12:       $multitrip\_cost(e_{ij})$ 
13:    end if
14:    while  $i \geq R - p$  and  $V_j + e_{ij} < W_i$  and  $V_j + e_{ij} \leq UB$  do
15:       $W_i \leftarrow V_j + e_{ij}$ 
16:       $Suc_{pi} \leftarrow j$ 
17:       $i \leftarrow i - 1$ 
18:      if  $e_{ij}$  is not already computed then
19:         $multitrip\_cost(e_{ij})$ 
20:      end if
21:    end while
22:     $j \leftarrow j - 1$ 
23:  end while
24:  for  $i \leftarrow 0$  to  $n$  do
25:     $V_i \leftarrow W_i$ 
26:     $Suc_{p+1,i} \leftarrow Suc_{p,i}$ 
27:  end for
28: end for

```

$$load_i = load_{i-1} + q_{T_i} \quad (4.77)$$

$$length_i = length_{i-1} + w_{T_{i-1}, T_i} \quad (4.78)$$

In Algorithm 4.5 R multitrips are created, which are composed by sites in the sequence $(i + 1, j)$ where $j \leftarrow Suc_{pi}$ is the successor of site i in the multitrip p . Similarly, trips are

generated with sequences $(i' + 1, j')$ where $j' \leftarrow Suc'_{i',j}$ is the successor of site i' in a trip of the multitrip finished at site j .

Algorithm 4.4 – Second level Split for the mt-CCVRP

```

1: if  $load_j - load_i \leq Q$  then
2:   if  $e'_{i,j}$  is not already computed then
3:      $e'_{i,j} \leftarrow e'_{i+1,j} + (j - i) \cdot w_{0,T_{i+1}} + (j - i - 1) \cdot (w_{T_{i+1},T_{i+2}} - w_{0,T_{i+2}})$ 
4:      $Suc'_{i,j} \leftarrow j$ 
5:   end if
6:    $e_{i,j} \leftarrow e'_{i,j}$ 
7: else
8:    $j' \leftarrow i + 1$ 
9:    $e_{i,j} \leftarrow UB$ 
10:  while  $j' \leq j$  and  $load'_j - load_i \leq Q$  do
11:    if  $e'_{i,j'}$  is not already computed then
12:       $e'_{i,j'} \leftarrow e'_{i+1,j'} + (j' - i) \cdot w_{0,T_{i+1}} + (j' - i - 1) \cdot (w_{T_{i+1},T_{i+2}} - w_{0,T_{i+1}})$ 
13:    end if
14:    if  $e_{i,j} > e'_{i,j'} + e_{j',j} + (w_{0,T_{i+1}} + length'_j - length_{i+1}) \cdot (j - j')$  then
15:       $e_{i,j} \leftarrow e'_{i,j'} + e_{j',j} + (w_{0,T_{i+1}} + length'_j - length_{i+1}) \cdot (j - j')$ 
16:       $Suc'_{i,j} \leftarrow j'$ 
17:    end if
18:     $j' \leftarrow j' + 1$ 
19:  end while
20: end if
    
```

The character “0” is added at the start of each multitrip in line 4 and at the end of each trip in line 13, which means the start and finalization of each trip at the depot. Lines 15 ($i \leftarrow Suc_{pi}$) and 18 ($i' \leftarrow Suc'_{i',j}$) indicate that the last site of a multitrip and trip respectively become the initial point for the next multitrip and trip respectively. The objective function value is placed in the label V_0 as shown in line 20.

4.5 Memetic Algorithm with Population Management

Memetic algorithms (MA) are a class of metaheuristics which combine ideas from population-based algorithms and local search techniques (Neri & Cotta, 2012). Hart *et al.* (2005) define memetic algorithms as a population-based metaheuristic composed of an

Algorithm 4.5 – Extraction of mt-CCVRP solution after Split

```

1:  $S \leftarrow \emptyset$ 
2:  $i \leftarrow 0$ 
3: for  $p \leftarrow 1$  to  $R$  do
4:    $multitrip \leftarrow \{0\}$ 
5:    $j \leftarrow Suc_{pi}$ 
6:    $i' \leftarrow i$ 
7:   while  $i' < Suc'_{i',j}$  and  $i' < Suc_{pi}$  do
8:      $trip \leftarrow \emptyset$ 
9:      $j' \leftarrow Suc'_{i',j}$ 
10:    for  $v \leftarrow i' + 1$  and  $j'$  do
11:      add site  $T_v$  at the end of  $trip$ 
12:    end for
13:    add  $\{0\}$  at the end of  $trip$ 
14:    add  $trip$  at the end of  $multitrip$ 
15:     $i' \leftarrow Suc'_{i',j}$ 
16:  end while
17:  add  $multitrip$  at the end of  $S$ 
18:   $i \leftarrow Suc_{pi}$ 
19: end for
20:  $Z_S \leftarrow V_0$ 

```

evolutionary framework and a set of local search algorithms which are activated within the generation cycle of the external framework. Nevertheless, most of the implementations in combinatorial optimization are hybrid metaheuristics combining genetic algorithms (GA) with local search procedures. GA allows to explore the search space due to maintain a pool of solutions simultaneously. On the other hand, local search procedures aim to intensify the search on promising zones of the solution space.

In basic GAs, an initial population generator creates a set of randomized solutions. A selection operator selects solutions (parents) to be combined and generate new solutions (children). The children are generated by a crossover operator. To avoid a premature convergence, some children are selected, usually with a small probability, to be mutated by a mutation operator. Finally, the population is updated to keep the best solutions. Here, the best solutions are not necessarily chosen by a quality criteria, but distance criteria helps to maintain the diversity in the population. The cycle of selection, crossover, mutation and

update is repeated until a stopping criteria is met. Local search moves are used to improve individual solutions in the population.

Memetic algorithms have been successfully used to solve vehicle routing problems and its variants. For instance, Prins (2004) propose a memetic algorithm with split for the CVRP. More recently, Vidal *et al.* (2012) have developed a memetic algorithm which can manage three problems: multidepot VRP, periodic VRP and multidepot periodic VRP. The unified framework in Vidal *et al.* (2014) use general purpose local search, genetic operator and split procedure to solve 29 variants of VRPs.

The multitrip CVRP (mt-VRP) is treated by Cattaruzza *et al.* (2014), and the cumulative CVRP is addressed by Ngueveu *et al.* (2010).

The following sections describe a memetic algorithm which combines genetic operators, variable neighborhood search metaheuristic, split procedure and population management for the mt-CCVRP.

4.5.1 Principle and General Structure

The proposed evolutionary algorithm is a *memetic algorithm with population management* (MA|PM). In this version of the algorithm, the selection and crossover operator are maintained, following by a local search improvement which is a VND similar to the one used in the MS-ILS of Section 4.3. The mutation operator is controlled by a *population management* strategy. As explained by Sörensen & Sevaux (2006), population management works by measuring and controlling the diversity of a small population of high-quality solutions, and makes ample use of *distance measures*.

A population POP_P of n_{pop} parent-chromosomes is generated by the initial population generator. Chromosomes are coded as giant-tour solutions, and their objective function values are obtained by the two phases split procedure described in Section 4.4. n_{gen} generations are performed, which allow to evolve and improve the fitness of the population. In every generation, a new population POP_C of n_{pop} child-chromosomes are generated, where each child is a combination of two parent-chromosomes selected by the selection operator. Both parents are combined by the crossover operator and translated to a mt-CCVRP solution by the splitting procedure. Finally, the VND metaheuristic is performed to improve the solution.

An acceptance criteria based on population management strategies is used to decide which chromosomes are included in population POP_P for the next generation. The mutation operator is also controlled by the acceptance criteria. The main structure of the proposed memetic algorithm is sketched in the Algorithm 4.6.

Algorithm 4.6 – MA|PM for the mt-CCVRP

```

1:  $POP_P \leftarrow initial\_population ()$ 
2:  $Z(T_{best}) \leftarrow min(Z(POP_P))$ 
3: for  $gen = 1 \rightarrow ngen$  do
4:   for  $child = 1 \rightarrow nchildren$  do
5:      $(T_{P1}, T_{P2}) \leftarrow Select (POP_P)$ 
6:      $T_C \leftarrow Crossover (T_{P1}, T_{P2})$ 
7:      $S_C \leftarrow Split (T_C)$ 
8:      $S_C \leftarrow VND (S_C)$ 
9:      $T_C \leftarrow Concatenate (S_C)$ 
10:     $POP_C \leftarrow POP_C \cup T_C$ 
11:    if  $Z(T_C) \leq Z(T_{best})$  then
12:       $T_{best} \leftarrow T_C$ 
13:    end if
14:  end for
15:   $POP_P \leftarrow Update (POP_P, POP_C)$ 
16: end for
17: return  $T_{best}$ 

```

The following subsections describe the components of the proposed MA|PM metaheuristic.

4.5.2 Solution Representation

This algorithm alternates between two solution representations: giant-tour solutions and mt-CCVRP solutions. The giant-tour solutions, noted by T , are TSP-like solutions starting at the depot, covering all required sites and without multitrip nor trip delimiters, as it is described in Section 4.4. On the other hand, mt-CCVRP solutions, noted by S , are lists of required sites in the order to be visited where every list correspond to a multitrip, and depot delimiters (site 0) are used to separate consecutive trips, as it is explained in the Subsection 4.3.2.

To handle with both solution representations, the proposed metaheuristics in Algorithm 4.6 makes use of the procedures *Split* (line 7) and *Concatenate* (line 9). The procedure *Split* optimally translates in $O(R n^3)$ a giant tour T to an adequate mt-CCVRP solution S by using the two-level split procedure described in Section 4.4. On the contrary, the procedure *Concatenate* transforms a mt-CCVRP solution S to its correspondent giant-tour T , by removing depot delimiters and merging the mt-CCVRP list in a single-list in $O(n)$.

4.5.3 Initial Population

A randomized constructive procedure based on nearest neighbor heuristic for the TSP (NNH_TSP) is used to generate the initial population. The first $n_{pop}/2$ chromosomes are generated by NNH_TSP procedure and their objective function values are obtained by calling the split method described in Section 4.4. The second half population corresponds to the resulting solutions after applying the procedures VND and Concatenate to the solutions of the first half population.

That means half of the population is composed by local optima solutions, and other half are random solutions which contribute with the diversity of the population.

4.5.4 Selection Operator

Every child is created by the combination of two parents which are selected by the function *Select* (line 5 in Algorithm 4.6). This function uses the roulette method to choose two solutions, T_{P1} and T_{P2} , from POP_P . The roulette method randomly selects two parents, where the selection probabilities depend on the quality of each solution. In the proposed method the selection probabilities are computed according with the function depicted in the Figure 4.11.

In this function, Z_{min} and Z_{max} are the smallest and the largest objective function of the solutions in the population, respectively. Similarly, pr_{max} is the probability of the solution with smallest objective function in the population, and pr_{min} , the probability of the solution with largest one. The probability of the intermediate solutions is proportional to their objective function values. The values pr_{max} and pr_{min} , as well as probabilities of the remaining solutions in the population are computed each generation because they depends on the distribution of the objective function values. Nevertheless, in this research these values are set in such a way that $pr_{max} = 3 \cdot pr_{min}$ after preliminary test.

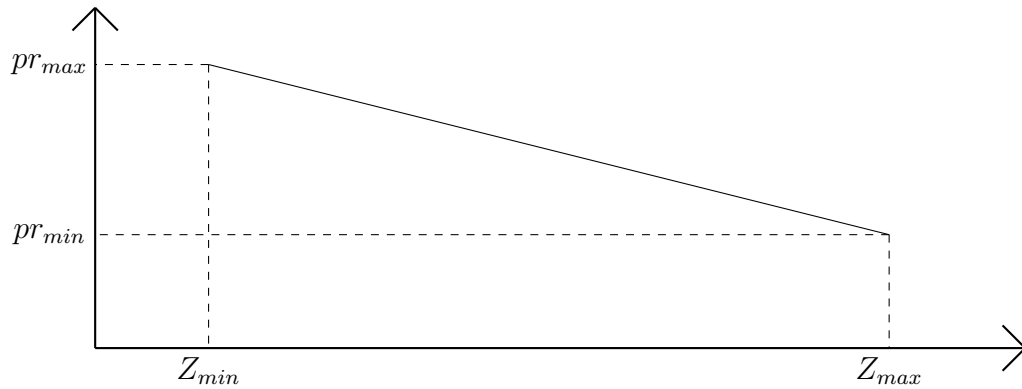


Figure 4.11: Probability function of selection operator.

4.5.5 Crossover Operator

The crossover operator used in this algorithm (line 6 in Algorithm 4.6) is an one-point crossover operator where the child takes the first m genes of one parent and the chromosome is completed with the unvisited sites in the order in which their appear in the second parent. The value r is a random integer number between $0.2 \cdot n$ and $0.8 \cdot n$.

The Figure 4.12 shows an example of the crossover operator for the example introduced in Figure 4.1 with $m = 4$.

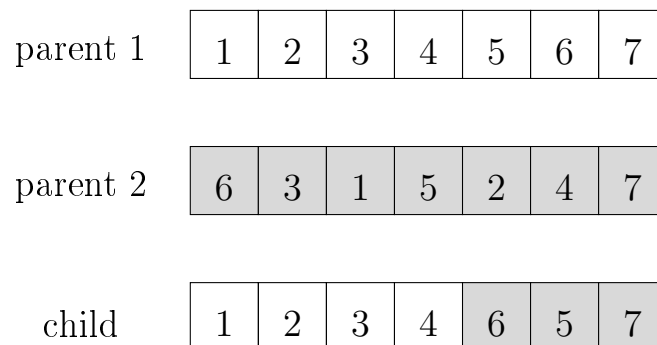


Figure 4.12: Crossover example.

4.5.6 Variable Neighborhood Descent

A VND metaheuristic is used by the MA|PM to improve the solutions quality (line 8 in Algorithm 4.6). Starting from $b = 1$ and one input mt-CCVRP solution S , the basic iteration of VND consists in exploring the neighborhood N_b of S . When the current solution S is

improved, it replaces S and b is reset to 1, otherwise b is incremented. The procedure stops when the exploration of N_d brings no improvement. Only feasible solutions are accepted. This VND is very similar to the one presented in Subsection 4.3.5, but with two main differences: first three neighborhoods are divided in two different neighborhoods each one, one neighborhood corresponding to 2-opt moves and the second one to λ -interchange moves. The second difference, contrary to the VND in MS-ILS, this VND selects the best neighbor of the current solution, because this strategy performs better for MA|PM.

With these differences, the VND applied in this MA|PM is composed by the following $d = 8$ neighborhoods:

N_1 : 2-OPT moves on one trip

N_2 : λ -interchanges on one trip

N_3 : 2-OPT moves involving two trips in a multitrip

N_4 : λ -interchanges involving two trips in a multitrip

N_5 : 2-OPT moves on two trips done by distinct vehicles

N_6 : λ -interchanges on two trips done by distinct vehicles

N_7 : Trip interchange

N_8 : Trip splitting

Moves N_1 to N_6 are already described in Subsection 4.3.5. It can be noted that first neighborhood in Subsection 4.3.5 merges neighborhoods N_1 and N_2 of this VND; second neighborhood merges N_3 and N_4 ; and third merges N_5 and N_6 .

Neighborhood N_7 is a new one which has not been tested in the VND of Subsection 4.3.5 and it is described in the following. Neighborhood N_8 is the same as fifth neighborhood from Subsection 4.3.5.

N_7 : **Trip interchange**

This neighborhood interchanges two trips from different multitrips. Every trip is inserted in the corresponding multitrip in non-decreasing order of mean trip duration by following the dominance rule given by Theorem 3.3. For instance, note that after the insertion or

removing of a trip in a multitrip, only one order must be considered. This neighborhood can be browsed in $O(m^3 R^2)$, where R is the number of multitrips and $m = \left(\sum_{i=1}^n q_i \right) / (Q \cdot R)$ is the average number of trips per multitrip.

4.5.7 Concatenation

The concatenation procedure allows to transform a mt-CCVRP solution, S , to a giant tour solution, T , without trip or multitrip delimiters in a list of size n . The procedure starts by adding to an empty list, in order, the required nodes visited by the first multitrip, after that it continues adding the required nodes of the second multitrip, and so on, until multitrip R .

4.5.8 Population Updating

The function *Update* (line 15 in Algorithm 4.6) defines the population for the following generation, POP_P . The population POP_C is composed of all generated children in the current iteration. A solution in POP_C is accepted to be included in the next population if its distance to other solutions in POP_C is larger than a parameter Δ , otherwise mutation operator is applied. Note that all solutions in POP_C are local optima, then the accepting criteria depend just on the distance measurement. No quality criteria is used. When all solutions in POP_C satisfy the distance criteria, POP_C replace POP_P .

Several distance measures have been proposed to compare two solutions. For instance, Sevaux & Sörensen (2005) describe 9 different distance measures. Sörensen & Sevaux (2006) use edit based distance in their MA|PM metaheuristic.

The exact match distance (Ronald, 1998) counts the number of times the characters in the same position are different in two strings. This distance is relevant to permutation problems where the absolute position of a node in the solution is important. Deviation distances measure the difference between the position of the items in both strings. Different definitions of deviation (i.e. absolute deviation or square deviation) produce different measures. The A distance is proposed by Campos *et al.* (2005) as the sum of absolute differences between the positions of all items in both strings. R distance (Campos *et al.*, 2005), measured as the number of times two consecutive nodes in one string are not in the other. Edit distance (Ukkonen, 1985; Sörensen, 2007) measures the cost to transform a string to another. Different implementations can produce different distance values. Similarly, reversal distance count the

number of reversals required to convert a string to the other. Longest common subsequence and longest common substring can also be used. The difference between both is that in substrings the nodes must be adjacent. The distance Kendall's tau (Sevaux & Sörensen, 2005) counts the number of times two nodes appear in the same order in both strings.

In this research, the R distance is used. This distance is relevant for the objective function studied because it measures the number of arcs in common and takes into account the arc direction. R distance is not dependent of the multitrip order in the giant-tour. The distance between two giant-tours, T and T' , is measured by metric R in Equation (4.79):

$$d_R(T, T') = \sum_{i=1}^n \xi_i, \quad (4.79)$$

$$\text{where } \xi_i = \begin{cases} 1, & \text{if } \exists j : T_i = T'_j \text{ and } T_{i+1} = T'_{j+1} \\ 0, & \text{otherwise} \end{cases}$$

The distance between a giant-tour and the population is defined as the minimum distance between the given giant-tour and all remaining giant-tours in the population (Equation (4.80)):

$$d_R^{min}(T) = \min\{d_R(T, T') \mid T, T' \in POP, T \neq T'\} \quad (4.80)$$

4.5.9 Mutation Operator

When a solution is not accepted, it is mutated to increase the distance to the population until the distance criteria is satisfied. This procedure is performed in order to increase the population diversity.

The mutation operator consists in reversing the direction of a sequence of sites in the non-accepted solution. These moves normally produce solutions with lower quality but with a large distance when it is compared with the population while some solution structures as the nearer sites are maintained. This procedure is performed during the updating procedure *Update* in Algorithm 4.6.

4.6 Multi-Start Evolutionary Local Search

The Evolutionary Local Search (ELS) is an iterative procedure in which each iteration a set of children are created by a mutation operator from a incumbent solution (Wolf & Merz, 2007). Every child is improved by a local search procedure and the best child updates the incumbent solution in case of improvement. As noticed by Prins (2009a), when the number of children is one, the procedure is equivalent to an ILS.

Prins (2009a) compares the performance of ELS with ILS and their multi-start versions called GRASP×ELS and GRASP×ILS, respectively. Duhamel *et al.* (2010) apply GRASP×ELS to the capacitated location-routing problem and use a splitting procedure. The implementation for the Two-Echelon Location Routing Problem of Nguyen *et al.* (2010) alternates between ELS and ILS in an intensification strategy. Villegas *et al.* (2010) compare a multi-start ELS with a GRASP/VND for the single truck and trailer routing problem with satellite depots. The multi-start ELS of Duhamel *et al.* (2011) solves the two-dimensional loading capacitated vehicle routing problem (2L-CVRP) from a relaxation of the problem based on the resource-constrained project scheduling problem (RCPSP-CVRP). Duhamel *et al.* (2012) propose an ELS with a new generic split procedure based on a depth first search.

In the following subsections the proposed MS-ELS and its components are described.

4.6.1 Principle and General Structure

The proposed *multistart evolutionary local search* (MS-ELS) alternates between two solution representations: mt-CCVRP solutions, which are sets of trips grouped in multitrips, and giant tours solutions without trip nor multitrip delimiters. The metaheuristic also calls a *split* procedure and a *variable neighborhood descent* (VND) as improving phases. The MS-ELS is sketched in Algorithm 4.7.

A number (*MaxStart*) of successive Randomized Greedy Solutions are constructed. Each solution S is built by the procedure *Greedy_Randomized_Heuristic* (line 3) described in Subsection 4.3.4. The resulting solution is an mt-CCVRP solution in which each multitrip is coded as a list of nodes in the order to be visited and uses the node “0” to delimit at the start and end of trips which means the visit to the depot.

The procedure *Precomputations* computes the attributes of each sequence σ described in Subsection 4.3.3 (line 4). Every randomized initial solution S , is immediately improved by

the VND procedure in line 5. In this algorithm, the VND described in Subsection 4.5.6 is used.

Algorithm 4.7 – MS-ELS for the mt-CCVRP

```

1:  $Z_{S^*} \leftarrow \infty$ 
2: for  $Start \leftarrow 1$  to  $MaxStart$  do
3:   Greedy_Randomized_Heuristic ( $S$ )
4:   Precomputations ( $S$ )
5:   VND ( $S$ )
6:   for  $Iter \leftarrow 1$  to  $MaxIter$  do
7:     Concatenate ( $S, T$ )
8:      $Z_{S'} \leftarrow \infty$ 
9:     for  $Child \leftarrow 1$  to  $MaxChildren$  do
10:       $T' \leftarrow T$ 
11:      Perturbation ( $T'$ )
12:      Split ( $T', S''$ )
13:      Update_Precomputations ( $S''$ )
14:      VND ( $S''$ )
15:      if  $Z_{S''} < Z_{S'}$  then
16:         $S' \leftarrow S''$ 
17:      end if
18:    end for
19:    if  $Z_{S'} < Z_S$  then
20:       $S \leftarrow S'$ 
21:    end if
22:  end for
23:  if  $Z_S < Z_{S^*}$  then
24:     $S^* \leftarrow S$ 
25:  end if
26: end for
    
```

Concatenate (line 7) procedure allows to translate the solution S in a giant tour T , where the last one is a TSP-like solution composed of a single list of all required nodes in the order to be visited without trip or multitrip delimiters as described in Subsection 4.5.2.

After that, a number ($MaxIter$) of iterations are performed. In each iteration

MaxChildren copies (T') of T are taken, a perturbation procedure (*Perturbation*) is applied (line 11) to each copy by the procedure *Mutation* described in Subsection 4.5.8. This perturbation procedure changes the absolute position of several nodes in the giant tour while the connections between nodes remains similar. This perturbation method keeps linked most of the nodes of the original solution but the new solution is different because a) the cost of a sequence changes when it is reversed and b) new trips are defined by the split procedure. The last feature means that even if the reversed sequence belong to the same trip, after the Split procedure the nodes in the trip can change.

The perturbed giant tours are optimally split up in multitrips to give mt-CCVRP solutions (S'') by the *Split* procedure (line 12) described in Section 4.4 and improved by VND in line 14 of Subsection 4.5.6. The procedure *Update_Precomputations* of line 13 updates the attributes of solution S to speed up the VND procedure.

The best of the *MaxChildren* solutions is used to replace S , in case of improvement. Finally, the best solution found S^* is updated when S improves the latter. In Algorithm 4.7, Z_{S^*} and $Z_{S'}$ define the global best cost and the cost of the best child of the current generation, respectively.

4.7 Computational Experiments

Computational experiments have been performed to evaluate and compare the performance of the mixed integer mathematical models and the metaheuristics algorithms for the CCVRP and the mt-CCVRP. The Subsection 4.7.1 presents the instances. Parameter tuning is described in Subsection 4.7.2. A comparison between the results of five mathematical formulations for the CCVRP on small instances is performed in Subsection 4.7.3. The results on larger instances for the CCVRP are shown in Subsection 4.7.4. For the mt-CCVRP, Subsections 4.7.5 and 4.7.6 describe the results on small and larger instances, respectively. Subsection 4.7.7 compares the solutions for the CCVRP and the mt-CCVRP.

4.7.1 Implementation and Instances

The metaheuristic algorithms, MS-ILS, MA|PM and MS-ELS, have been implemented in Visual C++ 2010 and the mathematical models in CPLEX 12.5. All of them have been tested on a 2.50 GHz Intel Core i5 computer with 4 GB of RAM and Windows 7 Professional. Four sets of experiments are reported. The first one compares the five mathematical models for the CCVRP via a commercial solver (CPLEX). Similarly, the second set of

experiments compares the resolution of the 0–1 MILP with the hybrid metaheuristic for the mt-CCVRP. Obviously, such comparisons are only possible on small instances. The third set of experiments applies the MS-ILS metaheuristic to the CCVRP, the particular case of the mt-CCVRP with one trip per multitrip, because the results of three published metaheuristics can be used for comparison. MA|PM and MS-ELS cannot solve the special case of CCVRP because split procedure needs complex modifications to force it to find only one trip per vehicle. Finally, the fourth set of experiments consists in evaluating the three metaheuristics on the mt-CCVRP.

The comparison with CPLEX involves 12 randomly generated instances with 15 demand nodes. For the CCVRP, the instances of the literature are used, which are derived from classical CVRP instances: the 14 CMT instances (Christofides *et al.*, 1979), with $n = 50$ to 199 demand nodes, and the 20 GWKC larger instances (Golden *et al.*, 1998), with $n = 200$ to 483. To get CCVRP problems, the fleet size is fixed (otherwise the optimum is trivial, with one direct route to each demand node) and the maximum trip length defined for some instances is ignored. As the seven CMT instances with a trip length constraint share the same network as the seven unconstrained instances, $7 + 20 = 27$ CCVRP instances are finally obtained. In all instances the traveling time w_{ij} on arc (i, j) is equal to the Euclidean distance, computed as a double-precision real number.

The same 27 instances are taken for the mt-CCVRP, but reducing the number of vehicles to force the metaheuristic to build several trips per vehicle. For the sake of simplicity, the resulting data files are still called CMT and GWKC instances in the sequel.

4.7.2 Parameter Tuning

Multi-Start Iterated Local Search

A big advantage of MS-ILS is to have only three parameters: the number of successive starts (*MaxStart*), the number of iterations per ILS (*MaxIter*), and the maximum number of consecutive sites in λ -interchange moves (λ).

As the running time of MS-ILS is roughly proportional to the number of calls to the VND, we decided to allocate a fixed “computing budget” of 5 000 calls to control the execution time. Like Prins (2009a), who developed an MS-ILS for the CVRP, it is observed that the partition of the calls among several restarts is critical, with two extreme options: using *MaxStart* = 1, the metaheuristic reduces to a single ILS with 5 000 iterations, while for

$MaxIter = 1$ it becomes a GRASP with 5 000 independent iterations. Several tests have been performed with different parameter values: $MaxStart$ tested values are $\{1, 3, 5, 10\}$, while $MaxIter = 5\,000/MaxStart$. Preliminary results suggest $\lambda = 3$ to balance quality and execution time. For the mt-CCVRP the best results on average are obtained using $MaxStart = 3$, $MaxIter = 1\,000$ and $\lambda = 3$.

For the CCVRP, the MS-ILS lasts longer than the metaheuristics from literature because it is designed for a more general problem: to limit the running time on these instances, we preferred to reduce the number of successive ILS restarts to two, while keeping five runs.

Memetic Algorithm with Population Management

Five parameters must be tuned when MA|PM is applied: Size of the population ($npop$), number of generations ($ngen$), the number of children ($nchildren$), the minimum distance threshold (Δ) and the maximum number of consecutive sites in λ -interchange moves (λ).

Preliminary results show that MA|PM requires additional computational effort due to resulting children from crossover operator are normally far from local optima solutions. For that reason the “computing budget” is reduced to 3 000 VND calls where the execution time is similar to the tests for the MS-ILS on CMT and small instances. As in the proposed MA|PM, the VND is applied to each generated child, the number of VND calls can be computed as $ngen \cdot nchildren$. As explained before, $\lambda = 3$ to balance quality and execution time. The number of children $nchildren$ is equal to the population size $npop$. They are tested with the values $npop = \{10, 30, 50\}$. Note that large population size values require excessive computational effort due to distance measures demand many comparisons. The number of generations $ngen$ is equal to $ngen = 3\,000/nchildren$. The parameter Δ has been tested with the values $\Delta \in \{0.2 \cdot n, 0.3 \cdot n, 0.4 \cdot n\}$. Large Δ values accept only very dissimilar solutions while values close to zero only reject very close solutions. The chosen values allow a relative good diversity while the quality is not very penalized. On average, the best results are obtained using $npop = nchildren = 30$, $ngen = 100$, $\lambda = 3$ and $\Delta = 0.3 \cdot n$. The parameter Δ is the minimum distance to accept a child, where high values require larger differences between solutions.

Note that in this method, contrary to most MA implementations, the size of the population is relatively small. Indeed too many comparisons must be performed when larger populations are used, which increases the running time.

Multi-Start Evolutionary Local Search

The MS-ELS has only four parameters: the number of successive starts ($MaxStart$), the number of iterations per ELS ($MaxIter$), the number of children ($MaxChildren$) and the maximum number of consecutive sites in λ -interchange moves (λ).

Due to the similarity between MA|PM and MS-ELS components, a “computing budget” of 3000 VND calls is fixed. Like in MS-ILS, $MaxStart = \{1, 3, 5, 10\}$ and $\lambda = 3$ are tested. Nevertheless, computing budget is now computed as $MaxStart \cdot MaxIter \cdot MaxChildren$. $MaxChildren = \{1, 5, 10\}$ values are tested. The best results on average are obtained with $MaxStart = 3$, $MaxIter = 100$, $MaxChildren = 10$ and $\lambda = 3$.

For large instances results are also compared with $MaxStart = 3$, $MaxIter = 1000$, $MaxChildren = 1$ and $\lambda = 3$ which is equivalent to a MS-ILS by performing the splitting procedure.

4.7.3 Comparison of Different Models for the CCVRP

In this subsection the five mathematical models presented in Subsection 4.2.1 are compared. The results are presented in Table 4.2.

Table 4.2 keeps the same four first columns than Tables 4.5 and 4.6: the instance name, the number of required sites n , the number of vehicles R , and the factor $\left(\sum_{i=1}^n q_i\right)/(Q \cdot R)$. Note that the meaning of this factor changes. As only one trip is allowed in the CCVRP, this factor defines the average loading level of vehicles and the strength of the capacity constraint. Then, for each model, four columns are shown: the linear relaxation (LR), the solution cost (Cost), the percentage gap between the best integer solution and the best lower bound given by CPLEX (gap), and the running time in seconds (Time), limited to one hour. When CPLEX must be interrupted, the best lower bound and the cost of the best integer solution found are reported instead of solution value. Gaps equal to zero (displayed in boldface) means that CPLEX has found the optimal solution in less than one hour.

Models which are based on the use of decision variables t_i , defining the arrival times at site i , always find the same linear relaxation, for instance models CCVRP1, CCVRP2, CCVRP4 and CCVRP5. These linear relaxation values also are shared for instances with the same node locations and demands, as instances test₀₁ to test₀₆ and instances test₀₇ to test₁₂. Model CCVRP3, where the sum of arrival times is computed using arc coefficients

Table 4.2: Comparison of the mathematical models for small CCVRP instances.

File	n	R	$\sum \frac{q_i}{Q}$	CCVRP1			CCVRP2			CCVRP3			CCVRP4			CCVRP5									
				LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time						
test ₀₁	15	2	1.00	1072.94	625.18	71.62	3600.00	502.41	1072.94	680.42	57.69	3600.00	684.04	1072.94	0.00	93.95	502.41	1116.26	549.12	103.28	3600.00	502.41	1072.94	0.00	209.53
test ₀₂	15	2	0.83	1072.94	631.96	69.78	3600.00	502.41	1072.94	705.92	51.99	3600.00	682.83	1072.94	0.00	141.59	502.41	1148.03	507.88	126.04	3600.00	502.41	1072.94	0.00	191.58
test ₀₃	15	3	0.95	785.60	595.18	31.99	3600.00	502.41	807.93	733.26	10.18	3600.00	630.55	785.60	0.00	11.58	502.41	838.00	529.50	58.26	3600.00	502.41	785.60	0.00	78.78
test ₀₄	15	3	0.83	785.60	583.12	34.72	3600.00	502.41	785.60	0.00	2757.27	630.55	785.60	0.00	5.66	502.41	838.95	66.99	838.95	66.99	3600.00	502.41	785.60	0.00	111.44
test ₀₅	15	4	1.00	687.29	563.40	21.99	3600.00	502.41	687.29	0.00	118.17	588.75	687.29	0.00	2.70	502.41	687.29	513.96	33.72	3600.00	502.41	687.29	0.00	168.75	
test ₀₆	15	4	0.83	646.67	0.00	3583.58	502.41	646.67	0.00	35.69	587.48	646.67	0.00	0.64	502.41	657.08	30.79	502.41	657.08	30.79	3600.00	502.41	646.67	0.00	146.72
test ₀₇	15	2	1.00	2456.63	1333.98	84.16	3600.00	1034.13	2348.37	1401.88	67.52	3600.00	1392.78	2348.37	0.00	207.37	1034.13	2442.52	1076.28	126.94	3600.00	1034.13	2348.37	0.00	159.47
test ₀₈	15	2	0.83	2335.40	1344.39	73.71	3600.00	1034.13	2335.40	1433.16	62.95	3600.00	1387.66	2335.40	0.00	134.03	1034.13	2543.58	1034.13	145.96	3600.00	1034.13	2335.40	0.00	160.30
test ₀₉	15	3	1.00	1717.46	1226.77	40.00	3600.00	1034.13	1717.46	1430.05	20.10	3600.00	1308.22	1717.46	0.00	38.47	1034.13	1728.49	1087.13	59.00	3600.00	1034.13	1717.46	0.00	137.81
test ₁₀	15	3	0.83	1628.31	1248.16	30.46	3600.00	1034.13	1628.31	1476.81	10.26	3600.00	1298.90	1628.31	0.00	3.31	1034.13	1651.38	1034.13	59.69	3600.00	1034.13	1628.31	0.00	87.77
test ₁₁	15	4	0.94	1317.97	1217.81	8.22	3600.00	1034.13	1317.97	0.00	12.70	1235.82	1317.97	0.00	0.78	1034.13	1317.97	1068.66	23.33	3600.00	1034.13	1317.97	0.00	93.56	
test ₁₂	15	4	0.83	1317.97	1317.97	0.00	1612.30	1034.13	1317.97	0.00	17.61	1224.97	1317.97	0.00	0.89	1034.13	1360.25	1034.13	31.54	3600.00	1034.13	1317.97	0.00	72.97	
Mean						38.89	3433.16				23.39	2345.12		0.00	53.41			72.13		3600.00				0.00	134.89

variables y_{ij} instead of arrival time variables t_i find better (greater) linear relaxations for the 12 test instances. The linear relaxations for this model increase when the number of vehicles decreases.

Model CCVRP4 has the worst performance. It finds no optimal solutions in an one hour limit and its average gap is 72.13% (23.33% in the best case and 145.96% in the worst case).

Model CCVRP1 finds two optimal solutions, the optimal solutions for the two instances with larger number of vehicles and lower average load of vehicles. Its worst gap is 81.16%, and 38.89% on average. Model CCVRP2 finds four optimal solutions, and the average gap is reduced to 23.39%.

Only models CCVRP3 and CCVRP5 find all optimal solutions in less than one hour. Nevertheless, CCVRP5 is 2.5 times slower. CCVRP3 is faster in every instance.

When CPLEX finds the optimal solution, the time to find the optimal solution tends to increase when the number of vehicles decreases or when the average loading factor increases. Similarly, when the time limit is reached, the gap tends to increase when the number of vehicles decreases or when the average loading factor increases.

4.7.4 Results on Larger Instances for the CCVRP

To this date, as far as we know, four metaheuristics have been published for the CCVRP: the memetic algorithms MA1 and MA2 of Ngueveu *et al.* (2010), the adaptive large neighborhood search (ALNS) of Ribeiro & Laporte (2012) and the two-phase metaheuristic T-PM of Ke & Feng (2013). All these authors have solved the CMT and GWKC instances using 5 runs, except Ngueveu *et al.* who have tackled only the CMT problems, and with a single run. However, for a fair comparison with ALNS, Ribeiro and Laporte asked the codes of MA1 and MA2 to Ngueveu *et al.* to solve five times the CMT and GWKC instances on their computer. In this comparison the same recomputed costs is used.

From the metaheuristics proposed in this research, only MS-ILS can solve CCVRP (single trip version), but it must be modified to use at most one trip per vehicle. MA|PM and MS-ELS cannot solve this version of the problem because the split algorithm cannot guarantee to find a solution with only one trip per vehicle.

On CCVRP instances, the initial constructive heuristic of Subsection 4.3.4 builds in

general no more than R trips and the final step that packs these trips into multitrips is no longer used. The following simple procedure is applied when R trips are not sufficient: a) the node u not yet serviced with largest demand is determined; b) if u cannot be feasibly inserted, demand nodes are removed at random from the R trips until feasible insertions appear; c) the best possible insertion of u in the R trips is performed. This process, is repeated until all sites are routed, never fails on the instances tested.

Apart from the suppression of N_5 and the modification of the initial heuristic, the code of MS-ILS remains identical when applied to the CCVRP: for instance, complex moves in neighborhoods N_2 , N_3 and N_4 are not simplified or removed.

Table 4.3 compares the four published algorithms with MS-ILS on the seven CMT instances and Table 4.4 does it for the 20 GWKC instances. On each table, the four first column headers correspond to the instance name, the number of demand nodes n , the number of vehicles used R , and the best known solution (BKS) found using all heuristics. For each algorithm are indicated the best and average deviation to the best known solution over 5 runs (BestD and AvgD), and the average computational time in seconds (Time). Best known solutions have a null deviation and they are displayed in boldface.

On the seven CMT instances, the proposed MS-ILS retrieves six best known solutions. The running time increases quickly with instance size, but remains comparable with the durations of the other heuristics. The only instance where MS-ILS from this research fails to find the BKS is the largest one (199 demand nodes). Considering performance indicators averaged on the seven instances, the execution time is less than two and a half minutes (variation between 19 seconds and 5 minutes) and the best and average deviations are quite small (0.05% and 0.49%). Moreover, the best deviation is outperformed only by T-PM and MS-ILS gives the second best average deviation after ALNS.

On the 20 larger GWKC instances, MS-ILS finds only one best known solution. TP-M and ALNS are better in terms of solution quality and running time, but MS-ILS metaheuristic is still better than MA1 and MA2 for these two criteria. Concerning the results averaged over the 20 instances, the average execution time is around 45 minutes, varying between 4 and 189 minutes, the best deviation is 1.01% and the average deviation 1.26%. These results are good, keeping in mind that MS-ILS is designed for a more general problem (the mt-CCVRP) and tested here without simplifying its code.

Table 4.3: Results on CMT instances (Christofides *et al.*, 1979) for the CCVRP.

Instance	n	R	BKS	MA1		MA2		ALNS		T-PM		MS-ILS						
				BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time			
CMT ₀₁	50	5	2 230.35	0.00	0.00	10.63	0.00	0.69	0.00	0.22	30.29	0.00	0.75	17.64	0.00	0.34	19.29	
CMT ₀₂	75	10	2 391.63	1.27	2.15	27.78	1.57	6.88	0.00	0.42	60.77	0.00	1.62	22.48	0.00	1.00	25.22	
CMT ₀₃	100	8	4 045.41	0.68	0.68	97.91	0.68	0.85	0.00	0.46	172.45	0.00	0.46	60.96	0.00	0.44	141.17	
CMT ₀₄	150	12	4 987.51	0.00	0.67	449.44	0.00	0.19	188.41	0.00	0.15	235.12	0.00	0.28	92.68	0.00	0.59	244.09
CMT ₀₅	199	17	5 809.59	0.01	0.56	1035.45	0.01	0.54	629.27	0.49	0.83	277.37	0.00	0.81	135.36	0.38	0.71	311.88
CMT ₁₁	120	7	7 314.55	0.05	1.11	160.64	0.45	1.11	68.83	0.02	0.37	202.07	0.00	0.27	71.64	0.00	0.27	136.15
CMT ₁₂	100	10	3 558.92	0.00	0.01	38.20	0.01	0.01	23.66	0.00	0.20	152.74	0.00	0.06	53.72	0.00	0.08	128.50
Average				0.29	0.74	260.01	0.39	1.47	136.62	0.07	0.38	161.54	0.00	0.61	64.93	0.05	0.49	143.76

– The costs and times reported for MA1 and MA2 are the ones recomputed in Ribeiro & Laporte (2012), see text.

– Best and average deviations given for 5 runs. Average times per run for a 2 GHz PC except for MS-ILS, tested on a 2.5 GHz PC.

Table 4.4: Results on GWKC instances (Golden *et al.*, 1998) for the CCVRP.

Instance	n	R	BKS		MA1		MA2		ALNS		T-PM		MS-ILS					
			BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	BestD	AvgD	Time	
GWKC ₀₁	240	9	54786.92	0.24	0.36	1593.60	0.26	0.43	866.80	0.19	0.31	1038.27	0.00	0.13	347.63	0.40	0.57	1188.64
GWKC ₀₂	320	10	100621.30	0.17	0.25	4549.05	0.14	0.26	2054.68	0.00	0.27	1484.74	0.00	0.12	848.55	0.12	0.14	5491.50
GWKC ₀₃	400	10	171027.05	0.14	0.21	9295.64	0.16	0.28	3336.09	0.34	0.70	2061.75	0.00	0.17	1466.81	0.00	0.09	11394.18
GWKC ₀₄	480	10	262049.31	0.17	0.26	12810.31	0.19	0.33	6557.67	0.49	1.17	2626.89	0.00	0.17	1946.57	0.08	0.09	10181.15
GWKC ₀₅	200	5	114163.52	0.00	0.06	471.45	0.00	0.15	246.05	0.29	0.60	1200.67	0.00	0.01	349.82	0.03	0.05	2004.68
GWKC ₀₆	280	7	140429.95	0.00	0.02	2358.40	0.02	0.09	788.71	0.27	0.36	1547.43	0.00	0.07	548.18	0.04	0.09	4534.21
GWKC ₀₇	360	8	180481.56	1.55	3.45	1537.36	5.48	6.70	258.45	0.00	0.63	1926.19	1.68	2.10	429.32	3.28	4.77	1703.93
GWKC ₀₈	440	10	194100.64	0.24	0.34	9598.42	0.22	0.31	5165.83	0.58	0.68	2330.34	0.00	0.16	1493.28	0.38	0.43	5376.19
GWKC ₀₉	255	14	4725.58	0.15	0.35	2453.61	0.29	0.43	1582.86	0.04	0.09	864.89	0.00	0.08	310.34	0.57	0.67	1002.82
GWKC ₁₀	323	16	6713.92	0.27	0.49	7652.00	0.11	0.43	4332.24	0.00	0.06	1092.34	0.02	0.08	484.25	0.29	0.38	2052.32
GWKC ₁₁	399	18	9214.07	0.31	0.49	8835.74	0.29	0.47	7429.60	0.00	0.03	1356.99	0.04	0.12	709.16	0.37	0.40	3663.71
GWKC ₁₂	483	19	12526.17	0.82	0.98	9881.24	1.06	1.46	3449.06	0.00	0.13	1540.32	0.19	0.35	829.51	1.20	1.57	2836.79
GWKC ₁₃	252	26	3628.30	0.68	0.90	2012.17	1.61	9.31	74.31	0.00	0.28	632.72	0.20	0.77	195.68	1.20	1.34	300.69
GWKC ₁₄	320	29	5216.80	10.60	15.88	2263.84	11.69	23.07	9.59	0.00	0.79	682.19	1.40	1.78	158.83	2.64	3.00	254.69
GWKC ₁₅	396	33	7010.41	0.96	1.85	5597.98	22.34	22.34	0.00	0.00	0.18	855.25	0.24	0.93	798.44	1.48	1.83	627.96
GWKC ₁₆	480	37	9250.98	0.54	0.96	16204.52	1.04	1.83	5307.29	0.00	0.19	1104.53	0.22	0.46	1218.37	1.31	1.45	949.39
GWKC ₁₇	240	22	3065.46	0.83	1.29	1631.86	1.01	1.74	431.31	0.03	0.12	618.70	0.00	0.47	345.22	1.11	1.25	235.96
GWKC ₁₈	300	27	4221.14	7.27	8.56	2410.31	20.43	21.72	0.10	0.00	0.56	630.47	2.13	3.10	405.28	3.39	4.28	240.15
GWKC ₁₉	360	33	5523.38	1.03	1.37	5535.25	1.57	2.14	763.56	0.18	0.33	853.43	0.00	0.37	630.62	0.61	0.93	432.90
GWKC ₂₀	420	38	7223.08	2.64	3.47	6264.59	17.04	17.04	0.00	0.00	0.25	881.92	0.20	1.11	725.80	1.62	1.79	449.05
Average				1.43	2.08	5647.86	4.25	5.53	2132.71	0.12	0.39	1266.50	0.32	0.63	712.08	1.01	1.26	2746.05

– The costs and times reported for MA1 and MA2 are the ones recomputed in Ribeiro & Laporte (2012), see text.

– Best and average deviations given for 5 runs. Average times per run for a 2 GHz PC except for MS-ILS, tested on a 2.5 GHz PC.

4.7.5 Results on Small Instances for the mt-CCVRP

Table 4.5 presents the results for the 0–1 mixed integer linear program on small instances with $n = 15$ required sites, while Table 4.6 presents the results of the three metaheuristics on the same instances.

In Table 4.5 the first four columns display the instance name, the number of required sites n , the number of vehicles R , and the average number of trips per vehicle $\left(\sum_{i=1}^n q_i\right)/(Q \cdot R)$. This table presents two types of results for the mathematical model developed in Section 4.2: the first without valid inequalities (4.59) to (4.62), called 0–1 MILP, and then with valid inequalities, called 0–1 MILP \star . For both CPLEX results the cost of the linear relaxation of the mathematical model (LR), the solution value (Cost), the percentage gap between the best integer solution found and the best lower bound found (gap), which is given by CPLEX, and the running time in seconds (Time), limited to one hour. When CPLEX must be interrupted, the best lower bound and the cost of the best integer solution found are reported instead of solution value. The comparison between both models, 0–1 MILP and 0–1 MILP \star , allows to evaluate the impact of the valid inequalities (4.59) to (4.62).

Table 4.5: Results of the mathematical models on small mt-CCVRP instances.

File	n	R	$\frac{\sum q_i}{Q \cdot R}$	0-1 MILP				0-1 MILP \star			
				LR	Cost	gap	Time	LR	Cost	gap	Time
test ₀₁	15	4	1.00	588.75	687.29	0.00	15.97	619.04	687.29	0.00	8.66
test ₀₂	15	4	1.25	605.08	741.91	0.00	77.63	633.47	741.91	0.00	33.99
test ₀₃	15	4	1.67	661.11	855.91	0.00	260.60	686.32	855.91	0.00	38.38
test ₀₄	15	4	2.50	836.57	1090.67	0.00	264.80	851.99	1090.67	0.00	24.20
test ₀₅	15	3	1.11	632.83	817.22	0.00	138.95	697.26	817.22	0.00	16.13
test ₀₆	15	3	1.33	643.13	942.45	0.00	2943.82	708.68	942.45	0.00	414.64
test ₀₇	15	3	1.67	674.25	(941.99/1008.03)	7.01	-	733.24	1008.03	0.00	560.17
test ₀₈	15	3	2.22	749.57	(1035.92/1111.44)	7.29	-	788.88	1111.04	0.00	124.54
test ₀₉	15	2	1.25	687.48	(1070.01/1201.30)	12.27	-	802.20	(1116.97/1182.66)	5.88	-
test ₁₀	15	2	1.67	702.20	(1059.41/1310.17)	23.67	-	814.44	(1100.45/1327.76)	20.66	-
test ₁₁	15	2	2.00	720.16	(1101.58/1403.74)	27.43	-	830.58	(1199.17/1391.60)	16.05	-
test ₁₂	15	2	2.50	761.58	(1164.97/1513.06)	29.88	-	862.40	(1296.97/1513.06)	16.66	-
Mean							8.96	2108.48		4.94	1301.73

Table 4.6 keeps the same first column of Table 4.5. Then, for each metaheuristic, the best solution value over five runs (Best cost), the average solution value (Avg cost), the percentage deviation of the best solution cost $Z(heur)$ to the optimum $Z(MILP)$

of CPLEX (or to the best lower bound when the solver must be stopped) computed as $\text{gap} = (Z(\text{heur})/Z(\text{MILP}) - 1) \times 100$, and the average running time per run in seconds (Time) are indicated.

Table 4.6: Results of metaheuristics on small mt-CCVRP instances.

File	n	R	$\frac{\sum q_i}{QR}$	MS-ILS				MA PM				MS-ELS			
				Best cost	Avg cost	gap	Time	Best cost	Avg cost	gap	Time	Best cost	Avg cost	gap	Time
test ₀₁	15	4	1.00	687.29	687.29	0.00	1.95	687.29	687.29	0.00	2.27	687.29	687.29	0.00	1.87
test ₀₂	15	4	1.25	741.91	741.91	0.00	2.76	741.91	741.91	0.00	2.35	741.91	741.91	0.00	2.54
test ₀₃	15	4	1.67	855.91	855.91	0.00	1.96	855.91	855.91	0.00	1.90	855.91	855.91	0.00	1.76
test ₀₄	15	4	2.50	1090.67	1090.67	0.00	1.41	1090.67	1090.67	0.00	1.40	1090.67	1090.67	0.00	1.24
test ₀₅	15	3	1.11	817.22	817.22	0.00	4.78	817.22	817.22	0.00	2.53	817.22	817.22	0.00	4.15
test ₀₆	15	3	1.33	942.45	942.45	0.00	4.28	942.45	942.45	0.00	2.55	942.45	942.45	0.00	3.98
test ₀₇	15	3	1.67	1008.03	1008.03	0.00	2.32	1008.03	1008.03	0.00	2.33	1008.03	1008.03	0.00	2.06
test ₀₈	15	3	2.22	1111.04	1111.04	0.00	2.53	1111.04	1111.04	0.00	1.82	1111.04	1111.04	0.00	2.38
test ₀₉	15	2	1.25	1182.66	1182.66	5.88	5.45	1182.66	1182.66	5.88	2.56	1182.66	1182.66	5.88	5.07
test ₁₀	15	2	1.67	1310.17	1310.17	19.06	3.69	1310.17	1310.17	19.06	2.40	1310.17	1310.17	19.06	3.16
test ₁₁	15	2	2.00	1391.60	1391.60	16.05	3.49	1391.60	1391.60	16.05	2.11	1391.60	1391.60	16.05	3.28
test ₁₂	15	2	2.50	1513.06	1513.06	16.66	1.57	1513.06	1513.06	16.66	1.97	1513.06	1513.06	16.66	1.11
Mean							4.80				2.18			4.80	2.72

The results indicate that the relaxed LP yields acceptable lower bounds, a good sign for a future study on valid inequalities. For the eight first instances, CPLEX finds an optimal solution. The model with valid inequalities outperforms the model without them: all linear relaxations are improved, two more optimal solutions are found in less than one hour, the gap is reduced when time limit is reached, and the running times are decreased when optimal solutions are found by both models. The instances look harder when the number of vehicles decreases and the average number of trips per vehicle increases: the running time of CPLEX augments quickly and the four last instances cannot be solved in one hour.

The three proposed metaheuristics always return a solution in at most 5.5 seconds. When optimal solutions are found by CPLEX (on the first 8 instance), the three metaheuristics have found the same solution. When time limit is reached by CPLEX (on the last 4 instances), the gaps are computed using the best lower bound found by CPLEX and the solutions found by the metaheuristics have the same values in all cases. These gap values for the last 4 instances are probably overestimated since solutions found by metaheuristics are always better than best integer solutions found by CPLEX and perhaps they are optimal but it is not possible to prove it. In addition, the columns average cost indicate that the same solution cost is found on the five runs.

The running times of the metaheuristics are similar. Nevertheless, the average running time is lower for the MA|PM than for the other two. Contrary to CPLEX, the running time is not affected significantly by the number of vehicles, but when the vehicle capacity increases. In that case, the average number of trips per vehicle $\left(\sum_{i=1}^n q_i\right)/(Q \cdot R)$ decreases, which impacts the complexity of neighborhoods 2-OPT on one trip, λ -interchanges on one trip, trip interchange and trip reordering.

4.7.6 Results on Larger Instances for the mt-CCVRP

In this subsection the three proposed metaheuristics (MS-ILS, MA|PM and MS-ELS) are compared on larger mt-CCVRP instances. Two results are shown for MS-ELS: MS-ELS¹⁰ refers to MS-ELS with ten children and 100 iterations while MS-ELS¹ refers to the same method with one child and 1 000 iterations. MS-ELS¹ is an MS-ILS but differs from the one presented in Section 4.3 because it operates in two solution spaces and applies the split procedure described in Section 4.4.

Tables 4.7 and 4.8 present the results for the CMT instances and GWKC instances respectively, using the same columns: instance name, number of nodes n , number of vehicles R , average number of trips per vehicle $\sum_{i=1}^n q_i/(Q \cdot R)$, best known solution BKS , and for each method (MS-ILS, MA|PM, MS-ELS¹ and MS-ELS¹⁰) deviation from the best known solution in 5 runs in percent (D_{best}), average deviation of the 5 solutions from best known solution in percent (D_{avg}) and average duration per run in seconds ($Time$). It can be recalled that the CMT and GWKC instances for the mt-CCVRP are identical to the CCVRP versions, except a reduced fleet size to get several trips per vehicle.

On the CMT instances, the average computational time is less than 6 minutes, varying between 0.54 and 14.55 minutes. MS-ILS is the slowest algorithm, its average computational time is 1.72 times the computational time of the fastest algorithm (MS-ELS¹). Nevertheless, its execution time for instance CMT₀₅ is lower than MA|PM execution time.

MA|PM is 50% slower than MS-ELS¹, and 34% slower than MS-ELS¹⁰. Although these three methods use the same VND algorithm and the same splitting procedure, these difference in the execution time can be explained by the children generation strategies and their impact on the VND precomputations: as MS-ELS is based on small changes on a single incumbent solution, children can reuse most of the precomputations of the incumbent

Table 4.7: Results on CMT instances (Christofides *et al.*, 1979) for the mt-CCVRP.

Instance	n	R	$\sum q_i$	$Q \cdot R$	BKS	MS-ILS		MA PM		MS-ELS ¹		MS-ELS ¹⁰				
						D_{best}	D_{avg}	Time	D_{best}	D_{avg}	Time	D_{best}	D_{avg}	Time	D_{best}	D_{avg}
CMT ₀₁	50	3	1.62	3856.39	0.00	0.16	78.26	0.00	0.00	33.97	0.00	0.14	34.62	0.00	0.14	32.61
CMT ₀₂	75	3	3.25	8300.15	0.00	0.04	68.37	0.00	0.02	57.94	0.00	0.01	47.59	0.00	0.03	48.22
CMT ₀₃	100	3	2.43	10957.00	0.00	0.42	238.07	0.37	0.57	169.01	0.40	0.63	144.66	0.00	0.45	139.25
CMT ₀₄	150	3	3.73	20595.93	0.01	0.25	479.34	0.46	0.57	446.92	0.00	0.07	316.62	0.00	0.21	329.71
CMT ₀₅	199	3	5.31	33981.40	0.18	0.48	750.80	0.61	0.85	872.77	0.16	0.42	502.71	0.00	0.97	551.14
CMT ₁₁	120	3	2.29	15797.40	0.00	0.17	470.41	0.17	0.19	326.44	0.00	0.11	217.54	0.00	0.14	275.24
CMT ₁₂	100	3	3.02	10658.70	0.00	0.00	329.87	0.00	0.09	198.21	0.00	0.00	137.85	0.00	0.00	149.42
Average					0.03	0.22	345.02	0.23	0.33	300.75	0.08	0.20	200.23	0.00	0.28	224.46

Table 4-8: Results on GWKC instances (Golden *et al.*, 1998) for the mt-CCVRP.

Instance	n	R	$\sum \frac{q_i}{Q \cdot R}$	BKS	MS-ILS		MA PM		MS-ELS ¹		MS-ELS ¹⁰					
					D_{best}	D_{avg}	D_{best}	D_{avg}	D_{best}	D_{avg}	D_{best}	D_{avg}	Time	Time		
GWKC ₀₁	240	4	2.18	122885.56	0.41	0.27	2443.14	1.32	1.74	3215.87	0.52	0.29	1376.02	0.00	0.35	1530.17
GWKC ₀₂	320	4	2.29	254443.84	0.06	0.23	5330.64	1.24	1.39	6953.87	0.00	0.19	3624.22	0.00	0.22	3647.97
GWKC ₀₃	400	4	2.22	434710.60	0.25	0.63	10123.85	1.65	1.82	13670.02	0.77	0.83	8536.12	0.00	0.24	7779.81
GWKC ₀₄	480	4	2.40	675615.36	0.16	0.44	15887.69	1.67	1.81	22263.42	0.18	0.71	13478.36	0.00	0.66	14041.62
GWKC ₀₅	200	3	1.48	193339.55	0.00	0.15	3018.47	0.29	0.60	2124.88	0.07	0.43	1685.89	0.04	0.32	1799.27
GWKC ₀₆	280	3	2.07	335034.88	0.00	0.22	5846.47	0.28	0.80	5675.23	0.12	0.29	3381.19	0.02	0.26	3658.15
GWKC ₀₇	360	3	2.67	493826.53	0.00	0.34	9144.21	1.23	1.44	10412.00	0.15	0.47	7451.69	0.15	0.55	7301.47
GWKC ₀₈	440	5	1.96	399995.84	0.00	0.42	13280.66	0.99	1.63	19741.19	0.17	0.45	11153.84	0.51	0.60	11017.77
GWKC ₀₉	255	5	2.69	13752.04	0.23	0.32	1433.82	1.27	1.76	2592.28	0.00	0.39	993.54	0.35	0.51	1026.94
GWKC ₁₀	323	6	2.53	18410.68	0.18	0.65	2374.63	1.68	2.03	4290.85	0.00	0.52	1812.07	0.36	0.61	1905.67
GWKC ₁₁	399	7	2.43	24239.20	0.13	0.33	3920.85	2.02	2.21	8647.16	0.40	0.42	3565.39	0.00	0.57	3525.50
GWKC ₁₂	483	8	2.34	31144.78	0.00	0.68	5831.21	1.55	1.89	13320.34	0.08	0.61	5785.64	0.03	0.55	6210.10
GWKC ₁₃	252	10	2.51	10056.24	0.25	0.27	913.49	1.31	1.54	1639.02	0.13	0.36	492.32	0.00	0.27	541.42
GWKC ₁₄	320	12	2.39	13413.98	0.32	0.25	1687.19	0.54	0.93	2842.03	0.03	0.13	975.29	0.00	0.21	1031.94
GWKC ₁₅	396	15	2.15	16385.79	0.14	0.16	2263.48	0.92	1.12	5175.61	0.00	0.13	1884.62	0.14	0.19	1917.32
GWKC ₁₆	480	15	2.39	24188.15	0.00	0.39	3737.54	0.07	1.38	8830.69	0.25	0.86	3426.15	0.23	0.79	3320.24
GWKC ₁₇	240	10	2.16	6618.18	0.00	0.63	791.04	2.09	2.16	1470.16	0.29	0.68	433.94	0.25	0.62	443.48
GWKC ₁₈	300	12	2.25	9385.02	0.00	0.43	1265.00	1.57	1.71	2618.37	0.27	0.64	812.46	0.27	0.70	789.69
GWKC ₁₉	360	12	2.70	14894.56	0.00	0.33	2032.59	2.08	2.15	4508.08	0.72	0.89	1283.22	0.79	1.14	1352.93
GWKC ₂₀	420	15	2.52	17939.90	0.00	0.49	2554.13	2.26	2.50	6807.04	0.49	0.77	1946.51	0.64	1.05	2078.44
Average					0.11	0.38	4694.01	1.30	1.63	7339.91	0.23	0.50	3654.92	0.19	0.52	3745.99

solution and need less VND moves to find a local optima solution, while children in MA|PM do not reuse precomputations from parents and can be far from local optima solutions. The number of children in MS-ELS has an impact on the computational time: from $nchildren = 1$ to $nchildren = 10$ the computational time increases 12%. MS-ELS¹ is, on average, the fastest algorithm, nevertheless, MS-ELS¹⁰ presents the best times for two instances: CMT₀₁ and CMT₀₃. A comparison with Table 4.3 (CCVRP) shows that the running time for MS-ILS is roughly multiplied by 2.5 when multitrips must be handled.

With five runs, MS-ILS finds five out seven best known solutions. It fails on the two largest instances which also have the largest average number of trips. Nevertheless, MS-ILS solutions are very close to best known solutions since its deviation is 0.03%. MA|PM finds three best known solutions and its average deviation (0.23%) indicates that its solutions are close to best known solutions.

MS-ELS¹⁰ always finds the best known solution (with five runs). When the number of children is reduced the method does not find two of them and the average deviation from the best known solution increase from 0% to 0.08%. The average costs for five runs is close to the best cost on the four approaches (0.22%, 0.33%, 0.20% and 0.28%, respectively), indicating that they are robust.

The average deviations from best known solutions for MS-ILS, MS-ELS¹ and MS-ELS¹⁰ increase moderately on GWKC instances, with 0.11%, 0.23% and 0.19%, respectively. For MA|PM algorithm this increasing is more pronounced (1.30%).

MS-ILS finds 10 best known solutions and its average deviation is 0.11%. The average computational time is 78 minutes, ranging from 13 minutes to 265 minutes. The proposed MS-ILS is still relatively stable in terms of solution quality but its running time becomes important on the largest instances with 480 demand nodes.

MA|PM has the poorest performance. Its average deviation is 11.8 times the deviation in MS-ILS, and the computational time is 1.5 times larger. MA|PM has not found any best known solution.

MS-ELS still being the fastest of the proposed methods, around 60 minutes (22% faster than MS-ILS) for $nchildren = 1$ and 62 minutes (20% faster than MS-ILS) when $nchildren = 10$. Nevertheless, their average deviation are now greater, 0.23% and 0.19%

respectively, compared with 0.11% in MS-ILS. MS-ELS¹ finds four best known solutions and MS-ELS¹⁰ finds seven, with five runs per instance.

The deviation from the best solution found (D_{best}) from MS-ELS¹⁰ is larger than the D_{best} from MS-ILS due to the values of a few instances. For instance, if instances GWKC₀₈, GWKC₁₉ and GWKC₂₀ are ignored, D_{best} from MS-ELS¹⁰ becomes 0.11% while D_{best} from MS-ILS becomes 0.13%.

Moreover, the execution times vary a lot among instances of the same size: for example, instances GWKC₀₁ and GWKC₁₇ contain 240 demand nodes but require respectively 2 443 and 791 seconds of processing time by MS-ILS metaheuristic. This comparison remains similar for MA|PM, MS-ELS¹ and MS-ELS¹⁰ metaheuristics. This variation is mainly due to the neighborhood N_1 in which in the VND spends most of its time. Indeed, it can be seen in Subsection 4.3.5 that $|N_1| = O(n^2/m)$, where the number of trips m can be computed as $m = R \cdot \left(\sum_{i=1}^n q_i / (Q \cdot R) \right)$: GWKC₀₁ and GWKC₁₇ respectively require $m = 8.72$ and $m = 21.6$ trips. Similar analysis can be done with different subsets of instances. Note that grater number of trips with the same (or similar) number of required nodes implies a greater average number of nodes per trip.

Figure 4.13 shows the evolution of the average deviation to the best known solution during the iterations of MS-ILS, on the CMT instances, using a logarithmic scale for the deviation axis. For each number of iterations, the deviation reported is the average of 35 values (7 instances and 5 runs per instance). The initial deviation is 7.78% for the first local optimum, obtained by the greedy randomized heuristic followed by one call to the VND. A major part of the descent is accomplished using the allocated budget of 5 000 iterations: the deviation reduces to 0.22%. Additional iterations bring only a minor improvement, at the expense of larger running times.

Table 4.9 details the percentage deviations for each instance and adds the deviation of the initial heuristic solution. As it can be observed, the deviation is below 1% (0.77%) after 1 000 iterations versus 0.22% after 5 000. Hence, a first way to reduce the large running times on GWKC instances (up to 15 000 seconds) is to use 1 000 iterations only: in that case no running time would exceed 50 minutes. A much faster option is to run the initial constructive heuristic and improve it using a single call to the VND, which gives an average deviation of 7.78% in at most 3 seconds per run, on all instances tested.

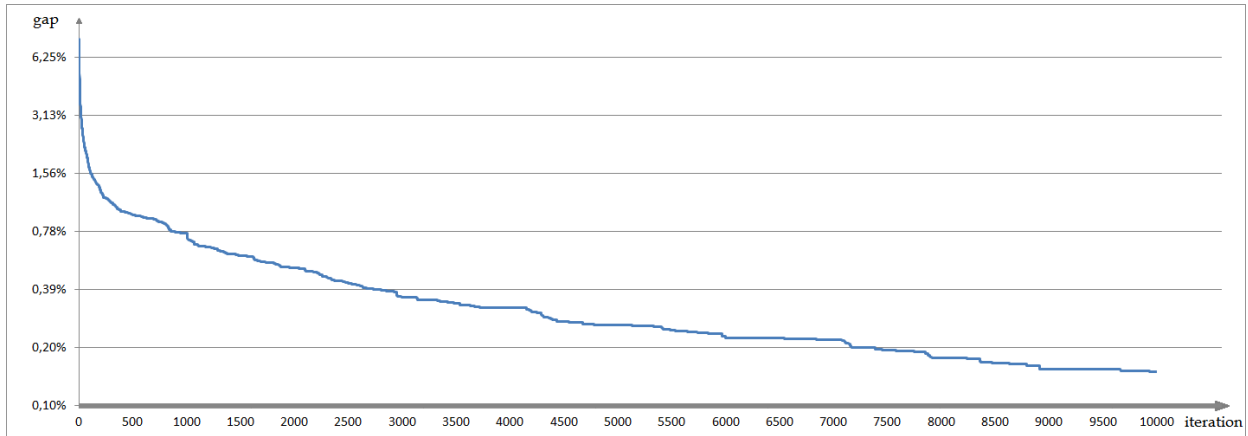


Figure 4.13: Evolution of average deviation to best known solution on CMT instances.

Table 4.9: Improvement of initial solution (average of 5 runs).

Instance	Initial gap to BS	Gap after 1st call to VND	Gap after 1 000 calls	Gap after 5 000 calls
CMT ₀₁	19.61	4.12	0.35	0.16
CMT ₀₂	36.08	8.20	0.36	0.04
CMT ₀₃	35.13	6.41	1.02	0.42
CMT ₀₄	43.38	8.93	0.55	0.25
CMT ₀₅	39.12	9.02	1.80	0.48
CMT ₁₁	63.30	7.40	0.42	0.17
CMT ₁₂	53.52	10.39	0.88	0.00
Average	41.45	7.78	0.77	0.22

4.7.7 Comparison between CCVRP and mt-CCVRP solutions

A comparison between the best known solutions of both problems, CCVRP and mt-CCVRP, allows to evaluate the impact of the fleet size on the average arrival time. Tables 4.10 and 4.11 compare the main features of each instance and the resulting average arrival time at affected sites. The first two columns indicate the instance name and the number of sites n . For each problem, CCVRP and mt-CCVRP, three columns are presented:

the number of vehicles R , the average vehicle load (CCVRP) or average number of trips (mt-CCVRP) computed as $\sum_{i=1}^n q_i / (Q \cdot R)$, and the average arrival time \bar{t} computed as the objective function value of the best known solution in the corresponding instance-problem divided by the number of nodes ($\bar{t} = BKS/n$).

Table 4.10: Comparison of the average arrival times of CCVRP and mt-CCVRP for CMT instances (Christofides *et al.*, 1979).

Instance	n	CCVRP			mt-CCVRP		
		R	$\frac{\sum q_i}{Q \cdot R}$	\bar{t}	R	$\frac{\sum q_i}{Q \cdot R}$	\bar{t}
CMT ₀₁	50	5	0.97	44.61	3	1.62	77.13
CMT ₀₂	75	10	0.98	31.89	3	3.25	110.67
CMT ₀₃	100	8	0.91	40.45	3	2.43	109.57
CMT ₀₄	150	12	0.93	33.25	3	3.73	137.31
CMT ₀₅	199	17	0.94	29.19	3	5.31	170.76
CMT ₁₁	120	7	0.98	60.95	3	2.29	131.65
CMT ₁₂	100	10	0.91	35.59	3	3.02	106.59
Mean			0.95	39.42		3.09	120.52

Table 4.10 compares the CMT instances. The fleet size has been reduced between 40% and 82% in mt-CCVRP. This reduction implies the increasing of the average arrival time between 73% and 485%. The average arrival time for the CCVRP instances is 39.42 time units, while for mt-CCVRP is 120.52 (3 times greater). The increasing in the average arrival time is greater when the factor $\sum_{i=1}^n q_i / (Q \cdot R)$ becomes greater.

In Table 4.11 the values of the average number of trips are between 1.48 and 2.70. Here the fleet size is reduced between 40% and 64%, and the average arrival time increases between 69% and 191%. On average, the average arrival time increases 2.40 times from single-trip to multitrip version.

Table 4.11: Comparison of the average arrival times of CCVRP and mt-CCVRP for GWKC instances (Golden *et al.*, 1998).

Instance	n	CCVRP			mt-CCVRP		
		R	$\frac{\sum q_i}{Q \cdot R}$	\bar{t}	R	$\frac{\sum q_i}{Q \cdot R}$	\bar{t}
GWKC ₀₁	240	9	0.97	228.28	4	2.18	512.02
GWKC ₀₂	320	10	0.92	314.44	4	2.29	795.14
GWKC ₀₃	400	10	0.89	427.57	4	2.22	1 086.78
GWKC ₀₄	480	10	0.96	545.94	4	2.40	1 407.53
GWKC ₀₅	200	5	0.89	570.82	3	1.48	966.70
GWKC ₀₆	280	7	0.89	501.54	3	2.07	1 196.55
GWKC ₀₇	360	8	1.00	501.34	3	2.67	1 371.74
GWKC ₀₈	440	10	0.98	441.14	5	1.96	909.08
GWKC ₀₉	255	14	0.96	18.53	5	2.69	53.93
GWKC ₁₀	323	16	0.95	20.79	6	2.53	57.00
GWKC ₁₁	399	18	0.95	23.09	7	2.43	60.75
GWKC ₁₂	483	19	0.99	25.93	8	2.34	64.48
GWKC ₁₃	252	26	0.97	14.40	10	2.51	39.91
GWKC ₁₄	320	29	0.99	16.30	12	2.39	41.92
GWKC ₁₅	396	33	0.98	17.70	15	2.15	41.38
GWKC ₁₆	480	37	0.97	19.27	15	2.39	50.39
GWKC ₁₇	240	22	0.98	12.77	10	2.16	27.58
GWKC ₁₈	300	27	1.00	14.07	12	2.25	31.28
GWKC ₁₉	360	33	0.98	15.34	12	2.70	41.37
GWKC ₂₀	420	38	0.99	17.20	15	2.52	42.71
Mean			0.96	187.32		2.32	439.91

4.8 Concluding Remarks

This chapter introduces a new problem, the multi-trip cumulative capacitated vehicle routing problem (mt-CCVRP). This problem constitutes a good way to model the delivery

of relief supplies after a humanitarian disaster, where the number of vehicles is limited and the time to reach affected areas is critical.

Four mathematical models for the single-trip version (CCVRP) have been proposed and compared with the one found in the literature. The best results are obtained by the model CCVRP3 which is generalized to the multitrip case. The mathematical model for the mt-CCVRP, a 0–1 MILP without vehicle nor multitrip indexes, is proposed and tested on small instances.

The chapter develops also three metaheuristics to solve the mt-CCVRP: MS-ILS, MA|PM and MS-ELS. MS-ILS hybridizes a multi-start ILS with a VND which evaluates each move in constant time. Five moves are proposed, some precomputations speed up the VND, and a dominance rule described in Chapter 3 avoids to consider multiple orderings for multitrips. The same precomputations and dominance rule, with similar neighborhood structures are used by MA|PM and MS-ELS.

The chapter also presents an adapted two-level split procedure which considers the cumulative objective (the sum of arrival times) and the multiple trips. This procedure solves optimally a shortest path problem in polynomial time complexity where the first level defines the set of nodes in each multitrip, and the cost of each multitrip is computed by solving the second level split which defines the different trips of each multitrip.

On small instances, the resulting metaheuristics find the same results as the mathematical model when the latter can find the optimal solution. They are able to produce competitive results in relatively short computing times.

MS-ILS competes with published methods for the case without multitrips, the CCVRP. Although the running time becomes important on the largest instances close to 500 demand nodes, it has been shown that the fast descent can be truncated to get good solutions in a reduced duration.

A comparison between three metaheuristics is performed for larger mt-CCVRP instances. The results show the efficacy of the proposed splitting procedure and the developed moves, as well as impact of the fleet size on the average arrival time.

Conferences and Publications

Preliminary results of the proposed methods have been presented in one national conference (ROADEF 2012) and four international conferences:

- J. C. Rivera, H. M. Afsar and C. Prins. Une méthode ILS pour le problème de tournées de véhicules cumulatives avec routes multiples. 13ème congrès annuel de la Société française de recherche opérationnelle et d'aide à la décision, ROADEF 2012. Angers, France. April, 2012.
- J. C. Rivera, H. M. Afsar and C. Prins. An MS-ILS algorithm for the multi-trip cumulative vehicle routing problem. Workshop on combinatorial optimization, routing and location, CORAL 2012. Benicàssim, Spain. May, 2012.
- C. Prins, J. C. Rivera and H. M. Afsar. A mathematical model and a metaheuristic for the mt-CCVRP. Network Optimization Workshop, NOW 2013. Sicily, Italy. June, 2013.
- J. C. Rivera, H. M. Afsar and C. Prins. Split-based metaheuristic for the multitrip cumulative capacitated vehicle routing problem (mt-CCVRP). Vehicle Routing and Logistics Optimization, VeRoLog 2013. Southampton, England. July, 2013.
- J. C. Rivera, H. M. Afsar and C. Prins. Multistart evolutionary local search for a disaster relief problem. Artificial Evolution, EA 2013. Bordeaux, France. November, 2013.

A full-paper version with results based on MS-ILS metaheuristic has been accepted at Computational Optimization and Applications:

- J. C. Rivera, H. M. Afsar and C. Prins. A Multistart Iterated Local Search for the Multitrip Cumulative Capacitated Vehicle Routing Problem. To appear in Computational Optimization and Applications.

A paper comparing MS-ILS and MS-ELS has been accepted to be published, for a post-conference book with selected papers, in Lecture Notes in Computer Science (LNCS):

- J. C. Rivera, H. M. Afsar and C. Prins. Multistart Evolutionary Local Search for a Disaster Relief Problem. Lecture Notes in Computer Science. *Accepted*.

A technical report about MS-ILS for mt-CCVRP is available online: http://losi.utt.fr/fr/publications/cahier_du_losi_3.html (Technical report No. UTT-LOSI-13001).

Chapter 5

The Cumulative Capacitated Vehicle Routing Problem with Indirect Deliveries

5.1 Introduction

In this chapter a new vehicle routing problem is studied. This problem is raised by the relief aid delivery in the response phase of humanitarian logistics. As in previous chapters, the objective function is the sum of arrival times, but here vehicles can perform one single trip and not all sites need to be visited. The relief aid for unvisited sites must be distributed by an auxiliary fleet from a visited site. This problem is named the *cumulative capacitated vehicle routing problem with indirect deliveries* (CCVRP-ID).

The aim of this new problem is to reduce the average arrival time at sites by using different transportation modes. As mentioned in previous chapters, early arrival times at communities can increase the survival rate of the population. So communities close to each other can be served from a common location, supplied by a fleet of vehicles (regular fleet), based at the depot and then, if necessary, relief aid is distributed to unvisited communities by smaller auxiliary vehicles, better suited for short distances.

This approach reduces the average arrival time at sites. For instance, when no auxiliary vehicles are used, the solution is a CCVRP solution, but auxiliary vehicles can improve the distribution system performance while preserving its feasibility. For the sequence $\sigma = (1, \dots, i - 1, i, i + 1, \dots, M_\sigma)$ of M_σ required sites visited by a regular vehicle, if the transportation mode to reach site i from site $i - 1$ is changed by a slower auxiliary vehicle,

it is clear that the arrival time at site i is increased. This increased arrival time at site i can be compensated by a decrease in the arrival times at sites $i + 1$ and all the $M_\sigma - i$ sites after site i in the sequence, especially when the triangle inequality holds, since site $i + 1$ is visited from site $i - 1$ by the regular vehicle.

The problem is formulated as an integer linear program, based on the formulation CCVRP3 presented in Subsection 4.2.1. Additionally, a hybrid MS-ELS metaheuristic calling a VND procedure is proposed to solve it. This metaheuristic is inspired by the MS-ILS presented in Section 4.3, but with important modifications to handle the new problem features (unvisited sites and two transportation modes). Evolutionary local search strategy is applied instead of iterated local search because it has a more general structure which includes ILS as special case.

There are similarities with other VRPs such as the Generalized VRP (GVRP), the Selective VRP, and the two-echelon vehicle routing problem, and also problems with covering considerations like the Ring-Star problem and the covering tour problem, where the sites can be serviced with undirected arcs.

In the GVRP, studied by Pop *et al.* (2012), Pop *et al.* (2013) and Afsar *et al.* (2014), a subset of sites must be visited but, contrary to the CCVRP-ID, the sites are grouped in mutually exclusive geographical zones, auxiliary vehicles are not used, there is no cost for the unvisited sites, only one site per group is visited, and the objective function is the classical total travel time. Compared to GVRP, the selective VRP does not group the sites, but all unvisited sites must be within a given maximal distance to a visited site (Valle *et al.*, 2009, 2011).

The ring-star problem (RSP) consists in finding a subgraph that can be decomposed into a cycle (or ring) and a set of edges with exactly one vertex in the cycle. Simonetti *et al.* (2011) propose a branch-and-cut algorithm where the cost is computed as the sum of the cost of its edges. Calvete *et al.* (2013) have designed an evolutionary algorithm based on a bilevel formulation of the problem, where the solution cost is computed as the sum of the cost of installing facilities on the nodes on the cycle, the cost of connecting them and the cost of assigning the nodes not on the cycle to their closest node on the cycle. The goal of the Capacitated m -Ring-Star Problem (CmRSP) is to find m cycles visiting a central depot and a subset of demand nodes, while nodes not belonging to any cycle must be allocated to a visited node. Additionally the number of nodes visited or allocated to a

cycle cannot be greater than a given capacity. Solution approaches for the $CmRSP$ include the VNS metaheuristics developed by Salari *et al.* (2010b) and Naji-Azimi *et al.* (2012b), the ILS proposed by Naji-Azimi *et al.* (2010) and the branch-and-cut algorithm of Berinsky & Zabala (2011). the goal in the bi-objective RSP (B-RSP) introduced by Liefoghe *et al.* (2010) is to minimize the cycle cost and the allocation cost in two separate objectives and using four evolutionary algorithms based on iterative local search, IBEA (Indicator-Based Evolutionary Algorithm), NSGA-II (Non-Dominated Sorting Genetic Algorithm II) and a simple elitist EA.

In the two-echelon vehicle routing problem (2E-VRP) (Crainic *et al.*, 2010; Nguyen *et al.*, 2012a,b) goods are delivered from depot to intermediary satellites by first-level routes, while second-level routes deliver demands from satellites to customers. A special case is the truck and trailer routing problem (TTRP) or vehicle routing problem with trailers and transshipments (VRPTT), where trucks equipped with trailers may drop them to perform second-level trips (Villegas *et al.*, 2010; Drexler, 2013; Villegas *et al.*, 2013). In TTRP customers can be restricted to be visited only by trucks.

The CCVRP-ID proposed in this chapter is in fact a Cumulative Capacitated m -Ring-Star Problem where the cost of a ring-edge in a solution depends on its length and the number of nodes after it, the allocation costs are based on the edge length and a parameter to model the velocity of auxiliary vehicles, and the sum of the node demands in a cycle must fit the vehicle capacity.

The chapter is structured as follows: Section 5.2 presents the problem definition and a mixed integer program. A multistart evolutionary local search metaheuristic is proposed in Section 5.3. Results are described in Section 5.4 and concluding remarks are presented in Section 5.5.

5.2 Problem Definition and Mixed Integer Linear Model

The cumulative capacitated vehicle routing problem with indirect deliveries (CCVRP-ID), like the CCVRP and the mt-CCVRP, uses a fixed size fleet to deliver relief aid to minimize the sum of arrival times at required sites. In this new version, two options are possible to deliver relief aid: sites can be visited during trips or, when it is suitable they can be allocated to a visited close site to be delivered indirectly via an auxiliary transportation mode. When a second transportation mode is used, the regular fleet does not need to visit all sites and

can cover a larger region in a faster way. In our mind, the indirect deliveries can be done in the simplest case by pedestrians or personal vehicles coming to the closest site reached by one vehicle, which explains that we do not consider vehicle capacities for the second transportation mode.

The problem can be defined on an undirected complete graph $G = (V, E)$. The node-set $V = \{0, \dots, n\}$ includes a depot-node 0 and a subset $V' = V \setminus \{0\}$ of n affected sites or required sites. The arc-set E is composed of arcs (i, j) with travel times w_{ij} . A set of R vehicles of capacity Q are based at the depot. Each site $i \in V'$ has a known demand q_i . It is assumed without loss of generality that $R \leq n$, $q_i \leq Q \forall i \in V'$ and $\sum_{i \in V'} q_i \leq Q \cdot R$.

The objective is to identify a set of trips and their allocated sites such that each site is visited at most once, unvisited sites are supplied from a visited site by using an auxiliary slower vehicle, and the sum of arrival times at sites, visited and allocated, is minimized. The travel time w_{ij} is the time required to traverse arc (i, j) by a normal vehicle, while $\delta \cdot w_{ij}$ is the time required by an auxiliary vehicle. The parameter δ is the factor which indicates how many times the auxiliary vehicles are slower than vehicles in the regular fleet. A trip is defined as a circuit, starting and ending at the depot, with at least one visited site, and a set of allocated sites, such that the total demand serviced, including visited and allocated sites, does not exceed vehicle capacity Q . In our version, a site cannot be allocated to the depot.

A CCVRP-ID instance example is presented in Figure 5.1. In this example, $n = 10$ required sites with demands $q_i = 10 \forall i \in V'$ are covered by $R = 2$ vehicles of capacity $Q = 60$. Continuous arrows indicate the arcs traversed by the vehicles. The travel times w_{ij} are given on every arc. The dashed arcs (i, j) mean that site j is allocated to site i with travel times $\delta \cdot w_{ij}$, using $\delta = 2$. The arrival times are written near each node. The cost solution is $Z = 180$. Note that more than one site can be allocated to a visited site, for instance sites 7 and 8 are here allocated to site 1.

It can also be noted that the CCVRP is a special case where no allocations are present. That also implies that CCVRP-ID solutions are lower bounds for the CCVRP. Figure 5.2 compares optimal solutions for CCVRP and CCVRP-ID for an instance with $n = 15$, $R = 1$ and $\delta = 2.5$. As it can be seen, the CCVRP solution has a greater objective function than the CCVRP-ID solution.

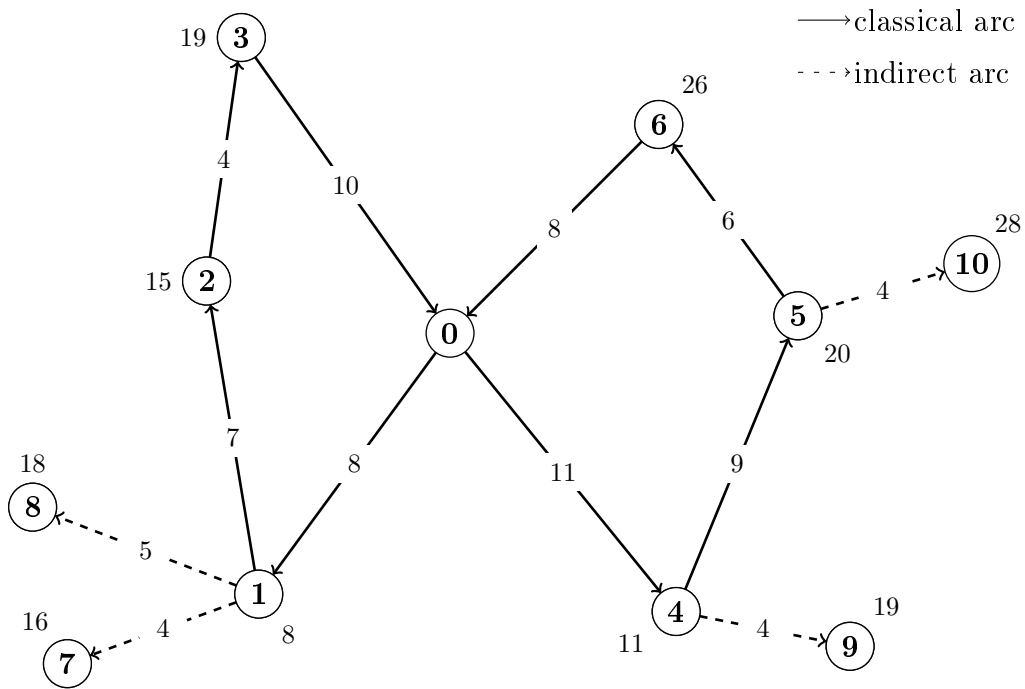


Figure 5.1: Example of a CCVRP-ID solution for an instance with $n = 10$ and $R = 2$.

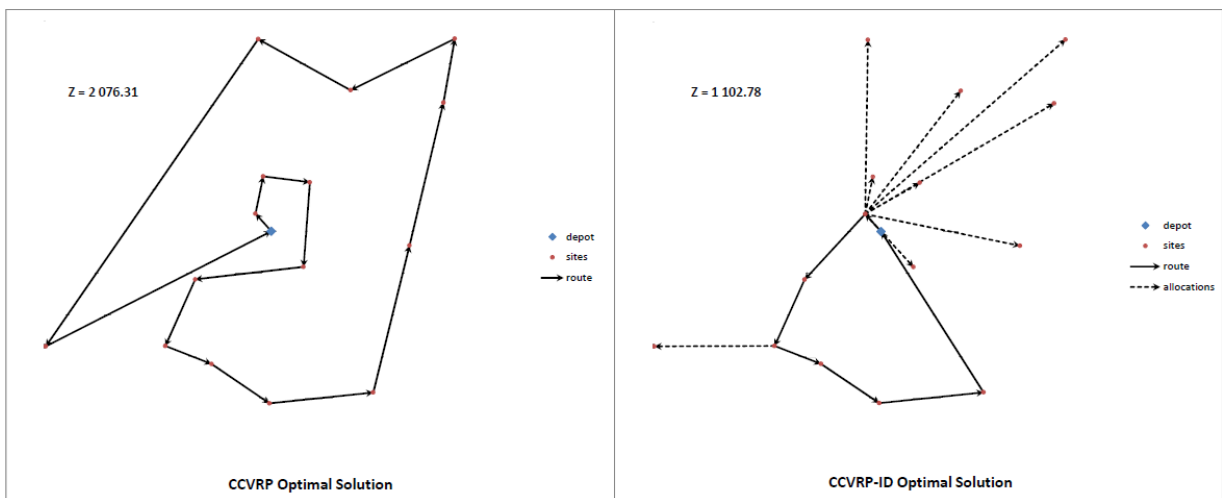


Figure 5.2: Comparison between optimal solutions for the CCVRP and the CCVRP-ID.

In the following, a 0–1 mixed integer linear program (MILP) for the CCVRP-ID is described. This formulation uses four sets of decision variables. As in models CCVRP3 and mt-CCVRP in Chapter 4, a binary variable x_{ij} indicates if arc (i, j) is traversed ($x_{ij} = 1$) or not ($x_{ij} = 0$), y_{ij} gives its arc coefficient and F_{ij} is the total load on that arc. A new set of assignment variables Ψ_{ij} indicates if site j is allocated to site i . The resulting mathematical model is given by Equations (5.1) to (5.14).

Equation (5.1) represents the objective function, the sum of arrival times. Note that decision variable y_{ij} in the first term gives the number of times the arc cost w_{ij} is counted in the objective function when that arc is traversed by a vehicle, but indirect arc costs $\delta \cdot w_{ij}$ in the second term are counted once. Equation (5.2) sets the number of available vehicles. Constraints (5.3) indicate that every site must be visited once ($x_{ij} = 1$) or allocated once ($\Psi_{ij} = 1$). Equations (5.4) imply that a vehicle arriving a site must leave it.

$$\min Z = \sum_{i \in V} \sum_{j \in V'} w_{ij} \cdot y_{ij} + \sum_{i \in V'} \sum_{j \in V'} \delta \cdot w_{ij} \cdot \Psi_{ij} \quad (5.1)$$

$$\sum_{j \in V'} x_{0j} = R, \quad (5.2)$$

$$\sum_{i \in V} x_{ij} + \sum_{i \in V'} \Psi_{ij} = 1, \quad \forall j \in V' \quad (5.3)$$

$$\sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji}, \quad \forall i \in V' \quad (5.4)$$

$$F_{ij} \leq Q \cdot x_{ij}, \quad \forall i \in V, j \in V' \quad (5.5)$$

$$\sum_{j \in V} F_{ji} - \sum_{j \in V'} F_{ij} = q_i \cdot \left(1 - \sum_{j \in V'} \Psi_{ji}\right) + \sum_{j \in V'} q_j \cdot \Psi_{ij}, \quad \forall i \in V' \quad (5.6)$$

$$y_{ij} \leq (n - R + 1) \cdot x_{ij}, \quad \forall i \in V, j \in V' \quad (5.7)$$

$$\sum_{j \in V} (y_{ji} - y_{ij}) = \left(1 - \sum_{j \in V'} \Psi_{ji}\right) + \sum_{j \in V'} \Psi_{ij}, \quad \forall i \in V' \quad (5.8)$$

$$y_{ij} \geq x_{ij}, \quad \forall i \in V, j \in V' \quad (5.9)$$

$$y_{ij} \geq 2 \cdot x_{ij} - x_{j0}, \quad \forall i \in V, j \in V' \quad (5.10)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V, i \neq j \quad (5.11)$$

$$\Psi_{ij} \in \{0, 1\}, \quad \forall i, j \in V, i \neq j \quad (5.12)$$

$$y_{ij} \geq 0, \quad \forall i, j \in V, i \neq j \quad (5.13)$$

$$F_{ij} \geq 0, \quad \forall i, j \in V, i \neq j \quad (5.14)$$

The total vehicle load on each arc is limited by the Equations (5.5) and (5.6). Constraints (5.5) limit the maximum load to vehicle capacity Q . Constraints (5.6) reduce the load when a site i is visited. The first two terms in Constraints (5.6) indicate the incoming flow and the outgoing flow at visited site i , respectively. The first term in the right-hand side expresses that the demand of site i is included in the flow difference, except if site i is allocated to site

j ($\Psi_{ji} = 1$ for a site $j \in V'$). The last term adds the demands of the allocated sites at site i to the flow difference.

Similarly, Equations (5.7) and (5.8) define arc coefficient values. Constraints (5.7) limit the maximum arc coefficient, while Constraints (5.8) reduce the arc coefficients when sites are visited. The difference between arc coefficients of input and output arcs is decreased by one on each visited site i . This difference is reduced to zero if site i is not visited ($\Psi_{ji} = 1$ for a site $j \in V'$), or it is increased by one for every allocated site j at site i ($\Psi_{ij} = 1$).

The valid inequalities of the mt-CCVRP model remain valid for the CCVRP-ID, where they are defined by Constraints (5.9) and (5.10). The domains of the decision variables are specified by Equations (5.11) to (5.14).

5.3 Multistart Evolutionary Local Search

As explained in Section 4.6, ELS is a metaheuristic which generalizes ILS by generating several children instead of one at every iteration, and it has been successfully applied to a number of vehicle routing problems.

In this section, a multistart ELS (MS-ELS) is applied to the CCVRP-ID, and its components, which are adapted to the new considered features, are described.

5.3.1 Principle and General Structure

The metaheuristic is described by the Algorithm 5.1.

A number (*MaxStart*) of successive randomized greedy solutions are built by the procedure *Greedy_Randomized_Heuristic* (line 3), which is based on best insertion heuristics. Each resulting solution undergoes a *VND* (line 5) metaheuristic and is used as initial solution for an ELS of *MaxIter* iterations.

In Each iteration of the current ELS, *MaxChildren* child-solutions S'' are generated by a *Mutation* procedure (line 11). The procedures *VND* (line 13) and *Post_Optimization* (line 14) are used to improve the quality of each child-solution.

The best child-solution is recorded in S' (lines 15 to 17). Once the children have been generated, the best-child can update the incumbent solutions in case of improvement (lines

19 to 21). At the end of the current ELS, the global best solution S^* is also updated when improved (lines 23 to 25).

The procedures *Precomputations* (after each randomized construction) and *Update_Precomputations* (after perturbation and during the procedures VND and Post_Optimization (Algorithm 5.2)), are used to speed up the local moves performed by the VND. The VND has the same general structure as in Algorithm 4.2 but includes new moves described later.

Algorithm 5.1 – MS-ELS for the CCVRP-ID

```
1:  $Z_{S^*} \leftarrow \infty$ 
2: for  $Start \leftarrow 1$  to  $MaxStart$  do
3:   Greedy_Randomized_Heuristic ( $S$ )
4:   Precomputations ( $S$ )
5:   VND ( $S$ )
6:   for  $Iter \leftarrow 1$  to  $MaxIter$  do
7:      $Z_{S'} \leftarrow \infty$ 
8:      $S' \leftarrow S$ 
9:     for  $Child \leftarrow 1$  to  $MaxChildren$  do
10:       $S'' \leftarrow S'$ 
11:      Perturbation ( $S''$ )
12:      Update_Precomputations ( $S''$ )
13:      VND ( $S''$ )
14:      Post_Optimization ( $S''$ )
15:      if  $Z_{S''} < Z_{S'}$  then
16:         $S' \leftarrow S''$ 
17:      end if
18:    end for
19:    if  $Z_{S'} < Z_S$  then
20:       $S \leftarrow S'$ 
21:    end if
22:  end for
23:  if  $Z_S < Z_{S^*}$  then
24:     $S^* \leftarrow S$ 
25:  end if
26: end for
```

The components of the metaheuristic are described in the sequel.

5.3.2 Solution Representation

Like in the MS-ILS metaheuristic in Subsection 4.3.2, a solution is encoded as R lists of nodes, with one list per vehicle or route. Each list indicates the order in which the required nodes are visited by the vehicles, using the depot (node 0) at the start and at the end of each list. Each node can represent a single-node, or a super-node grouping a visited site and its allocated sites. The sites allocated to a node i are stored in a set (A_i). For instance, the routes in Figure 5.1 can be represented as $k_1 = (1, 2, 3)$ and $k_2 = (4, 5, 6)$, where the corresponding sets are $A_1 = (7, 8)$, $A_4 = (9)$ and $A_5 = (10)$.

5.3.3 Precomputations

The same precomputations as in Chapter 4 are used here to compute some moves in constant time in VND procedure. Most of the moves in the proposed VND can be expressed by the concatenation operator \oplus defined in Subsection 4.3.3.

Equations (5.15) to (5.17) remind the equations (4.68) to (4.70) defined in Subsection 4.3.3, which allow to compute the cost, duration and number of sites of the sequence resulting from a concatenation of sequences σ and σ' (without allocated sites).

$$M_{\sigma \oplus \sigma'} = M_{\sigma} + M_{\sigma'} \quad (5.15)$$

$$D_{\sigma \oplus \sigma'} = D_{\sigma} + w_{j,i'} + D_{\sigma'} \quad (5.16)$$

$$C_{\sigma \oplus \sigma'} = C_{\sigma} + M_{\sigma'} \cdot (D_{\sigma} + w_{j,i'}) + C_{\sigma'} \quad (5.17)$$

Nevertheless, two new modifications are needed to apply these equations to CCVRP-ID. The first are based on the definition of super-nodes. A super-node here is composed of a visited site and a set of allocated sites, so for a sequence $\sigma = i$ of one super-node based at visited site i with a set A_i of $|A_i|$ allocated sites, its cost is computed as $C_{\sigma} = \sum_{j \in A_i} \delta \cdot w_{i,j}$, its duration is $D_{\sigma} = 0$ and its number of sites is $M_{\sigma} = 1 + |A_i|$.

For instance, consider the sequences $\sigma = (1, 2)$ and $\sigma' = (3)$ where $A_1 = \emptyset$, $A_2 = \emptyset$, $A_3 = (4, 5)$, $M_{\sigma} = 2$, $M_{\sigma'} = 3$, $D_{\sigma} = w_{12}$, $D_{\sigma'} = 0$, $C_{\sigma} = w_{12}$, $C_{\sigma'} = \delta \cdot (w_{34} + w_{35})$. The number of serviced sites, the cost and the duration of the sequence $\sigma \oplus \sigma'$ are $M_{\sigma \oplus \sigma'} = M_{\sigma} + M_{\sigma'} = 5$, $D_{\sigma \oplus \sigma'} = D_{\sigma} + w_{23} + D_{\sigma'} = w_{12} + w_{23}$ and $C_{\sigma \oplus \sigma'} = C_{\sigma} + M_{\sigma'} \cdot (D_{\sigma} + w_{23}) + C_{\sigma'} =$

$w_{12} + 3 \cdot (w_{12} + w_{23}) + \delta \cdot (w_{34} + w_{35})$, respectively.

The reversal of a sequence $\sigma = (1, 2, \dots, i-1, i)$ for the CCVRP-ID can also be computed recursively from the sequence $\sigma' = (1, 2, \dots, i-1)$ as $\overleftarrow{\sigma} = i \oplus \overleftarrow{\sigma}'$.

The second modification is the definition of the allocation operator \odot which means the allocation of a site. Note that a site can be allocated to another site (or super-node), but not to sequences with more sites. For instance $\sigma \odot j$ with $\sigma = i$ implies the allocation of site j to the super-node i . Equations (5.18), (5.19) and (5.20) allow to compute the number of sites, the duration and the cost of the super-node i .

$$M_{\sigma \odot j} = M_{\sigma} + 1 \quad (5.18)$$

$$D_{\sigma \odot j} = D_{\sigma} = 0 \quad (5.19)$$

$$C_{\sigma \odot j} = C_{\sigma} + \delta \cdot w_{ij} \quad (5.20)$$

As an example, the first route in the Figure 5.1 can be defined as $\sigma_1 \oplus \sigma_2 \oplus \sigma_3$, where $\sigma_1 = 0$, $\sigma_2 = (1) \odot \{7, 8\}$ and $\sigma_3 = (2, 3, 0)$.

The allocation operator \odot can be used to evaluate in constant time new moves like the relocation of allocated sites and the change of transportation mode, which are explained in Subsection 5.3.5.

5.3.4 Heuristic for the Initial Solution

Each ELS in the proposed MS-ELS begins with an initial solution computed by a greedy randomized heuristic that builds R routes simultaneously. Perturbed distances are used after the first start by generating random variables normally distributed ($\tilde{w}_{ij} \sim N(w_{ij}, 3)$). The distribution $N(w_{ij}, 3)$ generates values centered on w_{ij} and with standard deviation equal to 3. Preliminary tests have been performed with different standard deviation values.

The first route is initialized with the nearest site to depot, while the remaining routes are initialized with one randomly selected site.

Then, feasible insertions of the remaining sites in the incumbent routes are performed until all sites are inserted. No allocations are allowed in the initial solution. Every time, the site with greater difference between its best insertion and second best insertion is selected.

When a site cannot be inserted in a route, its insertion cost is penalized by adding a big M value to the best insertion cost, in order to promote its insertion.

When it is not possible to insert a site in any route, a randomly selected site already visited is removed from the R trips until a feasible insertion appears. This process, repeated until all sites are routed, never fails on the instances tested.

5.3.5 Variable Neighborhood Descent

Our MS-ELS for the CCVRP-ID uses a VND as improvement procedure, like in previous metaheuristics for mt-CCVRP. The procedure can be described by Algorithm 4.2 in Subsection 4.3.5. New moves are defined to distinguish between visited and allocated sites.

The proposed VND for the CCVRP-ID is based on $d = 6$ neighborhoods. Each neighborhood is implicitly defined by a type of move. Starting from $b = 1$ and one input solution S' , the basic iteration of VND consists in exploring the neighborhood N_b of S' ($N_b(S')$). The best improvement move is used: the best neighbor found after browsing the complete neighborhood becomes the incumbent solution in case of improvement and b is reset to 1, otherwise b is incremented. The procedure stops when the exploration of N_d brings no improvement.

Neighborhood N_1 involves 2-opt moves on one trip, while N_2 performs λ -interchanges on one trip. These two neighborhoods have been already used by MS-ILS, MA|PM and MS-ELS metaheuristics for the mt-CCVRP, described in Subsection 4.5.6. Neighborhoods N_3 and N_4 consider the same moves, but applied to two trips. They are similar to the ones used in previous metaheuristics, but here only one trip per vehicle is available. N_5 considers the relocation of allocated sites when they exist. Finally, neighborhood N_6 handles transportation mode changes.

N_1 : 2-OPT moves on one trip

Recall that this move consists in deleting two arcs from a trip and reconnecting the resulting fragments using two new arcs. As depicted in Figure 4.2, the arc coefficients between new arcs change and precomputations are needed to compute the cost variations in constant time. The variant shown in Figure 4.3 is also considered for this VND implementation. This neighborhood can be browsed in $O(n^2/R)$. A more detailed description is given in Subsection 4.3.5.

N₂ : λ -interchanges on one trip

This type of move consists in exchanging a sequence σ_1 from one to λ consecutive nodes with another (non-overlapping) sequence σ_2 containing zero to λ consecutive nodes. Each sequence with more than one node can be reversed in the reinsertion, giving four cases. Note that for the second sequence a length of zero is allowed, to include relocations of the first string. An example of this move is depicted in Figure 4.4. This figure also shows that arc coefficients in sequences σ_1 and σ_2 , as well as in the sequence between them, change. This neighborhood can be explored in $O((n^2\lambda^2)/R)$ by using the previously described precomputations. A more detailed description is presented in Subsection 4.3.5.

N₃ : 2-OPT moves on two trips

The moves performed in this neighborhood take two trips k and k' , delete one arc in each one and reconnect them with different arcs. Note that this neighborhood is equivalent to the exchange of two sequences of sites, beginning or ending at the depot. These moves are similar to the ones used in the previous chapter, except that now we cannot have more than one trip per vehicle. The same cost variations formulas can be used, since this neighborhood is a special case where the number of trips is equal to one. This neighborhood can be browsed in $O(n^2)$.

N₄ : λ -interchanges moves on two trips

Again, this neighborhood already browsed for the mt-CCVRP is here applied to the case of one single trip per vehicle. Note that arc coefficients can vary depending on the positions of the interchanged sequences and their number of nodes. N_4 can be searched in $O(n^2\lambda^2)$.

As explained in Subsection 5.3.3, a concatenation is always valid for the CCVRP-ID. The formulas can be adapted for the implementation of N_1 , N_2 , N_3 and N_4 for the CCVRP-ID.

N₅ : Node reallocation

This neighborhood consists in changing the visited sites to which are assigned the allocated sites. That means the removal of each allocated site and the evaluation of all its possible allocations, in the same and distinct trips. This move looks similar to the root refining moves defined in Chao (2002) and Villegas *et al.* (2011) for the truck and trailer routing problem, but becomes simpler because in CCVRP-ID only individual nodes can be allocated instead of second-level trips in the TTRP. An example of this move is depicted in Figure 5.3, where part (a.) shows a trip with node u allocated to node j , then this node is

reassigned to node i in the same trip in part (b.). If V_V is the set of visited sites and V_A the set of allocated sites ($V_V \cup V_A = V'$), N_5 can be explored in $O(|V_V| \cdot |V_A|)$. The number of neighbors (moves) can vary between $|V_V| \cdot |V_A| = 0$ (if $|V_A| = 0$) and $|V_V| \cdot |V_A| = n^2/4$ (if $|V_V| = |V_A|$).

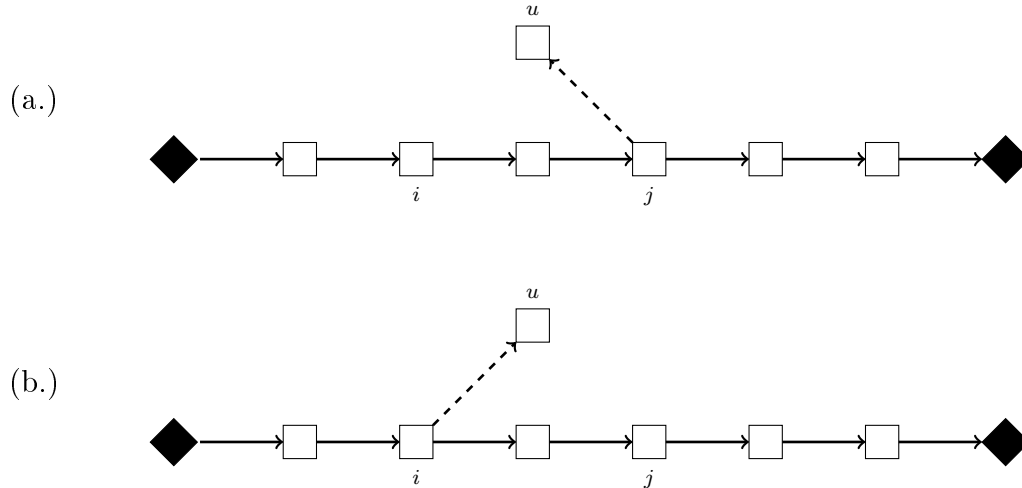


Figure 5.3: Example of node reallocation move for the CCVRP-ID.

N_6 : Transportation mode change

This last neighborhood considers changing the transportation mode to reach a site. That means a visited node can be removed and allocated to another node, or an allocated node can be removed and inserted in a trip to be visited by a regular vehicle.

As shown in Figure 5.4, three cases are considered. Part (a.) of the figure shows the original trip, where nodes i and j are visited while nodes u and v are allocated to node i . In part (b.), node u is no longer delivered indirectly: it is inserted to be visited between node i and j . In part (c.) the originally visited node j is removed and allocated to node i . Finally, part (d.) shows the case where the role of nodes i and u are interchanged. Here when an allocated node is removed, it can be inserted in the same or distinct trip. Similarly, when a visited site is removed, it can be allocated to another node in the same or distinct trip. Moves in this neighborhood can be computed in constant time by using the concatenation operator \oplus and the allocation operator \odot , and the neighborhood can be browsed in $O(n^2)$.

Salari *et al.* (2010b) define a neighborhood called Extraction-Assignment for the Ring-Star problem, which can be seen as the union of N_5 and N_6 . The moves consist in

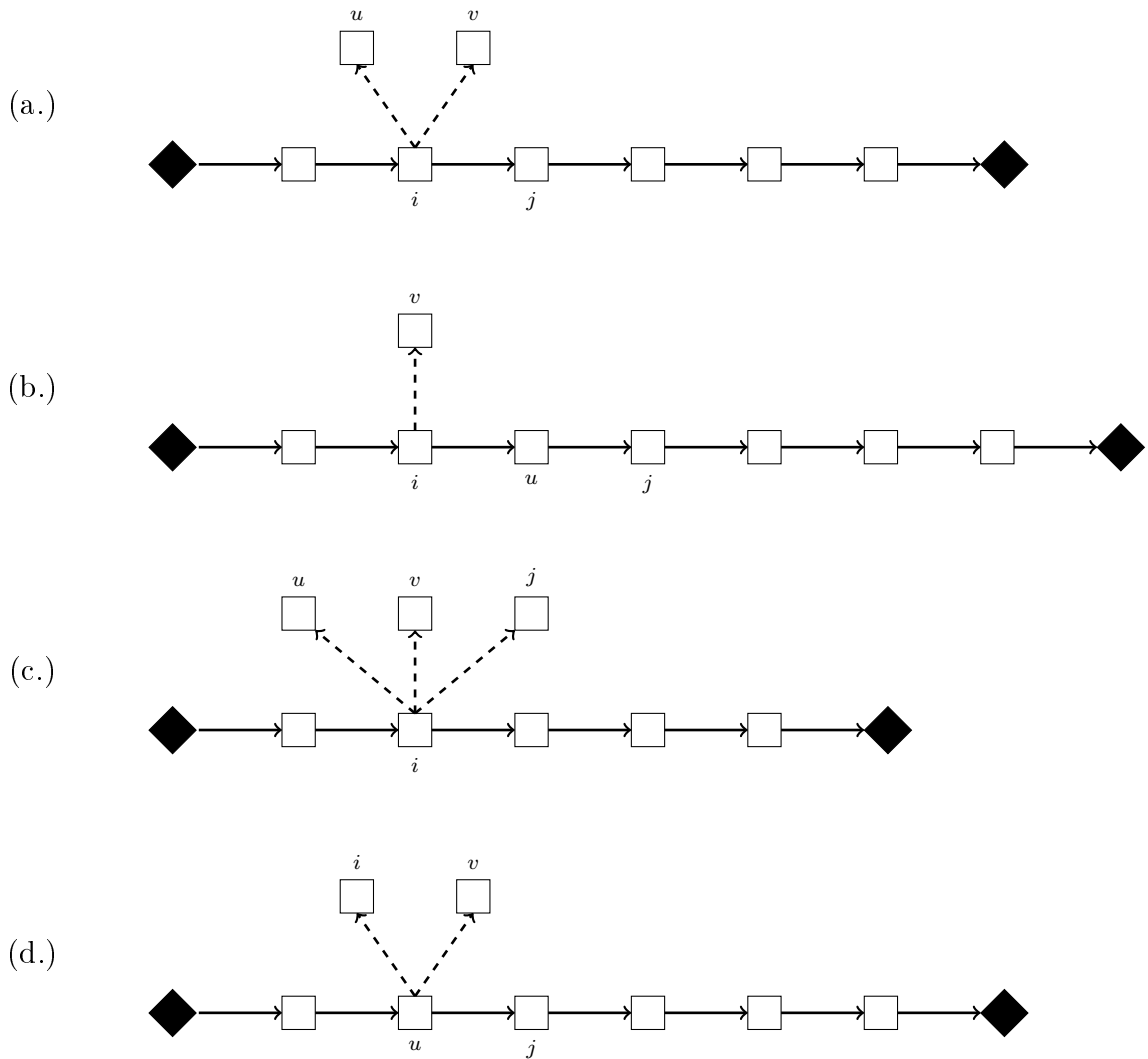


Figure 5.4: Example of transportation mode change moves for the CCVRP-ID.

removing a node to insert it or reallocate it in the best possible position in the current solution.

5.3.6 Post-Optimization Procedure

This procedure is used when there exist allocated sites in a solution. In this case, reversal of sequences can be limited because of the allocated sites. For instance, in Figure 5.1, the reversal of sequence (1, 2, 3) may produce better sum of arrival times when only sites 1, 2 and 3 are considered, but as sites 7 and 8 are assigned to site 1, this reversal does not produce a better result.

The VND improvement procedure is followed by a *Post_Optimization* procedure which

behaves like a kind of large neighborhood search, see Algorithm 5.2. Basically, on each iteration all allocated nodes are removed using the *Removing* procedure, the VND is applied to the remaining (visited) nodes, the removed nodes are reinserted like in the heuristic for initial solution (see Subsection 5.3.4) and the VND is applied again.

This procedure is repeated until no improvements are found. As before, *Update_Precomputations* is used to speed up local moves performed by VND.

Algorithm 5.2 – Post-Optimization for a solution S

```
1: repeat  
2:    $S' \leftarrow S$   
3:   Removing ( $S$ )  
4:   Update_Precomputations ( $S$ )  
5:   VND ( $S$ )  
6:   Inserting ( $S$ )  
7:   Update_Precomputations ( $S$ )  
8:   VND ( $S$ )  
9: until  $Z_{S'} \leq Z_S$   
10:  $S \leftarrow S'$ 
```

5.3.7 Perturbation

The perturbation procedure described here is based on the destroy and reconstruct procedures used by ALNS metaheuristics.

The proposed perturbation procedure consists in removing some sites from the incumbent solution, and insert them by using a similar procedure to the one used by the Initial Solution Heuristic (Subsection 5.3.4). As this procedure can reinsert the sites in their previous positions, their insertion in the same route is penalized. A site can be inserted in the same route only when insertions in another route are impossible.

The number of removed sites has been tested with different fixed values, randomly generated values and dynamic values. After the tests of these different options, this number has been set to three.

5.4 Computational Experiments

Computational experiments have been performed to evaluate and compare the performance of the mathematical model and the metaheuristic approach. The instances used are presented in Subsection 5.4.1. Subsection 5.4.2 refers to parameter tuning. Subsection 5.4.3 describes the results on small instances while larger instances are addressed in Subsection 5.4.4.

5.4.1 Implementation and Instances

The MS-ELS described in this chapter has been implemented in Visual C++ 2010 and the mathematical model in CPLEX 12.5. All of them have been tested on a 2.50 GHz Intel Core i5 computer with 4 GB of RAM and Windows 7 Professional. Two types of experiments are reported. The first one compares metaheuristic solutions with 0–1 MILP resolution via a commercial solver (CPLEX), and compares CCVRP-ID and CCVRP solutions in terms of average arrival time. The second type of experiments presents solutions on larger instances and compares CCVRP-ID with published solutions for the CCVRP.

The comparison with CPLEX involves 24 randomly generated instances: 12 with 15 demand nodes and 12 with 30 demand nodes. For larger instances, the same instances defined for the CCVRP have been used, which are derived from classical CVRP instances: the 14 CMT instances (Christofides *et al.*, 1979), with $n = 50$ to 199 demand nodes, and the 20 GWKC larger instances (Golden *et al.*, 1998), with $n = 200$ to 483. The fleet size is fixed as in the CCVRP (otherwise the optimum is trivial, with one direct route to each demand node) and the maximum trip length defined for some instances is ignored. As the seven CMT instances with a trip length constraint share the same network as the seven unconstrained instances, $7 + 20 = 27$ CCVRP instances are finally obtained. In all instances, the traveling time w_{ij} on arc (i, j) is equal to the Euclidean distance, computed as a double-precision real number. The tests are performed with five δ values: 10 000, 10, 5, 2.5 and 1.

5.4.2 Parameter Tuning

The MS-ELS described in this chapter has only four parameters: *MaxStart* defines the number of starts or number of calls to the ELS, *MaxIter* describes the number of iterations to be performed by each ELS, *MaxChildren* indicates the number of children to be generated on each ELS iteration, and λ limits the maximum number of consecutive sites to be interchanged in λ -interchange moves used in neighborhoods N_2 and N_4 .

Like in the previous chapter, a “computing budget” is fixed but, instead of 3 000 VND calls, the budget concern here the total number of children. Note that for the cases where no allocated sites are used both criteria are identical. But when solutions contain allocated sites, an unpredictable number of VND calls is executed in the *Post_Optimization* procedure. In those cases the execution time can be increased.

Different parameter values have been tested on a subset of 4 selected instances. Three values are used for *MaxStart*, *MaxChildren* and λ ($MaxStart \in \{1, 3, 5\}$, $MaxChildren \in \{1, 5, 10\}$, and $\lambda \in \{2, 3, 4\}$), while the value of *MaxIter* is computed as $MaxIter = Budget/MaxStart/MaxChildren$. The parameter λ is set to 3 to have a compromise between quality and execution time of the browsed neighborhoods, as in our previous implementations.

Table 5.1 summarizes the different tests performed for parameters *MaxStart*, *MaxIter* and *MaxChildren*. The first three columns indicate the parameter values while the fourth one shows the average percentage deviation with respect to the best solution found for the CCVRP-ID with $\delta = 5$ (*AvgDev*), for a subset of instances. The best results in terms of average deviation are obtained with $MaxStart = 3$, $MaxIter = 100$ and $MaxChildren = 10$, in boldface on Table 5.1.

Note that when $MaxStart = 1$, $MaxIter = 3\,000$ and $MaxChildren = 1$ the algorithm is equivalent to an ILS metaheuristic, and when $MaxStart \neq 1$ and $MaxChildren = 1$, the algorithm becomes an MS-ILS.

Table 5.1: Summary of the performed tests.

<i>MaxStart</i>	<i>MaxIter</i>	<i>MaxChildren</i>	<i>AvgDev</i>
1	3 000	1	0.22
1	600	5	0.21
1	300	10	0.19
3	1 000	1	0.12
3	200	5	0.10
3	100	10	0.10
5	600	1	0.14
5	120	5	0.12
5	60	10	0.11

5.4.3 Results on Small Instances

Tables 5.2 and 5.3 present the results for the 0–1 mixed integer linear program on small instances with $n = 15$ and $n = 30$ required sites, respectively. Each table is composed of five result types which correspond to the five values of the parameter δ . Similar results are shown in Tables 5.4 and 5.5 for the developed metaheuristic algorithm. The considered values of δ are 10000, 10, 5, 2.5 and 1. The value 10000 is used to avoid allocated sites in the solution, which means that the problem is equivalent to a CCVRP. Although there is no proof saying that this value guarantees to avoid allocations, that always happen for the tested instances. $\delta = 1$ is another extreme value which lead to the same edge costs in the two transportation modes.

In Tables 5.2 and 5.3, the first four columns display the instance name, the number of required sites n , the number of vehicles R , and the average loading level of vehicles $\left(\sum_{i=1}^n q_i\right)/(Q \cdot R)$, which varies between 0 and 1 (1 meaning that all vehicles are full). For each value of δ , the cost of the linear relaxation of the mathematical model (LR), the optimal solution value (Cost), the percentage gap between the best integer solution found and the best lower bound found (gap), which is given by CPLEX, and the running time in seconds (Time), limited to one hour, are presented. For all $\delta \neq 10000$ the percentage reduction of the average arrival time with respect to CCVRP, gap_δ , is also shown. Remind that when $\delta = 10000$, our CCVRP-ID model is equivalent to the CCVRP case. Note that the percentage reduction in the sum of arrival times is equivalent to the percentage reduction in the average arrival time. In Table 5.3, when optimal solutions are not found ($\text{gap} \neq 0$), the best integer solution and the best lower bound found by CPLEX are listed in the Cost column.

The twelve instances in each table can be classified by their different features: the first six instances present an average demand of 13.33 units, while the next six have an average demand of 20.00 units. The node locations on the instances in the same subset are the same. In both subsets, the number of vehicles and the average loading level of vehicles vary to evaluate the impact of these features on the execution time.

In general, for $n = 15$ sites (Table 5.2) the mathematical model always finds an optimal solution for the 12 instances and the five values of δ . The average execution time is larger for the last six instances (instances 7 to 12). A small number of vehicles also implies a larger running time, because of the subsequent larger number of sites per vehicle. In most of the

Table 5.2: Results of the mathematical model on small CCVRP-ID instances with $n = 15$ sites.

File	n	R	$\sum \frac{q_i}{Q_i R}$	$\delta = 10000$				$\delta = 10$				$\delta = 5$				$\delta = 2.5$				$\delta = 1$									
				LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time						
test ₀₁	15	2	1.00	804.36	1072.94	0.00	8.47	804.36	1072.94	0.00	8.44	0.00	8.44	0.00	804.36	1039.32	0.00	9.30	3.13	778.37	831.95	0.00	1.28	22.46	535.24	566.47	0.00	0.92	47.20
test ₀₂	15	2	0.83	804.36	1072.94	0.00	47.12	804.36	1072.94	0.00	49.39	0.00	5.72	5.32	804.36	1015.88	0.00	5.72	5.32	778.37	831.95	0.00	0.96	22.46	535.24	557.66	0.00	0.53	48.03
test ₀₃	15	3	0.95	695.40	785.60	0.00	2.12	695.40	785.60	0.00	2.50	0.00	2.17	0.00	695.40	785.60	0.00	2.17	0.00	686.23	722.08	0.00	0.87	8.09	526.46	539.99	0.00	1.01	31.26
test ₀₄	15	3	0.83	695.40	785.60	0.00	2.01	695.40	785.60	0.00	2.10	0.00	1.78	0.00	695.40	785.60	0.00	1.78	0.00	686.23	718.83	0.00	0.76	8.50	526.32	539.04	0.00	0.96	31.38
test ₀₅	15	4	1.00	618.87	687.29	0.00	2.05	618.87	687.29	0.00	1.63	0.00	2.11	0.00	618.87	687.29	0.00	2.11	0.00	617.38	666.59	0.00	1.40	3.01	522.52	538.73	0.00	0.95	21.62
test ₀₆	15	4	0.83	618.87	646.67	0.00	0.69	618.73	646.67	0.00	0.75	0.00	0.78	0.00	618.73	646.67	0.00	0.78	0.00	617.02	642.04	0.00	0.75	0.72	520.65	527.38	0.00	0.42	18.45
test ₀₇	15	2	1.00	1570.83	2348.37	0.00	1000.63	1570.83	2347.28	0.00	194.66	0.05	62.30	7.09	1570.83	2181.92	0.00	62.30	7.09	1525.13	1850.93	0.00	5.48	21.18	1078.83	1186.70	0.00	4.64	49.47
test ₀₈	15	2	0.83	1569.66	2335.40	0.00	260.48	1569.66	2335.40	0.00	298.78	0.00	147.15	6.57	1569.66	2181.92	0.00	147.15	6.57	1522.52	1786.85	0.00	2.50	23.49	1077.25	1148.82	0.00	1.64	50.81
test ₀₉	15	3	1.00	1394.95	1717.46	0.00	14.98	1394.95	1717.46	0.00	7.87	0.00	16.57	0.00	1394.95	1717.46	0.00	16.57	0.00	1385.23	1527.96	0.00	1.84	11.03	1072.69	1121.27	0.00	0.75	34.71
test ₁₀	15	3	0.83	1393.20	1628.31	0.00	3.81	1393.20	1628.31	0.00	2.76	0.00	3.64	0.00	1393.20	1628.31	0.00	3.64	0.00	1381.87	1527.96	0.00	2.00	6.16	1067.76	1103.76	0.00	1.39	32.21
test ₁₁	15	4	0.94	1277.88	1317.97	0.00	0.59	1277.88	1317.97	0.00	0.42	0.00	0.53	0.00	1277.88	1317.97	0.00	0.53	0.00	1273.95	1317.97	0.00	0.79	0.00	1063.75	1091.70	0.00	0.96	17.17
test ₁₂	15	4	0.83	1271.84	1317.97	0.00	0.39	1271.84	1317.97	0.00	0.53	0.00	0.53	0.00	1271.84	1317.97	0.00	0.53	0.00	1268.56	1317.97	0.00	0.85	0.00	1060.14	1087.60	0.00	0.84	17.48
Mean				0.00	112.53	0.00	47.49	0.00	21.05	1.84	0.00	1.62	10.59	0.00	1.24	33.32	0.00	1.24	33.32	0.00	1.24	33.32	0.00	1.24	33.32	0.00	1.24	33.32	

Table 5.3: Results of the mathematical model on small CCVRP-ID instances with $n = 30$ sites.

File	n	R	$\sum \frac{q_i}{Q.R}$	$\delta = 10000$			$\delta = 10$			$\delta = 5$			$\delta = 2.5$			$\delta = 1$												
				LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time	LR	Cost	gap	Time					
test ₁₃	30	2	1.00	1597.79	3065.80	40.37	3600.00	1597.79	2998.71	40.80	3600.00	2.19	1597.79	2478.34	27.55	3600.00	19.16	1597.79	1553.84	1866.58	0.00	1895.92	39.12	1117.02	1196.54	0.00	26.09	60.97
test ₁₄	30	3	0.83	1492.68	2255.97	25.30	3600.00	1492.68	2250.33	25.44	3600.00	0.25	1492.68	2060.55	18.29	3600.00	8.66	1492.68	1468.63	1676.72	0.00	3327.78	25.68	1108.43	1144.50	0.00	9.20	49.40
test ₁₅	30	4	1.00	1408.23	1883.58	17.45	3600.00	1408.23	1903.03	17.77	3600.00	-1.03	1408.23	1802.67	14.14	3600.00	4.30	1394.47	1394.47	1574.47	3.16	3600.00	16.41	1103.71	1144.11	0.00	29.89	39.26
test ₁₆	30	4	0.83	1407.51	1773.96	12.70	3600.00	1407.51	1773.96	11.53	3600.00	0.00	1407.51	1737.57	10.53	3600.00	2.05	1393.77	1511.44	0.00	188.84	14.80	1102.92	1126.36	0.00	9.12	36.51	
test ₁₇	30	5	0.80	1336.12	1540.46	4.91	3600.00	1336.12	1466.77	4.78	3600.00	0.00	1336.12	1522.28	3.08	3600.00	1.18	1327.82	1423.16	0.00	377.20	7.61	1097.98	1113.95	0.00	7.09	21.62	
test ₁₈	30	6	0.95	1281.80	1435.86	3.43	3600.00	1281.80	1435.86	3.15	3600.00	0.00	1281.80	1431.67	2.68	3600.00	0.29	1279.87	1375.52	0.00	561.61	4.20	1095.51	1105.11	0.00	2.67	23.03	
test ₁₉	30	2	1.00	3357.29	6469.38	40.05	3600.00	3357.29	5911.97	36.85	3600.00	8.62	3357.29	5297.20	26.00	3600.00	18.12	3269.91	4094.29	0.00	3438.20	36.71	2462.39	2572.97	0.00	39.68	60.23	
test ₂₀	30	3	0.83	3146.17	4742.34	25.77	3600.00	3146.17	4498.23	20.55	3600.00	5.15	3146.17	4210.26	17.58	3600.00	11.22	3092.93	3615.81	0.00	1621.05	23.75	2449.15	2540.22	0.00	23.59	48.54	
test ₂₁	30	4	1.00	2972.33	3906.87	14.71	3600.00	2972.33	3906.87	15.79	3600.00	0.00	2972.33	3826.82	14.09	3600.00	2.05	2951.02	3262.19	0.00	1838.84	16.50	2445.92	2504.94	0.00	9.50	35.88	
test ₂₂	30	4	0.83	2961.48	3837.97	14.75	3600.00	2961.48	3799.75	13.66	3600.00	1.00	2961.48	3625.06	9.10	3600.00	5.55	2943.46	3228.92	0.00	331.87	15.87	2439.66	2491.39	0.00	8.48	35.09	
test ₂₃	30	5	0.80	2825.53	3238.55	4.44	3600.00	2825.53	3238.55	4.59	3600.00	0.00	2825.53	3175.42	1.56	3600.00	1.95	2815.06	2962.57	0.00	52.86	8.52	2429.26	2457.89	0.00	4.07	24.11	
test ₂₄	30	6	1.00	2750.53	3173.65	5.64	3600.00	2750.53	3173.65	5.42	3600.00	0.00	2750.53	3176.11	5.05	3600.00	-0.08	2744.38	3056.72	2.45	3600.00	3.68	2434.28	2532.84	0.00	59.10	20.19	
Mean						17.46	3600.00			16.69	3600.00	1.35			12.47	3600.00	6.20			0.47	1736.18	17.82		0.00	19.04	37.90		

cases, for the instances with the same number of vehicles, the execution time increases when the average loading level gets close to one.

When $\delta = 10$ only the instance 7 has a different optimal solution value compared to the corresponding optimal solution values for the CCVRP ($\delta = 10\,000$). Nevertheless, the execution time to solve the problem is reduced by 84.72% on average. The number of optimal solution values which improve the corresponding CCVRP optimal solutions increases to 4 when $\delta = 5$. Only two objective values do not vary when $\delta = 2.5$ and all of them change for $\delta = 1$ with respect to the CCVRP optimal solutions. The previous one illustrates that when auxiliary vehicles are less than five times slower than regular vehicles, the indirect deliveries start to become more and more interesting, but that can vary for each instance.

Broadly, the execution time decreases with the value of δ . For instance, the average execution time reduces by 57.80% and 98.90% for $\delta = 10$ and $\delta = 1$, respectively. The execution time is also increased by larger numbers of vehicles and average loading levels of vehicles.

The average arrival time decreases by 1.81%, 10.59% and 33.32% when δ is reduced to 5, 2.5 and 1, respectively. When the number of vehicles is lower, which implies a larger number of nodes per vehicle, the average arrival time is more sensible to the δ value. For instance, for $R = 2$ the average arrival time reduces by 22.40% from $\delta = 10\,000$ to $\delta = 2.5$ and only by 0.93% for $R = 4$ with the same δ values.

When $n = 30$ sites (Table 5.3) similar results are found. However, the mathematical model cannot solve the problem when $\delta \geq 5$. The gap between the best integer solution and the best lower bound in CPLEX reduces when δ decreases. For $\delta = 2.5$, CPLEX finds 10 optimal solutions, while it finds all optimal values when $\delta = 1$, within a lower execution time.

Nevertheless, when comparing results for $\delta = 10\,000$ and $\delta = 10$, the gap between the best integer solution and the best lower bound sometimes increases and, in one case, the best integer solution has a worse objective function value for $\delta = 10$ (instance 15). The objective function value is improved five times when $\delta = 10$. Like in Table 5.2, instances look harder when the number of vehicles is smaller and when the average loading level is greater. When $\delta = 5$ one of the best integer solutions is worse than the best integer solutions found for $\delta = 10\,000$ and 10 (instance 24). Remind that these solutions are not optimal.

Tables 5.4 and 5.5 present the results of the MS-ELS for the small CCVRP-ID instances. The same first four columns as Tables 5.2 and 5.3 are kept. Then, for each value of the parameter δ , the best objective function value over five runs (Cost), the percentage deviation of the best solution cost $Z(heur)$ to the optimal solution $Z(MILP)$ of CPLEX (or to the best lower bound when the solver must be stopped after one hour) computed as $gap = (Z(heur)/Z(MILP) - 1) \times 100$, and the average running time per run in seconds (Time) are indicated. When $\delta \neq 10\,000$ the percentage reduction of the average arrival time with respect to CCVRP ($\delta = 10\,000$), gap_δ , is also shown.

For instances with $n = 15$ sites, all optimal solutions are found. Nevertheless, the execution time is less sensitive to the parameter δ , the number of vehicles or the average loading level of vehicles, compared to CPLEX.

For some instances, especially when δ is large, the MS-ELS is faster than the mathematical model, while for some others, especially when $\delta = 1$, this behavior is reversed. Based on 5 runs, MS-ELS has always found the same solution for each instance and each δ value.

For instances with $n = 30$ sites, MS-ELS retrieves the optimal solutions found by the mathematical model and, when $gap \neq 0$, it improves some of the best solutions found. As the mathematical model becomes faster when δ decreases, the number of improved solutions and the average improvement also diminish when δ decreases. When the mathematical model finds an optimal solution, MS-ELS finds the same solution, with $gap = 0$ in boldface.

When $\delta = 10\,000$, six best known solutions are improved, with an average improvement of 0.94% and the average execution time is 10.81 seconds. For $\delta = 10$, six best known solutions are again improved, but the average improvement is reduced to 0.39%. The execution times present small variations. These variations are notably larger for instances with a smaller number of vehicles R . The objective values are reduced by 0.81% on average. Only three best known solutions are improved when $\delta = 5$, with an average improvement equal to 0.13%. The execution time increases for most of the instances with $\delta = 5$. The sum of arrival times are reduced by 5.47% on average with respect to the CCVRP case ($\delta = 10\,000$). When $\delta = 2.5$ a best known solution is improved and the objective functions are 17.06% lower than CCVRP. Ten of the solutions found have an optimal status. The execution time is reduced with respect to the case $\delta = 5$. Finally, when $\delta = 1$, all optimal solutions are found, which improves by 37.74% on average the arrival time compared to the CCVRP model. The

Table 5.4: Results of MS-ELS on small CCVRP-ID instances with $n = 15$ sites.

File	n	R	$\sum_{i=1}^n q_i$	$\delta = 10000$			$\delta = 5$			$\delta = 2.5$			$\delta = 1$							
				Cost	gap	Time	Cost	gap	Time	Cost	gap	Time	Cost	gap	Time					
test ₀₁	15	2	1.00	1072.94	0.00	7.70	1072.94	0.00	7.98	1039.32	0.00	7.22	831.95	0.00	6.36	22.46	566.47	0.00	6.36	47.20
test ₀₂	15	2	0.83	1072.94	0.00	6.31	1072.94	0.00	6.44	1015.88	0.00	8.06	831.95	0.00	7.02	22.46	557.66	0.00	7.12	48.03
test ₀₃	15	3	0.95	785.60	0.00	5.25	785.60	0.00	5.25	785.60	0.00	6.72	722.08	0.00	8.01	8.09	539.99	0.00	5.77	31.26
test ₀₄	15	3	0.83	785.60	0.00	5.50	785.60	0.00	5.52	785.60	0.00	6.52	718.83	0.00	9.92	8.50	539.04	0.00	6.84	31.38
test ₀₅	15	4	1.00	687.29	0.00	5.38	687.29	0.00	5.41	687.29	0.00	5.52	666.59	0.00	7.34	3.01	538.73	0.00	7.38	21.62
test ₀₆	15	4	0.83	646.67	0.00	5.51	646.67	0.00	5.78	646.67	0.00	6.34	642.04	0.00	7.17	0.72	527.38	0.00	7.88	18.45
test ₀₇	15	2	1.00	2348.37	0.00	8.99	2347.28	0.00	10.06	2181.92	0.00	8.82	1850.93	0.00	9.29	21.18	1186.70	0.00	8.82	49.47
test ₀₈	15	2	0.83	2335.40	0.00	12.52	2335.40	0.00	10.11	2181.92	0.00	11.07	1786.85	0.00	12.86	23.49	1148.82	0.00	8.43	50.81
test ₀₉	15	3	1.00	1717.46	0.00	6.11	1717.46	0.00	6.09	1717.46	0.00	6.55	1527.96	0.00	8.49	11.03	1121.27	0.00	8.16	34.71
test ₁₀	15	3	0.83	1628.31	0.00	6.58	1628.31	0.00	6.60	1628.31	0.00	6.90	1527.96	0.00	9.53	6.16	1103.76	0.00	8.90	32.21
test ₁₁	15	4	0.94	1317.97	0.00	6.15	1317.97	0.00	5.53	1317.97	0.00	5.56	1317.97	0.00	6.42	0.00	1091.70	0.00	6.51	17.17
test ₁₂	15	4	0.83	1317.97	0.00	5.27	1317.97	0.00	5.29	1317.97	0.00	5.44	1317.97	0.00	5.81	0.00	1087.60	0.00	6.26	17.48
Mean			0.00	6.77	0.00	6.67	0.00	0.00	7.06	1.84	0.00	8.19	10.59	0.00	7.37	33.32				

Table 5.5: Results of MS-ELS on small CCVRP-ID instances with $n = 30$ sites.

File	n	R	$\sum \frac{q_i}{Q}$	$\delta = 10000$			$\delta = 10$			$\delta = 5$			$\delta = 2.5$			$\delta = 1$							
				Cost	gap	Time	Cost	gap	Time	Cost	gap	Time	Cost	gap	Time	Cost	gap	Time	Cost	gap	Time		
test ₁₃	30	2	1.00	3065.80	0.00	18.79	2990.06	-0.29	24.65	2.47	2478.34	0.00	20.30	19.16	1866.58	0.00	13.62	39.12	1196.54	0.00	7.93	60.97	
test ₁₄	30	3	0.83	2231.22	-1.10	14.09	2211.23	-1.74	22.34	0.90	2060.55	0.00	24.71	7.65	1676.72	0.00	14.46	24.85	1141.50	0.00	13.39	48.84	
test ₁₅	30	4	1.00	1875.24	-0.44	7.97	1875.24	-1.46	8.20	0.00	1802.67	0.00	10.98	3.87	1574.47	0.00	10.72	16.04	1144.11	0.00	7.78	38.99	
test ₁₆	30	4	0.83	1773.96	0.00	11.51	1773.96	0.00	11.64	0.00	1737.57	0.00	15.87	2.05	1511.44	0.00	9.71	14.80	1126.36	0.00	11.46	36.51	
test ₁₇	30	5	0.80	1540.46	0.00	9.35	1540.46	0.00	9.39	0.00	1522.28	0.00	12.82	1.18	1423.16	0.00	12.83	7.61	1113.95	0.00	8.91	27.69	
test ₁₈	30	6	0.95	1432.44	-0.24	7.11	1432.44	-0.24	7.12	0.00	1431.67	0.00	7.49	0.05	1375.52	0.00	9.01	3.97	1105.11	0.00	10.05	22.85	
test ₁₉	30	2	1.00	6231.40	-3.68	13.82	5911.97	0.00	18.95	5.13	5294.00	-0.06	14.63	15.04	4094.29	0.00	9.63	34.30	2572.97	0.00	7.25	58.71	
test ₂₀	30	3	0.83	4516.95	-4.75	13.71	4460.45	-0.84	18.61	1.25	4210.26	0.00	24.33	6.79	3615.81	0.00	9.73	19.95	2540.22	0.00	7.23	43.76	
test ₂₁	30	4	1.00	3906.87	0.00	8.06	3906.87	0.00	8.21	0.00	3789.47	-0.98	11.29	3.00	3262.19	0.00	10.55	16.50	2504.94	0.00	7.20	35.88	
test ₂₂	30	4	0.83	3795.63	-1.10	10.53	3795.63	-0.11	10.92	0.00	3625.06	0.00	13.88	4.49	3228.92	0.00	10.32	14.93	2491.39	0.00	7.43	34.36	
test ₂₃	30	5	0.80	3238.55	0.00	8.37	3238.55	0.00	8.37	0.00	3175.42	0.00	10.78	1.95	2962.57	0.00	10.94	8.52	2457.89	0.00	7.36	24.11	
test ₂₄	30	6	1.00	3173.65	0.00	6.36	3173.65	0.00	6.36	0.00	3159.56	-0.52	6.88	0.44	3043.54	-0.43	8.44	4.10	2532.84	0.00	7.86	20.19	
Mean				-0.94	10.81	-0.39	12.90	0.81	-0.13	14.50	5.47	-0.04	10.83	17.06	0.00	8.65	37.74						

average execution time is smaller than the other four cases.

5.4.4 Results on Larger Instances

In this subsection the results on CMT (Christofides *et al.*, 1979) and GWKC (Golden *et al.*, 1998) instances are described. Table 5.6 presents the results for CMT instances with the same five values of δ used in Subsection 5.4.3. Similar results are shown in Table 5.7 for GWKC instances.

The first five columns in Tables 5.6 and 5.7 display the instance name, the number of required sites n , the number of vehicles R , the average loading level $\left(\sum_{i=1}^n q_i\right)/(Q \cdot R)$, and the best solution found for the CCVRP from Tables 4.3 and 4.4 from Subsection 4.7.4 in Chapter 4 respectively. Then, for each δ value, the best solution value (Cost), the percentage deviation to the best solution (Dev) and the execution time in seconds (Time) are presented. When $\delta \neq 10\,000$, the percentage reduction in the average arrival time with respect to the best known solutions for the CCVRP is displayed. Remind that when $\delta = 10\,000$, our CCVRP-ID model is equivalent to the CCVRP.

When $\delta = 10\,000$ the proposed algorithm finds all best known solutions for the CMT instances over 5 runs. For GWKC instances it finds 4 of the best known solutions and improves 2 of them. The average deviation ranges between 0% and 0.55%, which indicates that the proposed MS-ELS is robust. The mean deviation is greater for CMT instances, contrary to the behavior in CCVRP and mt-CCVRP in Chapter 4. The average execution times, 84.85 seconds and 1 324.54 seconds, respectively, are lower than the running times of the MS-ILS applied to the CCVRP shown in Subsection 4.7.4.

The algorithm always finds a better objective function value when $\delta = 10$, compared to $\delta = 10\,000$ (CCVRP). The percentage reduction ranges between 0% and 1.31% for CMT instances, and between -0.87% and 27.71% for GWCK instances, where negative values indicate that the best objective function is not better than the best solution values for the CCVRP. Only 8 out of 27 instances have an objective function value greater than the best known solution for the CCVRP. The objective function value is reduced by 0.27% for CMT instances and by 8.94% for GWKC instances on average. The average execution time is increased by 70.78% for CMT instances, while it has decreased by 1.51% for GWKC instances.

Table 5.6: Results of MS-ELS on CMT instances (Christofides *et al.*, 1979).

File	n	R	$\sum \frac{q_i}{Q}$	CCVRP	$\delta = 10,000$			$\delta = 5$			$\delta = 2.5$			$\delta = 1$									
					Cost	Dev	Time	Cost	Dev	Time	gap _s	Cost	Dev	Time	gap _s	Cost	Dev	Time	gap _s				
CMT ₀₁	50	5	0.97	2230.35	2230.35	0.43	12.49	2230.35	0.40	15.15	0.00	2060.62	0.02	26.77	7.61	1731.81	0.00	16.09	22.35	1254.39	0.00	12.19	43.76
CMT ₀₂	75	10	0.98	2391.63	2391.63	0.00	15.11	2391.63	0.00	15.32	0.00	2382.92	0.08	22.54	0.36	2205.51	0.45	30.73	7.78	1845.16	0.00	23.94	22.85
CMT ₀₃	100	8	0.91	4045.42	4045.42	0.48	67.82	4035.77	0.20	135.02	0.24	3713.35	0.48	99.89	8.21	3244.52	0.14	64.26	19.80	2542.90	0.00	29.40	37.14
CMT ₀₄	150	12	0.93	4987.51	4987.51	0.24	131.50	4972.31	0.35	228.81	0.30	4763.83	0.16	244.04	4.48	4416.43	0.12	169.61	11.45	3712.54	0.00	71.75	25.56
CMT ₀₅	199	17	0.94	5809.59	5809.59	0.47	185.53	5809.59	0.27	281.46	0.00	5691.23	0.26	393.94	2.04	5398.11	0.29	281.78	7.08	4825.57	0.01	172.74	16.94
CMT ₁₁	120	7	0.98	7314.55	7314.55	0.45	112.77	7218.57	0.26	211.17	1.31	6974.99	0.15	281.75	4.64	6675.86	0.11	179.54	8.73	6138.21	0.00	47.49	16.08
CMT ₁₂	100	10	0.91	3558.92	3558.92	0.04	68.73	3556.69	0.01	127.44	0.06	3476.42	0.06	108.27	2.32	3289.19	0.04	72.66	7.58	2954.26	0.00	43.84	16.99
Mean						0.30	84.85		0.21	144.91	0.27		0.17	168.17	4.24		0.16	116.38	12.11		0.00	57.34	25.62

Similar results can be observed for $\delta = 5, 2.5$ and 1 . For instance, on CMT instances, the percentage reduction on average arrival times are 4.24% , 12.11% and 25.62% , respectively. The percentage average deviation with respect to the best solution found is also reduced by 0.17% , 0.16% and 0.00% , respectively. The average execution time increases when $\delta = 5$ but decreases for $\delta = 2.5$ and $\delta = 1$. It is important to note that when $\delta = 1$ the algorithm always finds the same solution for every instance over 5 runs.

On the GWKC instances, the percentage reduction on the average arrival compared to the best CCVRP solutions reaches 15.84% , 23.04% and 32.42% , respectively for $\delta = 5, 2.5$ and 1 . The percentage average deviation with respect to the best solution found for the CCVRP model is also reduced by to 0.19% , 0.15% and 0.01% , respectively. This deviation reduction shows a more robust behavior when δ decreases. As for CMT instances, the average execution time increases for $\delta = 5$ to $1\,517.25$ seconds, and decreases for $\delta = 2.5$ and $\delta = 1$ to $1\,215.66$ seconds and 993.76 seconds respectively. Contrary to CMT instances, 12 out of 20 instances have presented the same solution value over 5 runs for $\delta = 1$.

When $\delta \leq 5$ all objective function values are lower than the best values found for CCVRP model, which shows an improved survival rate.

5.5 Concluding Remarks

In this chapter a new VRP version raised from the relief aid delivery in the response phase of humanitarian logistics, called *cumulative capacitated vehicle routing problem with indirect deliveries* (CCVRP-ID), is introduced. This problem relaxes the CCVRP by allowing indirect deliveries to some sites, using an additional transportation mode, instead of including them in the routes performed by the main fleet.

To solve this problem two methods have been proposed: a mathematical model based on the model CCVRP3 presented in Chapter 4 and an MS-ELS metaheuristic which calls a VND and a post-optimization procedure.

The results show that the mathematical model can solve instances optimally up to 30 nodes. However, instances with larger values of δ , the parameter which models the different velocity between transportation modes, become more difficult to solve. Other features like the number of vehicles and the average loading level also impact the execution time required to solve the problem.

On the other hand, MS-ELS succeeds to solve instances with different features. On small instances, when the mathematical model finds the optimal solution, the metaheuristic finds them too. In other cases, the metaheuristic always finds the best known solution. For larger instances the algorithm is competitive compared to CCVRP in terms of execution time and finds good quality solutions for different values of δ .

When δ has a large value, the CCVRP-ID model finds the same solutions as the CCVRP. For smaller values of δ , the problem behaves as a relaxation of the CCVRP and finds solutions with reduced average arrival time, which means a higher survival rate in disaster cases.

Conferences and Publications

Short papers with descriptions of the problem, methods and results introduced in this chapter have been accepted to be presented in one national conference (ROADEF 2015) and one international conference:

- J. C. Rivera, H. M. Afsar and C. Prins. Multistart evolutionary local search for humanitarian relief operations with indirect deliveries. 16ème congrès annuel de la Société française de recherche opérationnelle et d'aide à la décision, ROADEF 2015. Marseille, France. Février, 2015.
- J. C. Rivera, H. M. Afsar and C. Prins. A cumulative capacitated vehicle routing problem with indirect deliveries in relief response operations. Sixth international workshop on freight transportation and logistics, Odysseus 2015. Ajaccio, France. May, 2015.

Chapter 6

General Conclusions and Future Research Directions

This final chapter summarizes the contributions of the thesis and offers promising future research directions.

This thesis is devoted to solve logistic optimization problems raised from humanitarian disasters such as tsunamis, floods and earthquakes. As it can be seen in the state of the art, Operations Research applications in humanitarian logistics has an important growth in last years. Although several models have been proposed for commercial cases, different features and challenges are present in humanitarian contexts due to special constraints, systems characteristics and strategic goals.

Transportation and routing problems are frequently applied on response operations, where urgency and an important number of new demands are the main features. Nevertheless, most of these applications for relief delivery use cost-based objectives. Literature review suggests that developments are still needed with serviced-based objective functions which may better reflect the strategic goal in humanitarian logistics. The problems studied consider the minimization of the sum of arrival times, which is adapted to this context.

The multitrip cumulative capacitated single-vehicle routing problem (mt-CCSVRP) is defined in Chapter 3. In this version of the problem only one vehicle is available, which is able to perform multiple trips with a limited capacity and range. Two mixed integer linear models are formulated to solve this problem. Both are compared with an exact approach based on a reformulation of the mt-CCSVRP as a resource-constrained shortest path problem

in an ad hoc auxiliary graph. While a commercial software solves only a minority of instances for 20 required sites, the proposed shortest path algorithm can tackle instances with up to 40 required sites, thanks to good initial solutions, dominance rules and lower bounds which accelerate the solution procedure. Like many dynamic programming approaches, the proposed exact method reaches its limits for large values of the range L_{max} , which lead to huge state-graphs.

The Chapter 4 introduces the multi-trip cumulative capacitated vehicle routing problem (mt-CCVRP). This problem extends the CCVRP to multiple trips and the mt-CCSVRP of Chapter 3 to several vehicles.

Four mixed integer linear models for the single-trip version (CCVRP) are proposed and compared with the one found in the literature. The best results are found by the model CCVRP3 where variables are defined in terms of the load and arc coefficient of each arc. The mathematical model for the mt-CCVRP is a non trivial mixed integer linear model without vehicle nor multitrip indexes which is tested on small instances.

In addition, three metaheuristics are also developed to solve the mt-CCVRP: MS-ILS, MA|PM and MS-ELS. MS-ILS hybridizes a multi-start ILS with a VND which evaluates some moves in constant time. Five neighborhoods are proposed which use some precomputations to speed up the VND procedure. A dominance rule described in Chapter 3 avoids to consider multiple trip orderings for multitrips. Similar neighborhood structures are used by MA|PM and MS-ELS with the same precomputations and dominance rule.

MA|PM and MS-ELS also perform an adapted two-level split procedure which considers the cumulative objective (the sum of arrival times) and multiple trips. This procedure solves optimally a shortest path problem in polynomial time complexity where the first level defines the set of nodes in each multitrip, and the cost of each multitrip is computed by solving the second level split which defines the set of trips in each multitrip.

On small instances, the resulting metaheuristics find the same results as the mathematical model when the latter can find the optimal solution. They are able to produce competitive results in relatively short computing time.

MS-ILS competes with published methods for the case without multitrips, the CCVRP. A comparison between three metaheuristics is performed for larger mt-CCVRP instances.

The results show the efficacy of the proposed splitting procedure and the developed moves, as well as impact of the fleet size on the average arrival time.

In the Chapter 5 the cumulative capacitated vehicle routing problem with indirect deliveries (CCVRP-ID) is introduced. This problem relaxes the CCVRP by allowing the attachment of sites and delivering relief aid with an auxiliary transportation mode, instead of visiting them.

To solve this problem two methods have been proposed: a mixed integer linear program based on the model CCVRP3 presented in Chapter 4 and an MS-ELS metaheuristic which calls a VND and a post-optimization process.

The results show that the mathematical model can solve instances optimally up to 30 nodes. Although instances with slower auxiliary vehicles become more difficult to solve. Other features like number of vehicles and average loading level also impact the necessary execution time to solve the problem.

The MS-ELS metaheuristic is successful to solve instances with different features. On small instances, when the mathematical model finds an optimal solution, the metaheuristic finds it too. In other cases, the metaheuristic always finds the best known solution. For larger instances the algorithm is competitive compared to CCVRP in terms of execution time and it finds good quality solutions for different auxiliary vehicle velocities.

When auxiliary vehicles are slow enough, optimal solutions for the CCVRP-ID model are identical to the optimal solutions of CCVRP. For faster auxiliary vehicles the problem behaves as a relaxation of the CCVRP and it can find solutions with reduced average arrival time, which means a higher survival rate in disaster cases.

As future research directions, additional models are proposed to enrich the studied problems by introducing more realistic constraints to bring them closer to real situations.

A first extension would be to add the possibility for each vehicle to make multiple tours in the CCVRP-ID model. This would provide a more complex problem, combining the two challenges constituted by the possibility of indirect deliveries and the use of multiple trips (both complications have been addressed separately in the three problems studied in the manuscript).

A second possible perspective would be to manage inventory in CCVRP-ID for the directly visited sites, and then supply the sites visited by auxiliary vehicles. This problem can consider storage capacity on visited sites and multiple periods.

Split deliveries can be also an interesting extension where full loaded vehicles may produce a more efficient use of vehicles and solutions with a lower average arrival time.

The fourth generalization would turn around a dynamic version of the problem with changing circumstances over time, as new affected populations by subsequent aftershocks of a major earthquake. This supposes considering a planning horizon of several days, with reoptimizations to adapt to new situations. This kind of dynamic optimization problem can be very complex when the road network is changed, for example when bridges collapse weakened because of a reply. Shortest paths in the network are affected and then it is even necessary in some cases redirect trucks entered on routes that are blocked.

Finally, a wide research field is offered by multi-criteria generalizations, which must handle several optimization criteria. As a solution minimizing the average access time some sites may still be rescued by an unusually long time, it seems appropriate to consider at the same time minimizing the maximum access time, which would introduce more fairness in the distribution of aid.

Appendix A

Résumé Étendu de la Thèse en Français: Optimisation Logistique pour les Opérations Post-Catastrophe

Chapitre 1 - Introduction générale

La thèse est consacrée à des problèmes d'optimisation logistique soulevés par des catastrophes humanitaires comme des tsunamis, inondations et tremblements de terre. On observe depuis quelques décennies une augmentation du nombre de catastrophes à grande échelle et de leurs impacts humains. Une première explication est le réchauffement climatique qui favorise les incendies de forêt, les inondations ou au contraire les périodes de sécheresse, les glissements de terrains etc. Une deuxième raison est l'augmentation de la population dans des zones exposées, par exemple dans le delta du Gange au Bangladesh où les terres sont très fertiles mais sujettes à de fréquentes inondations. Une troisième raison réside dans le développement de grosses installations industrielles à risques, dont la centrale nucléaire de Fukushima au Japon constitue un exemple typique. Enfin, la mondialisation accentue les conséquences de catastrophes comme les épidémies, qui se propagent plus facilement d'un pays à l'autre, voir l'épidémie récente du virus Ebola en Afrique occidentale.

Outre les conséquences financières énormes de telles catastrophes (destructions et arrêts des activités économiques par exemple), l'impact en termes de vies humaines ou de souffrances diverses est également très important. On peut atténuer les conséquences d'une catastrophe en s'y préparant, par exemple en stockant des moyens de secours, en faisant des simulations ou en entraînant les populations. Mais la plupart des catastrophes sont inévitables et, surtout, imprévisibles, ce qui a motivé un courant de recherche récent pour

augmenter la réactivité et l'efficacité des opérations de secours une fois que le désastre est survenu.

Cette optimisation est critique comme cela a pu être constaté après le séisme qui a détruit Port au Prince en Haïti : un an après la catastrophe, seulement 10% des déblais avaient été évacués, faute d'engins de génie civil en nombre suffisant, ce qui a longtemps bloqué la reconstruction. Les analystes de l'ONU ont souligné que ces problèmes viennent d'une mauvaise organisation logistique sur le terrain : les donateurs ont fourni rapidement beaucoup d'argent et de nombreuses organisations humanitaires se sont rendues sur place, mais sans coordination générale.

La Recherche Opérationnelle avec ses techniques de modélisation et d'optimisation de grands systèmes peut apporter beaucoup à la logistique de catastrophe, à l'instar des succès obtenus en logistique industrielle. Cependant, les méthodes développées en logistique industrielle ne sont pas directement transposables car les critères d'optimisation et beaucoup de contraintes sont différents. Alors que la logistique pour l'industrie cherche essentiellement à réduire les coûts, la logistique de catastrophe vise à remédier à l'urgence, par exemple en distribuant le plus possible de secours par heure, sans oublier des contraintes très particulières comme l'équité entre les personnes affectées.

Le Chapitre 1 d'introduction de la thèse rappelle donc le rôle général de la recherche opérationnelle, les principaux problèmes d'optimisation posés par le transport et les tournées de véhicules, avant d'aborder la logistique humanitaire en soulignant ses caractéristiques et ses différences vis-à-vis de la logistique classique (commerciale).

Le domaine étant très vaste, le chapitre se concentre ensuite sur la phase de réponse à une catastrophe, la seule considérée dans la thèse. En d'autres termes, les problèmes différents posés par les phases de préparation (avant la catastrophe) ou de retour à la normale (longtemps après celle-ci) sortent du cadre de la thèse. Les principaux critères d'optimisation ou d'efficacité sont listés : outre les critères classiques de coût ou de durée totale, on y trouve la minimisation du temps maximal pour atteindre un site sinistré, la minimisation du temps moyen d'intervention et la maximisation du nombre de sites secourus dans une période donnée.

Les problèmes de tournées de véhicules particuliers sur lesquels se concentre la thèse sont finalement présentés. Leur caractéristique commune est un critère d'optimisation spécial, la

minimisation de la somme des temps d'arrivée sur l'ensemble des sites sinistrés, par exemple des villages inondés. A une constante près (le nombre de sites à secourir), ce critère est évidemment équivalent au temps moyen de secours. Il est aussi appelé "coût cumulatif" dans la littérature. Dans le cas d'un taux de mortalité constant par période de temps, il est clair que minimiser le temps moyen de secours permet aussi de maximiser le nombre de survivants.

Par exemple, si un camion atteint (dans l'ordre) un village 2 jours après la catastrophe, puis un autre village 3 jours après le premier, pour retourner à la base en 2 jours, la durée totale de la tournée est $2 + 3 + 2 = 7$ jours, ce qui constitue un critère classique en logistique industriel. En logistique de catastrophe, si on choisit le temps de secours moyen, le premier village est livré en 2 jours et le second en $2 + 3 = 5$ jours, soit un temps de secours moyen de $(2+5)/2 = 3.5$ jours. On voit que le critère "cumulatif" a en général une valeur différente de celle du critère classique et que le temps de retour au dépôt n'est plus considéré.

Lors de la thèse de Sandra Ulrich NGUEVEU au LOSI (soutenue fin 2009), un premier problème de ce type avait été introduit et étudié, le problème cumulatif de tournées de véhicules avec capacités (en anglais *cumulative capacitated vehicle routing problem* ou CCVRP). Comme tous les problèmes de tournées, le CCVRP est un problème difficile d'optimisation combinatoire, appartenant à la classe des problèmes NP-difficiles d'optimisation combinatoire. Cette thèse se penche sur trois nouveaux problèmes avec la même fonction-objectif, qui sont introduits dans le Chapitre 1.

Le premier problème étudié au Chapitre 3 est le problème cumulatif de tournées avec capacités, un véhicule et des tournées multiples (*multitrip cumulative capacitated single-vehicle routing problem* ou mt-CCSVRP). Comparé au CCVRP étudié par Ngueveu *et al.* (2010), on considère un seul véhicule au lieu de plusieurs, mais ce véhicule est autorisé à revenir charger au dépôt pour enchaîner plusieurs tournées (dans la version de Ngueveu, les véhicules partent en parallèle pour effectuer une seule tournée). C'est la séquence de ces tournées successives qui est appelée *multitrip* en anglais. Nous conservons dans la suite le terme anglais qui n'a pas d'équivalent clair en français. Ce mt-CCSVRP est un problème de base intéressant. En effet, la limitation à un seul véhicule, par exemple un hélicoptère, n'est pas si restrictive que ça puisque les gestionnaires de catastrophes découpent souvent le territoire affecté en mailles rectangulaires puis affectent un hélicoptère à chaque maille. Si on suppose que ce découpage est déjà réalisé, il reste à résoudre un mt-CCSVRP dans chaque maille.

Le second problème, abordé au Chapitre 4, est le problème cumulatif de tournées de véhicules avec capacités et tournées multiples (*multitrip cumulative capacitated vehicle routing problem* ou mt-CCVRP). Il peut être vu comme une généralisation à plusieurs véhicules du problème précédent et comme une généralisation à tournées multiples du CCVRP de Ngueveu.

Le troisième problème (Chapitre 5) est le problème cumulatif de tournées de véhicules avec livraisons indirectes (*cumulative vehicle routing problem with indirect deliveries* ou CCVRP-ID). Il concerne la situation réaliste où la flotte de véhicules, par exemple des hélicoptères, n'a pas le temps de visiter tous les sites affectés, par exemple des villages. Cependant, les habitants des sites non visités peuvent venir chercher l'aide qui leur est attribué au plus proche village visité, par exemple en utilisant des véhicules terrestres. L'aide arrive ainsi en deux étapes à ces villages, constituant ainsi des livraisons dites indirectes. L'objectif du problème est donc de choisir les villages visités par les hélicoptères à partir de leur base, de définir une tournée de villages visités pour chaque hélicoptère, et de choisir pour chaque site non visité un village visité à partir duquel une livraison indirecte sera effectuée. L'objectif est toujours de minimiser le temps moyen d'arrivée des secours aux villages, qu'ils soient livrés directement ou indirectement.

Il faut souligner la grande difficulté de ces problèmes d'optimisation de tournées. Dans un problème classique de tournées où il faut minimiser par exemple la longueur totale des tournées (kilométrage), la variation de la longueur totale si un nouveau village k est inséré entre deux villages i et j d'une tournée est simplement $c_{ik} + c_{kj} - c_{ij}$, où c_{ij} désigne la longueur du trajet entre i et j . Avec le critère cumulatif, les calculs sont plus compliqués puisque l'insertion de k retarde les heures d'arrivée à tous les villages qui suivent i dans la tournée existante. De plus, la possibilité de tournées multiples dans les deux premiers problèmes induit une complication supplémentaire. En effet, avec le critère classique de la longueur, la longueur totale d'un "multitrip" ne dépend pas de l'ordre des tournées composant ce multitrip : par exemple, si on affecte à un véhicule trois tournées de 7, 5 et 4 km, ces tournées peuvent être réalisées dans n'importe quel ordre et, de plus, chacune des tournées peut être parcourue à l'envers sans que cela change sa longueur. Dans les problèmes cumulatifs, le temps moyen de secours change si une tournée est inversée ou si on modifie l'ordre des tournées dans un multitrip.

Le Chapitre 1 se termine en annonçant la liste des chapitres de la thèse.

Chapitre 2 - Etat de l'art

Le chapitre a pour objectif de présenter les principaux travaux publiés, en trois blocs: les principales problématiques en logistique humanitaires, les problèmes de tournées de véhicules posés par la phase de réponse après une catastrophe, et un rappel des techniques de modélisation et de résolution des problèmes de tournées de véhicules en général.

La partie sur la logistique humanitaire cite des références utiles, comprenant aussi bien des synthèses introductives que des études très spécialisées, sur les principales décisions pouvant être optimisées dans un système de réponse à une catastrophe : le placement optimal de moyens de secours (véhicules, médicaments, nourriture etc.), la gestion des stocks et les réapprovisionnements, les problèmes de transport (comprenant la distribution d'aide aussi bien que les évacuations), et les problèmes de réhabilitation des infrastructures comme la réparation des ponts ou la restauration de réseaux d'eau ou d'énergie.

La partie sur les problèmes de véhicules cite les publications où les véhicules de secours distribuent l'aide en visitant chacun plusieurs points de livraison. Rappelons que la thèse se limite uniquement à ce type de distribution. Il existe une autre famille de problèmes de transport où chaque véhicule fait des allers et retours entre la base et une seule destination pour livrer des pleines charges. Les problèmes de tournées conviennent aux cas où on livre une quantité plus petite, mais à plusieurs destinations, par exemple pour effectuer une première évaluation en livrant uniquement des médicaments d'urgence. L'état de l'art insiste particulièrement sur les nombreux critères d'optimisation spéciaux qui ont été considérés dans la littérature sur les problèmes de tournées en logistique de catastrophe.

La dernière partie rappelle les principales méthodes de résolution pour les problèmes d'optimisation de tournées : méthodes exactes (résolution directe d'un modèle mathématique, branch-and-bound etc.), heuristiques constructives simples, procédures d'amélioration itérative (recherche locale), métaheuristiques et matheuristiques (hybridation entre une métaheuristique et une méthode exacte).

L'état de l'art montre que la littérature contient beaucoup d'articles de typologie sur les opérations de logistique humanitaire et des discussions prospectives sur les apports possibles de la Recherche Opérationnelle, mais encore relativement peu de travaux appliquant réellement des techniques d'optimisation qui tiennent compte des critères d'urgence spécifiques à ce contexte.

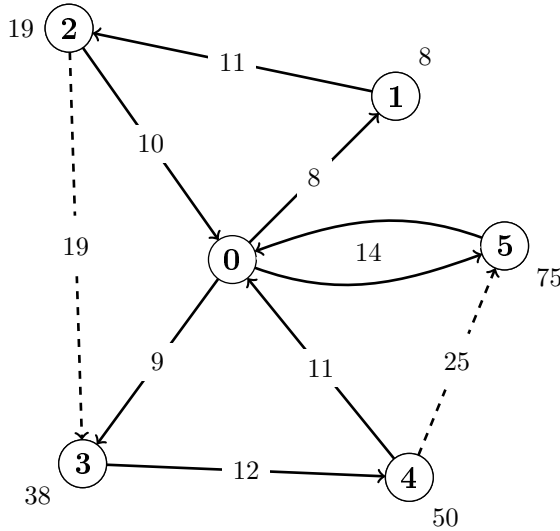
Chapitre 3 - Une méthode exacte pour le mt-CCSVRP

Ce chapitre est consacré au premier problème mentionné dans le Chapitre 1, le problème cumulatif de tournées avec capacités, un seul véhicule et des tournées multiples. Les données comprennent un dépôt comme une base aérienne (nœud d'indice 0), n sites sinistrés ayant chacun des besoins exprimés par une quantité, un véhicule basé au dépôt (par exemple un hélicoptère) et ayant une capacité de transport et une durée de vol maximale (autonomie en temps) sans revenir au dépôt, et les temps de vol entre tout couple de localités (dépôt ou sites affectés). L'hélicoptère peut effectuer des tournées successives en repassant à sa base pour recharger des marchandises et faire le plein de carburant. La durée de chaque tournée ne doit pas dépasser l'autonomie. L'objectif est de déterminer un ensemble de tournées et leur ordre de réalisation, de manière à minimiser la durée moyenne d'accès aux sites à visiter.

Deux modèles mathématiques (des programmes linéaires mixtes) basés sur des techniques différentes sont développés. Le premier est un modèle de type flots. Dans ce type d'approche, les tournées correspondent à des chemins dans un graphe étendu, entre une copie du nœud-dépôt servant de point de départ, et une autre copie servant de point d'arrivée. Les principales variables sont des variables binaires x_{ij} , égales à 1 si une tournée inclut une étape de vol du village i au village j , et des variables $F_{i,j}$ qui expriment la charge du véhicule sur cette étape (ces variables permettent de modéliser les déchargements successifs). Alors que des modèles de la littérature sur les problèmes à tournées multiples nécessitent un indice de tournée pour les variables, par exemple $x_{ijk} = 1$ si l'étape (i, j) est réalisé pendant la tournée numéro k , le modèle proposé ici n'utilise aucun indice de tournée, ce qui n'est pas du tout évident. L'astuce repose sur le concept d'arc de réapprovisionnement (*replenishment arc*) : si une tournée se termine au village i pour revenir au dépôt (nœud 0) et est suivie d'une tournée qui part de 0 pour aller à un premier village j , on peut remplacer les deux arcs $(i, 0)$ et $(0, j)$ par un arc de réapprovisionnement (i, j) , de durée égale au coût des deux arcs substitués. Une solution consiste alors en une séquence unique d'arcs, mixant des étapes réelles de vol et des arcs de réapprovisionnement, ce qui revient à travailler sur une grande tournée unique en s'affranchissant du découpage en tournées successives.

La figure suivante montre une solution pour un cas simple avec 5 sites à visiter à partir du dépôt (nœud 0) et 3 tournées. La demande de chaque site est de 10 tonnes et la capacité du véhicule est de 20 tonnes. Les temps de déplacement sont indiqués sur chaque arc en heures et l'heure d'arrivée est mentionnée près de chaque nœud. Les arcs en pointillés correspondent

aux fameux arcs de réapprovisionnement. La somme des heures d'arrivée aux sites est ici $Z = 190$, à diviser par 5 pour avoir le temps de secours moyen par site sinistré.



Solution:

$[0, 1, 2, 0, 3, 4, 0, 5, 0]$

$Z = 190$

\longrightarrow classical arc

$-\ - \longrightarrow$ replenishment arc

Example of an mt-CCSVRP solution for an instance with $n = 5$, all $q_i = 10$ and $Q = 20$.

Le second modèle est de type partitionnement d'ensemble. L'ensemble des colonnes correspond aux tournées possibles et une variable binaire indique pour chaque colonne si la tournée associée est sélectionnée dans la solution. Par rapport à un problème de tournées classiques, le critère d'optimisation (temps d'accès moyen) dépend de l'ordre de réalisation des tournées par le véhicule, ce qui nécessite des variables binaires supplémentaires $\gamma_{kk'}$, égales à 1 si le véhicule réalise consécutivement les tournées k et k' .

Le chapitre développe ensuite une méthode exacte, équivalente au calcul d'un chemin de coût minimal dans un certain graphe auxiliaire, en présence de ressources consommables. Les nœuds de ce graphe, appelés nœuds-tournées, correspondent aux tournées possibles. Leur nombre peut être a priori énorme, bien que la capacité et l'autonomie limitées du véhicule éliminent les tournées visitant un grand nombre de villages. De plus, deux règles de dominance sont élaborées pour comparer deux tournées visitant le même sous-ensemble de villages mais dans des ordres différents. Ces règles permettent d'éliminer un grand nombre de nœuds-tournées, en prouvant qu'ils ne pourront pas faire partie d'une solution optimale.

Une fois que les nœuds-tournées du graphe auxiliaire ont été définis, un arc (i, j) est créé pour tout couple de tournées i et j pouvant être réalisés consécutivement par le véhicule. Une

condition nécessaire est évidemment que les deux tournées n'aient pas de village commun, car dans notre problème chaque village est visité une seule fois vue l'urgence. Une solution du problème de tournées correspond donc à un chemin (suite de nœuds-tournées) de durée minimale dans le graphe auxiliaire, tel que tous les villages soient visités. Par définition, chaque arc (i, j) implique que les tournées i et j n'aient aucun village en commun, mais cela est insuffisant pour garantir qu'un village visité au début d'un chemin ne soit pas revisité plus tard dans un autre nœud-tournée de ce chemin. L'algorithme de chemin utilisé, de type Bellman-Ford, doit donc mémoriser pour les différents chemins l'ensemble des villages déjà visités pour éviter des visites multiples. Ceci revient à considérer chaque village comme une ressource consommable unique, ce qui conduit à un algorithme à labels multiples.

Il est bien connu que les problèmes de chemins optimaux sont plus faciles quand le graphe n'a pas de circuits, comme en gestion de projet, car on peut alors ordonner les nœuds du graphe depuis les nœuds sans prédécesseurs jusqu'aux nœuds sans successeurs. Il est alors impossible de repasser deux fois par le même nœud. En présence de circuit, il faut un mécanisme supplémentaire pour éviter de repasser par le même nœud, ce qui reviendrait dans notre cas à effectuer deux fois la même tournée.

Ici, une propriété-clé a été démontrée pour aboutir à un graphe sans circuit. Elle stipule que le "coût" (temps d'accès moyen aux sites) d'un multitrip est minimisé si ses tournées sont ordonnées en ordre croissant de durée moyenne des tournées, la durée moyenne d'une tournée k étant définie comme sa durée totale D_k , divisée par le nombre de sites visités M_k . Par conséquent, les chemins qui ne respectent pas cette propriété ne peuvent pas être optimaux. En pratique, on peut donc éliminer tout arc (k, k') du graphe auxiliaire si la durée moyenne de la tournée k est supérieure à celle de k' , ce qui réduit énormément la taille du graphe.

L'algorithme de plus court chemin dans le graphe auxiliaire est finalement accéléré par cinq bornes inférieures sur le coût des chemins, ce qui permet d'éviter d'étendre de nombreux chemins partiels aux cours des calculs.

Les évaluations numériques comparent la résolution directe des deux modèles mathématiques à l'aide du solveur commercial CPLEX avec la méthode de chemin optimal programmée en C++. Les "instances" (problèmes-tests) utilisées sont dérivées d'instances de la littérature pour le problème classique de tournées de véhicules. Ces problèmes appelés CMT comportent de 50 à 199 clients à livrer qui sont interprétés ici comme des villages affectés. Nous conservons les positions géographiques de ces points, les quantités à livrer et

les temps de déplacement. Nous remplaçons cependant la flotte de plusieurs véhicules par un véhicule unique et remplaçons la fonction-objectif classique (durée totale des tournées) par le temps moyen d'arrivée au site sinistré. Nous considérons aussi plusieurs valeurs différentes pour l'autonomie de l'hélicoptère car une grande autonomie augmente le nombre de tournées possibles et donc la taille du graphe auxiliaire.

Les tests montrent que même en 2 heures de calcul les deux modèles mathématiques ne peuvent aller au-delà de 20 villages. Même pour 20 villages, le modèle de type flot ne trouve pas toujours une solution optimale. Le modèle de partitionnement d'ensemble trouve plus de solutions optimales, mais explose dans tous les autres cas par manque de mémoire pendant la résolution. Par contre, la méthode exacte résout optimalement tous les cas à 20 villages en 69 secondes en moyenne.

La méthode exacte est finalement évaluée sur des tests à 25, 30 et 40 sites, en comparant l'impact des règles de dominance et de l'autonomie du véhicule. Les règles de dominance permettent de diviser le nombre de nœuds-tournées du graphe auxiliaire par un factor de 2 à 4 en moyenne. Le nombre de nœuds-tournées du graphe augmente rapidement avec le nombre n de sites et l'autonomie du véhicule. Cependant, les problèmes jusqu'à 400 000 nœuds sont résolus en quelques minutes. La limite de la méthode est atteinte pour $n = 40$ et une grande autonomie: on arrive à presque 4 millions de nœuds avant application des règles de domination, 750 000 avec ces règles, et un temps de calcul atteignant 15 heures.

En conclusion, la méthode exacte est très performante jusqu'à 40 sites pour le problème à un véhicule et tournées multiples, à condition que l'autonomie du véhicule et le nombre moyen de sites livrés par tournée ne soient pas excessifs, ce qui semblent le cas pour un hélicoptère et les quantités relativement importantes à livrer à chaque village en cas de catastrophe.

La méthode exacte développée dans le Chapitre 3 a été présentée avec des résultats préliminaires à la conférence ROADEF 2014 à Bordeaux, en février 2014. Un article détaillé a été soumis à *Discrete Applied Mathematics* (Elsevier), sans retour des évaluateurs au moment où nous écrivons ces lignes.

Chapitre 4 - Le mt-CCVRP : formulations mathématiques et algorithmes de résolution

Rappelons que le mt-CCVRP désigne le problème cumulatif de tournées avec capacités et tournées multiples. Il s'agit d'une généralisation du problème précédent à une flotte de R véhicules de capacité identique. Il s'agit donc pour chaque véhicule de déterminer une séquence de tournées (multitrip) de façon à visiter chaque village affecté une seule fois tout en minimisant le temps d'accès moyen aux villages.

Le chapitre commence par reprendre le CCVRP étudié par Ngueveu *et al.* (2010), qui est une version simplifiée où chaque véhicule n'effectue qu'une tournée. Cette étape préliminaire permet de comparer différentes approches de modélisation mathématique, basées sur des variables différentes, avant d'introduire la complication que constitue la possibilité pour chaque véhicule d'enchaîner des tournées successives.

Cinq programmes linéaires en variables mixtes sont discutés. Le premier est celui proposé par Ngueveu dans sa thèse de 2009, avec des variables binaires indiquant si un chaque arc (i, j) est traversé ou pas par chaque véhicule k . L'utilisation de l'indice de véhicules (k) conduit à un grand nombre de variables. Quatre nouveaux modèles alternatifs sont proposés. Les modèles 2 et 3 sont des approches de type flot avec des variables sans indice de véhicule, comme dans le chapitre précédent. Le modèle 4 utilise des variables binaires spéciales exprimant l'ordre de traversée des arcs : xp_{il}^k , égales à 1 si le site i est visité en position l par le véhicule k . Enfin, le modèle 5 utilise des variables binaires d'enchaînement xb_{ij}^k , égales à 1 si le véhicule k visite consécutivement les sites i puis j .

Le modèle 3 est ensuite sélectionné pour y ajouter la possibilité de tournées multiples pour chaque véhicule. Ce modèle de type flot peut être évidemment vu comme une généralisation à une flotte de plusieurs véhicules du modèle du Chapitre 3, qui concerne le cas d'un véhicule unique. Son principal avantage est de limiter le nombre de variables en n'ayant ni indice de véhicule, ni indice pour distinguer les différentes tournées d'un véhicule, ce qui constitue un tour de force du point de vue de la compacité de modélisation. Comme dans le chapitre précédent, l'astuce de modélisation repose largement sur l'utilisation d'arcs de réapprovisionnement.

Trois métaheuristiques sont ensuite construites pour traiter des problèmes de taille trop grande pour les modèles mathématiques. Toutes partagent la même procédure de recherche

locale, une méthode de descente à voisinage variable (*variable neighborhood descent* ou VND). Le principe des VND en général est d'essayer une suite de voisinages de plus en plus grands, en effectuant des mouvements améliorants dans le voisinage courant tant qu'on peut en trouver, avant de passer au voisinage suivant plus consommateur de temps de calcul. La VND développée pour le mt-CCVRP utilise cinq voisinages. Le voisinage N_1 considère des mouvements 2-OPT et des échanges de deux chaînes contenant chacune de 1 à λ sites (λ -interchanges en anglais), mais appliqués dans une tournée. Le voisinage N_2 considère les mêmes mouvements mais appliqués à une paire de tournées effectuées par le même véhicule, il est donc moins "local". N_3 est basé encore sur les mêmes mouvements mais considère deux tournées de deux véhicules différents. N_4 examine des coupures de tournées tandis que N_5 change l'ordre des tournées d'un véhicule.

Comparé aux problèmes classiques de tournées optimisant la longueur ou la durée totale des tournées, le calcul efficace (rapide) des variations de coût dans ces mouvements est loin d'être évident avec notre critère de temps d'accès moyen aux sites sinistrés. Des formules permettant de calculer en temps constant les variations de coût ont pu être élaborées pour les voisinages N_1 à N_3 . Elles reposent sur le pré-calcul de certaines valeurs avant de balayer les mouvements des voisinages.

La règle de dominance du chapitre précédent est ici toujours valable : l'ordre optimal des tournées dans un multitrip consiste à les trier par durée moyenne croissante. Cependant, ce tri ne peut pas être effectué en temps constant : des essais préliminaires ont montré que la durée d'exécution de la VND était excessif si on applique l'ordre optimal après chaque mouvement dans les voisinages N_1 à N_3 . Les tests ont montré qu'il suffisait de réordonner les tournées d'un véhicule seulement dans N_4 , quand les voisinages précédents ne fournissent plus aucun mouvement améliorant : la qualité finale des solutions ne semble pas significativement diminuée.

La première métaheuristique est une recherche locale itérative à démarrages multiples (multi-start iterative local search ou MS-ILS). La métaheuristique ILS génère une suite d'optima locaux de coûts décroissants. Elle part d'une solution initiale calculée par une heuristique constructive et immédiatement améliorée via la VND. Chaque itération consiste ensuite à prendre une copie de la solution courante, à la perturber aléatoirement (comme l'opérateur de mutation des algorithmes génétiques), à appliquer la VND à la copie perturbée, et à remplacer la solution courante par la solution obtenue, mais seulement en cas d'amélioration. Souvent, les ILS stagnent après quelques itérations : les solutions

obtenues par perturbation et recherche locale n'améliorent plus la solution courante et l'algorithme reste bloqué sur celle-ci. Il vaut mieux alors stopper l'ILS après un certain nombre d'itérations et la redémarrer à partir d'une autre solution initiale dans l'espace de recherche. On utilise pour cela une forme randomisée de l'heuristique constructive initiale, pour obtenir une solution de bonne qualité mais différente à chaque redémarrage.

La deuxième métaheuristique travaille sur une population de solutions, il s'agit d'un algorithme mémétique avec "gestion de population" (memetic algorithm with population management ou MA|PM). Les algorithmes mémétiques (en simplifiant) sont des algorithmes génétiques renforcés par une procédure de recherche locale appliquée aux enfants générés par l'opérateur de croisement. La gestion de population consiste à contrôler la diversité de la population en utilisant une mesure de distance dans l'espace des solutions.

Alors que la première métaheuristique (MS-ILS) travaille directement sur des solutions complètes, MA-PM travaille sur des "tours géants" dans lesquels on ignore les contraintes de capacité. Ce codage de solution peut être vu comme une tournée unique couvrant tous les villages, effectuée par un véhicule de capacité infinie, ou plus simplement comme un ordre de priorité sur les différents villages. La population est formée de tels tours géants, générés au début par des heuristiques, puis par l'opérateur de croisement lors des itérations de l'algorithme mémétique.

La difficulté est évidemment de déduire une solution à notre problème pour chaque tour géant. Pour le problème classique de tournées de véhicules, on peut montrer qu'on peut découper le tour géant en tournées optimalement (sous contrainte de la séquence fournie) et dans un temps de calcul polynomial, en se basant sur la résolution d'un problème de chemin optimal dans un certain graphe auxiliaire. Cette procédure de découpage appelée Split (Prins, 2004) a été utilisée avec succès dans un algorithme mémétique pour les problèmes de tournées classique. L'algorithme mémétique peut donc explorer un espace de recherche plus petit (celui des tours géants) et il n'y a pas de perte d'information puisque la procédure Split permet de déduire rapidement le découpage optimal correspondant à chaque tour géant. De plus, les croisements applicables aux tours géants sont beaucoup plus simples que ceux nécessaires pour des couples de solutions complètes, en particulier la gestion des contraintes de capacité est évacuée. Il est important de préciser que Split découpe optimalement un tour géant donné mais on ne connaît pas le tour géant globalement optimal, c'est-à-dire celui qui va donner après découpage une solution complète optimale pour le problème de tournées qui nous intéresse.

Des procédures Split ont depuis été développées pour des problèmes de tournées plus complexes. La difficulté est que la gestion de contraintes supplémentaires dans le problème de découpage du tour géant en tournées peut le rendre NP-difficile. Pour le mt-CCVRP, nous avons élaboré une procédure Split pas du tout évidente, en préservant la polynomialité. Le tour géant est un simple ordre des sites visités. En simplifiant, la procédure proposée opère en deux niveaux : elle détermine pour chaque sous-séquence de sites, assimilée à un multitrip possible pour un véhicule, son découpage optimal en tournées et le coût correspondant. Puis elle détermine le découpage optimal en multitrips en choisissant des sous-séquences consécutives (déjà évaluées) pour minimiser le coût de la solution complète.

L’algorithme mémétique proposé travaille sur une population de tours géants. A chaque itération deux parents sont sélectionnés par une méthode de roulette favorisant les individus de meilleur coût. Un opérateur de croisement à un point est ensuite appliqué aux deux tours géants choisis et un seul enfant est conservé. Il est décodé par Split pour obtenir une solution complète qui est ensuite améliorée par la même VND que dans la MS-ILS. Le résultat de la VND est finalement reconverti en tour géant, en concaténant les tournées successives des différents véhicules.

La mise à jour de la population obéit aux principes de gestion de la population énoncés par Sörensen & Sevaux (2006). Si l’enfant améliore la meilleure solution connue, il est conservé. Sinon, la distance minimale entre l’enfant et les solutions de la population existante est calculée et l’enfant est accepté seulement si cette distance est au moins égale à un seuil donné Δ . Cette technique empêche que la population soit envahie par des solutions ayant des structures très voisines, ce qui peut conduire à une convergence prématurée. La distance choisie fonctionne sur l’espace des tours géants, il s’agit de la “distance des paires cassées” (Campos *et al.*, 2005) : pour deux tours géants T_1 et T_2 , elle est égale au nombre de paires de sites consécutifs dans T_1 qui sont séparées (cassées) dans T_2 . Cette distance augmente quand le nombre de sous-séquences communes diminue entre les deux tours géants.

Si la solution n’est pas acceptée, elle subit des opérations de mutation successives pour augmenter sa distance à la population, jusqu’à ce qu’elle soit acceptée.

La dernière métaheuristique est une recherche locale évolutionnaire à démarrages multiples (multi-start evolutionary local search ou MS-ELS). La métaheuristique ELS est une ILS dans laquelle un nombre fixé d’enfants sont générés par perturbation + recherche locale en prenant des copies de la solution courante. Si le meilleur enfant obtenu améliore

la solution courante, il remplace cette dernière. En fait, ILS est un cas particulier d'ELS avec un seul enfant par itération. Une autre différence est que notre MS-ELS alterne entre deux espaces de recherche, les tours géants et les solutions complètes du mt-CCVRP, un peu comme dans l'algorithme mémétique.

Plus précisément, chaque redémarrage de MS-ELS utilise une heuristique randomisée pour obtenir une solution complète S , améliorée aussitôt par la VND, et dont les tournées sont concaténées pour obtenir un tour géant initial T . On obtient donc un couple initial (S, T) , avec une solution complète et le tour géant associé, obtenu par concaténation des tournées. Chaque itération consiste ensuite à générer un nombre d'enfants comme suit et à remplacer la solution courante si elle est moins bonne que le meilleur enfant. Une copie du tour géant courant T est effectuée. Cette copie est perturbée aléatoirement en échangeant quelques sites, puis elle est convertie en solution complète S'' avec la même procédure de découpage Split que dans l'algorithme mémétique précédent. S'' est ensuite améliorée par la VND. La meilleure solution enfant S' est mémorisée. Si après la génération du nombre fixé d'enfants, S' est meilleure que la solution courante S , elle remplace cette dernière et le tour géant courant T est mis à jour en concaténant les tournées de S' . Cette technique paraît bien compliquée mais elle est très efficace. En effet, la VND ne contient que des mouvements d'amélioration locale, affectant au plus deux tournées, tandis que le saut temporaire dans l'espace des tours géants (concaténer les tournées puis appliquer Split) permet souvent de changer plusieurs limites de tournées en même temps. L'opérateur Split peut donc être vu comme un opérateur à grand voisinage, renforçant ainsi l'amélioration rapide des solutions.

Comme dans le Chapitre 3, les modèles ont été traduits en CPLEX et tous nos algorithmes en C++. Les cinq modèles mathématiques proposés pour le CCVRP (sans multitrips) ont tout d'abord été testés sur 12 petits cas à 15 sites, les temps de calcul devenant prohibitifs au-delà. Seuls les modèles numéro 3 et 5 résolvent tous les problèmes optimalement, et en moins de 4 minutes. Les autres échouent sur certaines instances, même en leur accordant une heure de calcul. L'intérêt de ces tests est de montrer que nos modèles 3 et 5 sont plus efficaces que le modèle 1 proposé par Ngueveu en 2009 et de nous conforter dans le choix du modèle 3 pour le généraliser au mt-CCVRP qui autorise les multitrips. Le modèle de Ngueveu est cependant meilleur que le modèle 4, qui est le seul à ne trouver aucune solution optimale.

Nous avons ensuite comparé notre MS-ILS sur le CCVRP, avec deux algorithmes mémétiques de Ngueveu *et al.* (2010), une recherche adaptative à large voisinage (ALNS)

de Ribeiro & Laporte (2012) et une métaheuristique en deux phases appelée T-PM (Ke & Feng, 2013). Notre MS-ILS étant conçue pour un problème plus général avec multitrips (le mt-CCVRP), il a fallu les modifier légèrement pour avoir une seule tournée par véhicule et ainsi traiter des instances du CCVRP présentes dans la littérature. Les résultats montrent que MS-ILS est la seconde métaheuristique en termes de qualité de résultats (après T-PM) sur les instances CMT qui comprennent de 50 à 199 sites à visiter, avec des temps de calcul du même ordre. Ceci est remarquable puisque notre algorithme est conçu pour un problème plus général. De plus, les calculs de variation de coûts dans la VND sont bien plus complexes dans le mt-CCVRP, comparé au cas plus simple du CCVRP et nous n'avons pas essayé de modifier notre code pour simplifier les calculs dans le cas du CCVRP.

Nous avons finalement évalué trois métaheuristicues MS-ILS, MA|PM et MS-ELS sur des instances construites pour le mt-CCVRP. 12 petits problèmes-tests à 15 sites ont permis de comparer l'écart des métaheuristicues à l'optimum, en résolvant optimalement le modèle mathématique avec CPLEX. CPLEX parvient à résoudre optimalement 8 problèmes sur les 12 si on lui octroie 1 heure de calcul. Toutes nos métaheuristicues retrouvent ces solutions optimales, en des temps beaucoup plus courts (5 secondes en moyenne).

Des tests sont ensuite menés pour comparer nos trois algorithmes sur des instances dérivées des problèmes-tests pour le CCVRP et disponibles dans la littérature : les CMT (50 à 199 sites) et les GWKC (200 à 483 sites). Sur les CMT les temps de calcul n'excèdent pas 15 minutes et MS-ELS donne toujours la meilleure solution par rapport aux deux autres méthodes, montrant ainsi l'intérêt d'alterner entre les deux espaces de recherche, les tours géants et les solutions complètes.

Sur les très grosses instances GWKC, les temps de calcul augmentent fortement à cause des calculs beaucoup plus lourds que pour un problème de tournées à critère d'optimisation classique. A nombre d'itérations égal, MS-ILS qui est conceptuellement la méthode la plus simple donne en moyenne de meilleures solutions. MS-ELS est presque aussi bonne.

Pour les deux jeux d'instances, CMT et GWKC, l'algorithme mémétique à population donne les moins bons résultats moyens alors que le mécanisme de gestion de la population a la réputation d'être puissant. Il est possible que l'opérateur de croisement utilisée bride l'algorithme et/ou que le travail sur une population de solutions se traduise par une convergence plus lente, comparé à MS-ILS et MS-ELS qui ne considèrent qu'une solution à la fois.

La MS-ILS a été présentée à ROADEF 2012 à Angers, en avril 2012. Elle a été exposée en anglais avec des résultats plus complets à la conférence CORAL 2012 à Benicassim près de Valencia (Espagne). Le modèle mathématique et la MS-ELS ont fait l'objet de communications aux conférences NOW 2013 (Syracuse, Italie) et, avec plus de résultats numériques, à VeRoLog 2013 (EURO Working Group on Vehicle Routing and Logistics, Southampton, Royaume Uni) et EA 2013 (Évolution artificielle, Bordeaux, France).

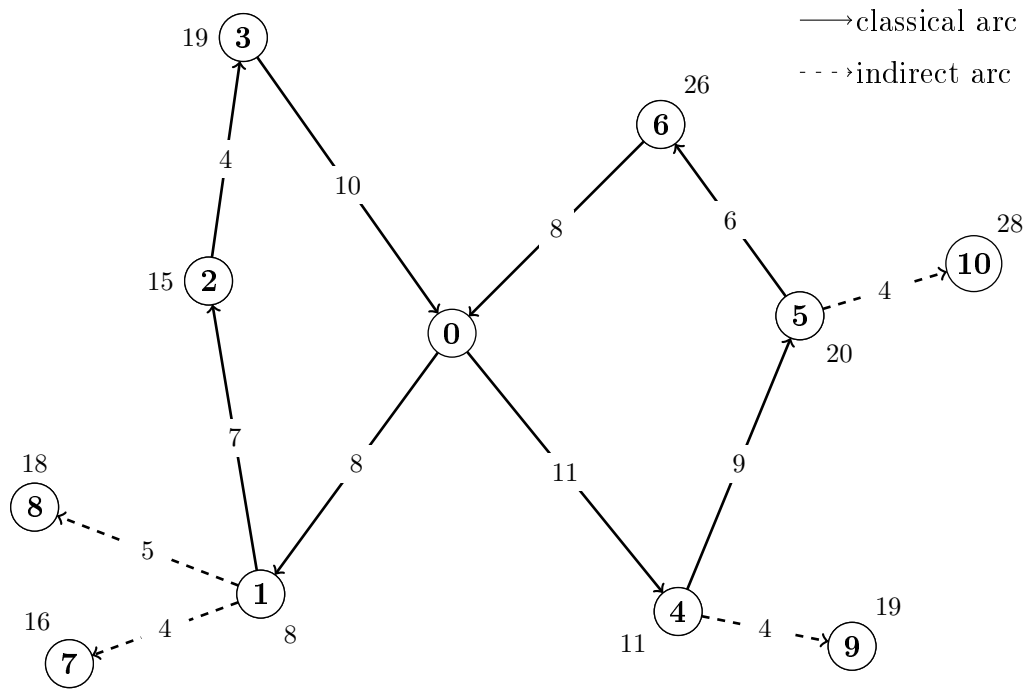
Deux articles plus détaillé ont été publié à Computational optimization and applications and Lecture notes in computer science.

Chapitre 5 - Le problème cumulatif avec capacités et livraisons indirectes (CCVRP-ID)

Rappelons que dans ce problème introduit au Chapitre 1, il est possible de ne pas visiter certains villages, dont l'aide est livrée via une seconde étape de transport (directe) à partir d'un des sites visités. Ce problème est assez réaliste et permet par exemple de distribuer massivement de l'aide à quelques aérodromes encore opérationnels dans la zone du désastre, et de diffuser ensuite cette aide aux villages sinistrés en utilisant par exemple des véhicules terrestres. Les véhicules du premier niveau de distribution sont dits "normaux" et nécessitent un temps donné w_{ij} pour se déplacer entre deux points i et j . Les déplacements du second niveau entre un site i visité par le premier niveau et un site non directement visité j sont effectués par des véhicules dits "auxiliaires", en une durée égale à $\delta \cdot w_{ij}$, où δ est un facteur constant permettant d'indiquer combien de fois les véhicules auxiliaires sont plus lents que les véhicules de la flotte principale. On peut évidemment s'attendre à ce que les solutions optimales puissent changer beaucoup selon la valeur choisie pour le paramètre δ .

La figure suivante donne un exemple simple avec $n = 10$ sites ayant tous une demande de 10 tonnes. On dispose de 2 véhicules "normaux" de capacité 60 tonnes. Les arcs en trait plein indiquent les déplacements effectués par ces véhicules. Les arcs en pointillé correspondent aux livraisons indirectes, en prenant $\delta = 2$, c'est-à-dire que les temps de transport sont supposés deux fois plus lent que si un véhicule normal était utilisé. Les heures d'arrivée figurent près de chaque nœud. La somme des temps d'arrivée pour cette solution est $Z = 180$, à diviser par 10 pour avoir le temps d'accès moyen par site. On peut remarquer que plusieurs sites comme 7 et 8 peuvent être livrés indirectement à partir d'un site visité directement.

Le CCVRP-ID équivaut au CCVRP si les livraisons indirectes sont interdites. Il ressemble au problème généralisé de tournées de véhicules (GVRP) et à certains problèmes



Example of a CCVRP-ID solution for an instance with $n = 10$ and $R = 2$.

de télécommunications comme le “ring-star problem”. Par contre, il ne doit pas être confondu avec les problèmes de tournées à deux échelons, où un second niveau de tournées doit être construit à partir des sites visités à partir de la base principale : en effet, dans notre cas, des livraisons directes sont opérées pour atteindre les villages non visités, par exemple par des véhicules-navettes ou tout simplement par les habitants venant chercher l’aide à pied. De plus, contrairement aux problèmes précédents, on ne considère pas de multitrips.

Après cette présentation du problème, le Chapitre 5 se consacre à la construction d’un modèle mathématique, un programme linéaire mixte. On peut dire qu’il s’agit d’un modèle de type flot utilisant les mêmes variables x_{ij} et F_{ij} que dans les modèles du Chapitre 3 pour le mt-CCSVRP et du Chapitre 4 pour le mt-CCVRP. Une simplification est qu’on ne considère plus les multitrips. Par contre, la livraison directe à chaque site n’est plus obligatoire mais les sites non livrés directement doivent être desservis à partir d’un des sites visités, ce qui nécessite des variables d’affectation supplémentaires.

Une métaheuristique est ensuite étudiée pour traiter des cas hors de portée d’une résolution directe du modèle mathématique. Il s’agit d’une recherche évolutionnaire évolutionnaire à démarrages multiples (MS-ELS). Une MS-ELS alternant entre tours géants et solutions complètes a été développée dans le chapitre précédent, pour résoudre le

mt-CCVRP. La MS-ELS du Chapitre 5 travaille sur des solutions complètes car il semble plus difficile de définir ici la procédure Split chargée de convertir optimalement un ordre sur les sites en une solution complète : il faudrait par exemple décider quels sites sont visités directement ou pas et à quel site visité directement doit être affecté les sites atteints indirectement.

La perturbation dans notre MS-ELS pour le CCVRP-ID consiste à enlever un ensemble de sites choisi aléatoirement pour le réinsérer dans les tournées.

Cette MS-ELS utilise aussi une procédure de descente à voisinage variable (VND) pour améliorer les solutions, mais les voisinages sont évidemment différents de ceux du chapitre précédent puisque nous avons maintenant deux types d'accès aux sites, directs ou indirects. Nous avons considéré six voisinages.

Le premier voisinage N_1 consiste à faire des transformations 2-OPT (inversion d'une sous-séquence de sites) dans une tournée. Le voisinage N_2 travaille aussi sur une tournée à la fois mais échange deux sous-séquences de sites pouvant comporter chacune jusqu'à λ sites. Les voisinages N_3 et N_4 correspondent respectivement à N_1 et N_2 mais travaillent sur des paires de tournées : par exemple, N_4 évalue l'échange d'une sous-séquence d'une tournée avec une autre sous-séquence présente dans une autre tournée. Dans ces quatre voisinages, l'affectation des sites visités indirectement n'est pas modifiée, ce qui est évidemment insuffisant pour envisager d'atteindre une solution optimale. Les deux autres voisinages considèrent les sites visités indirectement (s'ils sont présents dans la solution, évidemment).

Dans le voisinage N_5 , on essaie de réaffecter les nœuds atteints indirectement à un autre nœud visité directement. Dans un problème de tournées classique avec la minimisation de la longueur totale, ce genre de mouvement est inutile puisque l'affectation optimale se fait toujours au site le plus proche visité directement, mais cela n'est plus vrai dans notre version où on souhaite minimiser le temps moyen d'accès au site.

Les mouvements les plus violents sont examinés dans le voisinage N_6 qui consistent à modifier le mode de livraison (direct/indirect) d'un site. Un site visité directement peut donc être enlevé de sa tournée pour être atteint indirectement à partir d'un autre site, et un site visité indirectement peut être inséré dans une tournée de livraison directe si cela est profitable.

Une opération de post-optimisation plus globale a été incluse en fin d'algorithme. Elle consiste à enlever tous les clients visités indirectement, à appliquer la VND aux sites restants, puis à réinsérer au mieux les clients qui ont été enlevés. Ce processus est répété tant qu'on peut améliorer la solution.

Les tests numériques comportent deux parties. La première partie considère 12 problèmes-tests avec 15 sites et 12 autres avec 30 sites. Elle vise à comparer l'écart entre les solutions de la MS-ELS et la solution optimale calculée en résolvant le modèle mathématique avec CPLEX. La seconde partie utilise les 14 instances CMT (50 à 199 sites) et les 20 instances GWKC (200 à 483 sites) pour le CCVRP, déjà utilisées dans le chapitre précédent. Elles sont évidemment modifiées en autorisant maintenant les livraisons indirectes et en introduisant le facteur δ pour la durée de transport dans les livraisons indirectes. Ces instances sont évidemment trop grosses pour le modèle mathématique.

Dans les deux parties, on en profite pour comparer les coûts du CCVRP-ID avec le CCVRP où les livraisons indirectes sont interdites. On teste aussi 5 valeurs différentes du paramètre δ : 1, 2.5, 5, 10 et 10 000. Le cas 10 000 est une valeur extrême visant à empêcher (en les pénalisant fortement) les livraisons indirectes, ce qui ramène le CCVRP-ID au CCVRP classique. L'autre valeur extrême (1) considère les mêmes temps de transport sur un arc, que ce soit dans une livraison directe ou indirecte.

Pour les petits problèmes de la première partie, CPLEX arrive à résoudre tous les cas à 15 sites en deux minutes maximum, pour toutes les valeurs de δ . Pour les cas à 30 sites, CPLEX trouve des solutions optimales seulement pour $\delta = 1$ ou 2.5, si on lui accorde une heure de calcul. Sur toutes les instances (à 15 ou 30 sites), la métaheuristique MS-ELS retrouve toutes les solutions optimales obtenues par CPLEX.

Pour les grands problèmes de la seconde partie, avec $\delta = 1\ 000$, la MS-ELS retrouve toutes les meilleures solutions connues de la littérature pour le CCVRP sur les instances CMT. Pour les instances GWKC, plus grosses, elle retrouve 4 des meilleures solutions connues et en améliore 2. Pour les valeurs inférieures de δ qui induisent un vrai CCVRP-ID, on observe un gain moyen qui augmente quand δ diminue. Par exemple, la réduction du temps d'accès moyen aux sites atteint 26% pour $\delta = 1$ dans le cas des instances CMT.

Ces résultats montrent d'une part que la MS-ELS développée pour le CCVRP-ID est compétitive avec les meilleures métaheursistiques publiées sur le cas particulier que constitue

le CCVRP, où les livraisons indirectes sont interdites. D'autre part, les livraisons indirectes permettent de diminuer le temps d'accès moyen aux sites pour un large éventail de valeurs de δ .

La MS-ELS pour le CCVRP-ID a été acceptée pour être présentée à ROADEF 2015 à Marseille et à ODYSSEUS 2015 à Ajaccio.

Chapitre 6 - Conclusion et perspectives

Ce dernier chapitre récapitule les contributions de la thèse et propose des perspectives de recherche prometteuses. Trois nouveaux problèmes de tournées inspirés par la logistique de catastrophe ont été définis, modélisés mathématiquement et résolus exactement par un solveur de programmation linéaire (pour les petits cas) et par des métaheuristiques ad hoc pour des cas réalistes, allant jusqu'à 483 sites sinistrés. Les problèmes étudiés considèrent une fonction-objectif adaptée au contexte et reflétant l'urgence de la situation, le temps d'accès moyen aux sites sinistrés. Ils envisagent aussi des modes de fonctionnement réalistes, comme le réemploi des véhicules pour effectuer des rotations successives sous contrainte d'autonomie ou des livraisons en deux temps, par exemple en mode aérien puis terrestre. Des techniques variées de Recherche Opérationnelle et d'algorithmique avancée ont été utilisées. En particulier, des techniques de prétraitement sophistiquées ont dû être utilisées pour évaluer en temps raisonnable les variations de coût de la solution lors des mouvements de recherche locale.

Les perspectives proposées consistent à enrichir les problèmes étudiés en introduisant plus de contraintes réalistes, pour les rapprocher des situations réelles.

Une première extension consisterait à ajouter au CCVRP-ID la possibilité pour chaque véhicule de faire des tournées multiples. On obtiendrait ainsi un problème plus complexe, combinant les deux difficultés constituées par la possibilité de livraisons indirectes et l'utilisation de tournées multiples (ces deux complications ont été abordées séparément dans les trois problèmes étudiés dans la thèse).

Une seconde perspective possible serait de gérer des stocks dans le CCVRP-ID, au niveau des sites visités directement, pour ensuite alimenter les sites visités dans un deuxième temps.

La troisième généralisation tournerait autour d'une version dynamique des problèmes,

avec une évolution de la situation au cours du temps, comme de nouveaux villages affectés par des répliques ultérieures d'un tremblement de terre principal. Ceci suppose de considérer un horizon de planification de plusieurs jours, avec des réoptimisations pour s'adapter aux nouvelles situations. Ce genre de problème d'optimisation dynamique peut être très complexe quand le réseau routier est modifié, par exemple quand des ponts fragilisés s'écroulent à cause d'une réplique. Les plus courts chemins dans le réseau sont alors impactés et il faut même dans certains cas réorienter les camions engagés sur des itinéraires qui se retrouvent bloqués.

Enfin, un large terrain d'investigation est offert par des généralisations multi-critères, où on doit optimiser au sens de Pareto plusieurs critères d'optimisation. Comme dans une solution minimisant le temps d'accès moyen certains sites peuvent quand même être secourus en un temps anormalement long, il nous paraît opportun de considérer en même temps la minimisation du temps d'accès maximal, ce qui permettrait d'introduire plus d'équité dans la distribution de l'aide.

Bibliography

- Abdelgawad, Hossam, & Abdulhai, Baher. 2009. Emergency evacuation planning as a network design problem: a critical review. *Transportation Letters: The International Journal of Transportation Research*, **1**(1), 41–58.
- Abounacer, Rachida, Rekik, Monia, & Renaud, Jacques. 2014. An exact solution approach for multi-objective location-transportation problem for disaster response. *Computers & Operations Research*, **41**(0), 83–93.
- Afsar, H. Murat, Prins, Christian, & Santos, Andréa Cynthia. 2014. Exact and heuristic algorithms for solving the generalized vehicle routing problem with flexible fleet size. *International Transactions in Operational Research*, **21**(1), 153–175.
- Ahuja, Ravindra K., Ergun, Özlem, Orlin, James B., & Punnen, Abraham P. 2002. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, **123**(1–3), 75–102.
- Aksu, Dilek Tuzun, & Özdamar, Linet. 2014. A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation. *Transportation Research Part E: Logistics and Transportation Review*, **61**(0), 56–67.
- Allahviranloo, Mahdieh, Chow, Joseph Y. J., & Recker, Will W. 2014. Selective vehicle routing problems under uncertainty without recourse. *Transportation Research Part E: Logistics and Transportation Review*, **62**(0), 68–88.
- Allaoua, Hanane, Borne, Sylvie, Létocart, Lucas, & Calvo, Roberto Wolfler. 2013. A matheuristic approach for solving a home health care problem. *Electronic Notes in Discrete Mathematics*, **41**(0), 471–478.
- Almoustafa, Samira, Hanafi, Said, & Mladenović, Nenad. 2013. Multistart Branch and Bound for Large Asymmetric Distance-Constrained Vehicle Routing Problem. *Pages 15–38 of: Migdalas, Athanasios, Sifaleras, Angelo, Georgiadis, Christos K., Papathanasiou, Jason, & Stiakakis, Emmanuil (eds), Optimization Theory, Decision Making, and Operations*

Research Applications. Springer Proceedings in Mathematics & Statistics, vol. 31. Springer New York.

Altay, Nezh, & Green III, Walter G. 2006. OR/MS research in disaster operations management. *European Journal of Operational Research*, **175**(1), 475–493.

Altinkemer, Kemal, & Gavish, Bezalel. 1991. Parallel Savings Based Heuristics for the Delivery Problem. *Operations Research*, **39**(3), 456–469.

Alvarenga, G.B., Mateus, G.R., & de Tomi, G. 2007. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, **34**(6), 1561–1584.

Angel-Bello, Francisco, Martínez-Salazar, Iris, & Alvarez, Ada. 2013. Minimizing waiting times in a route design problem with multiple use of a single vehicle. *Electronic Notes in Discrete Mathematics*, **41**(0), 269–276.

Applegate, David, Cook, William, Dash, Sanjeeb, & Rohe, André. 2002. Solution of a Min-Max Vehicle Routing Problem. *INFORMS Journal on Computing*, **14**(2), 132–143.

Apte, Aruna. 2009. Humanitarian Logistics: A New Field of Research and Action. *Foundations and Trends in Technology, Information and Operations Management*, **3**(1), 1–100.

Archer, Aaron, & Williamson, David P. 2003. Faster Approximation Algorithms for the Minimum Latency Problem. In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Archetti, Claudia, Bianchessi, Nicola, & Speranza, M. Grazia. 2014. Branch-and-cut algorithms for the split delivery vehicle routing problem. *European Journal of Operational Research*, **238**(3), 685 – 698.

Ardekani, S. A., & Hobeika, A. G. 1988. Logistics Problems in the Aftermath of the 1985 Mexico City Earthquake. *Transportation Quarterly*, **42**, 107–124.

Arkin, Esther M., Hassin, Refael, & Levin, Asaf. 2006. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, **59**(1), 1–18.

Arora, S., & Karakostas, G. 2000. A $2 + \epsilon$ approximation algorithm for the k -MST problem. *Pages 754–759 of: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*.

- Auf-Der-Heide, Erik. 1989. *Disaster Response: Principles of Preparation and Coordination*. C.V. Mosby Company.
- Azi, Nabila, Gendreau, Michel, & Potvin, Jean-Yves. 2007. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research*, **178**(3), 755–766.
- Başar, Ayfer, Çatay, Bülent, & Ünlüyurt, Tonguç. 2012. A taxonomy for emergency service station location problem. *Optimization Letters*, **6**(6), 1147–1160.
- Baker, Barrie M., & Sheasby, Janice. 1999. Extensions to the generalised assignment heuristic for vehicle routing. *European Journal of Operational Research*, **119**(1), 147–157.
- Bakuli, David L., & Smith, J. MacGregor. 1996. Resource allocation in state-dependent emergency evacuation networks. *European Journal of Operational Research*, **89**(3), 543–555.
- Balcik, Burcu, & Beamon, Benita M. 2008. Facility Location in Humanitarian Relief. *International Journal of Logistics: Research and Applications*, **11**(2), 101–121.
- Balcik, Burcu, Beamon, Benita M., & Smilowitz, Karen. 2008. Last Mile Distribution in Humanitarian Relief. *Journal of Intelligent Transportation Systems*, **12**, 51–63.
- Baldacci, R., Hadjiconstantinou, E., & Mingozzi, A. 2004. An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a Two-Commodity Network Flow Formulation. *Operations Research*, **52**(5), 723–738.
- Balinski, M. L., & Quandt, R. E. 1964. On an Integer Program for a Delivery Problem. *Operations Research*, **12**(2), 300–304.
- Barbarosoğlu, Gülay, Özdamar, Linet, & Çevik, Ahmet. 2002. An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. *European Journal of Operational Research*, **140**(1), 118–133.
- Batista-Galván, María, Riera-Ledesma, Jorge, & Salazar-González, Juan José. 2013. The traveling purchaser problem, with multiple stacks and deliveries: A branch-and-cut approach. *Computers & Operations Research*, **40**(8), 2103 – 2115.
- Beamon, B. M. 2004. Humanitarian relief chains: issues and challenges. *In: Proceedings of the 34th International Conference on Computers and Industrial Engineering*.
- Beamon, Benita M., & Kotleba, Stephen A. 2006a. Inventory management support systems for emergency humanitarian relief operations in South Sudan. *International Journal of Logistics Management*, **17**(2), 187–212.

- Beamon, Benita M., & Kotleba, Stephen A. 2006b. Inventory Modeling for Complex Emergencies in Humanitarian Relief Operations. *International Journal of Logistics: Research and Applications*, **9**(1), 1–18.
- Beasley, JE. 1983. Route first - Cluster second methods for vehicle routing. *Omega*, **11**(4), 403–408.
- Belardo, S., Harrald, J., Wallace, W. A., & Ward, J. 1984. A Partial Covering Approach to Siting Response Resources for Major Maritime Oil Spills. *Management Science*, **30**(10), 1184–1196.
- Belenguer, José-Manuel, Benavent, Enrique, Labadi, Nacima, Prins, Christian, & Reghioui, Mohamed. 2010. Split-Delivery Capacitated Arc-Routing Problem: Lower Bound and Metaheuristic. *Transportation Science*, **44**(2), 206–220.
- Belenguer, José-Manuel, Benavent, Enrique, Prins, Christian, Prodhon, Caroline, & Calvo, Roberto Wolfler. 2011. A Branch-and-Cut method for the Capacitated Location-Routing Problem. *Computers & Operations Research*, **38**(6), 931 – 941.
- Bell, Michael G. H., Fonzone, Achille, & Polyzoni, Chrisanthi. 2013. Depot location in degradable transport networks. *Transportation Research Part B: Methodological*.
- Benavent, Enrique, Corberán, Ángel, Plana, Isaac, & Sanchis, José M. 2009. Min-Max K-vehicles windy rural postman problem. *Networks*, **54**, 216–226.
- Benavent, Enrique, Corberán, Ángel, & Sanchis, José M. 2010. A metaheuristic for the min-max windy rural postman problem with K vehicles. *Computational Management Science*, **7**(3), 269–287.
- Bennett, Brian T., & Gazis, Denos C. 1972. School bus routing by computer. *Transportation Research*, **6**(4), 317–325.
- Berinsky, Hernán, & Zabala, Paula. 2011. An Integer Linear Programming Formulation and Branch-and-Cut Algorithm for the Capacitated m -Ring-Star Problem. *Electronic Notes in Discrete Mathematics*, **37**(0), 273–278.
- Bianco, Lucio, Mingozzi, Aristide, & Ricciardelli, Salvatore. 1993. The traveling salesman problem with cumulative costs. *Networks*, **23**(2), 81–91.
- Blecken, Alexander. 2010. *Humanitarian logistics. Modelling supply chain processes of humanitarian organisations*. Haupt Verlag.

- Blum, Avrim, Chalasani, Prasad, Coppersmith, Don, Pulleyblank, Bill, Raghavan, Prabhakar, & Sudan, Madhu. 1994. The minimum latency problem. *Pages 163–171 of: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. STOC '94. New York, NY, USA: ACM.
- Boland, N. L., Clarke, L. W., & Nemhauser, G. L. 2000. The asymmetric traveling salesman problem with replenishment arcs. *European Journal of Operational Research*, **123**(2), 408–427.
- Boschetti, Marco A., Maniezzo, Vittorio, Roffilli, Matteo, & Bolufé Röhler, Antonio. 2009. Matheuristics: Optimization, Simulation and Control. *Pages 171–177 of: Blesa, María J., Blum, Christian, Di Gaspero, Luca, Roli, Andrea, Sampels, Michael, & Schaerf, Andrea (eds), Hybrid Metaheuristics*. Lecture Notes in Computer Science, vol. 5818. Springer Berlin Heidelberg.
- Bramel, Julien, & Simchi-Levi, David. 1995. A Location based Heuristic for General Routing Problems. *Operations Research*, **43**(4), 649–660.
- Brandão, José, & Mercer, Alan. 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, **100**(1), 180–191.
- Brandão, José, & Mercer, Alan. 1998. The Multi-Trip Vehicle Routing Problem. *Journal of the Operational Research Society*, **49**(8), 799–805.
- Brandeau, Margaret L., Sainfort, François, & Pierskalla, William P. 2004. Health Care Delivery: Current Problems and Future Challenges. *Pages 1–14 of: Brandeau, Margaret L., Sainfort, François, & Pierskalla, William P. (eds), Operations Research and Health Care*. International Series in Operations Research & Management Science, vol. 70. Springer US.
- Breedam, A. Van. 1994. *An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints*. Ph.D. thesis, University of Antwerp.
- Brotcorne, Luce, Laporte, Gilbert, & Semet, Frédéric. 2003. Ambulance location and relocation models. *European Journal of Operational Research*, **147**(3), 451–463.
- Bryson, Kweku-Muata (Noel), Millar, Harvey, Joseph, Anito, & Mobolurin, Ayodele. 2002. Using formal MS/OR modeling to support disaster recovery planning. *European Journal of Operational Research*, **141**(3), 679–688.

- Calvete, Herminia I., Galé, Carmen, & Iranzo, José A. 2013. An efficient evolutionary algorithm for the ring star problem. *European Journal of Operational Research*, **231**(1), 22–33.
- Campbell, Ann Melissa, Vandenbussche, Dieter, & Hermann, William. 2008. Routing for Relief Efforts. *Transportation Science*, **42**(2), 127–145.
- Campos, Vania, Bandeira, Renata, & Bandeira, Adriano. 2012. A Method for Evacuation Route Planning in Disaster Situations. *Procedia - Social and Behavioral Sciences*, **54**(0), 503–512. Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation, September 2012, Paris.
- Campos, Vicente, Laguna, Manuel, & Martí, Rafael. 2005. Context-Independent Scatter and Tabu Search for Permutation Problems. *INFORMS Journal on Computing*, **17**(1), 111–122.
- Caserta, Marco, & Voß, Stefan. 2010. Metaheuristics: Intelligent Problem Solving. *Pages 1–38 of: Maniezzo, Vittorio, Stützle, Thomas, & Voß, Stefan (eds), Matheuristics. Annals of Information Systems*, vol. 10. Springer US.
- Cattaruzza, Diego, Absi, Nabil, Feillet, Dominique, & Vidal, Thibaut. 2014. A memetic algorithm for the Multi Trip Vehicle Routing Problem. *European Journal of Operational Research*, **236**(3), 833–848. Vehicle Routing and Distribution Logistics.
- Caunhye, Aakil M., Nie, Xiaofeng, & Pokharel, Shaligram. 2012. Optimization models in emergency logistics: A literature review. *Socio-Economic Planning Sciences*, **46**(1), 4–13. Special Issue: Disaster Planning and Logistics: Part 1.
- Chaiken, Jan M., & Larson, Richard C. 1972. Methods for Allocating Urban Emergency Units: A Survey. *Management Science*, **19**(4), 110–130.
- Chao, I-Ming. 2002. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, **29**(1), 33–51.
- Charnes, A, Cooper, W. W., Harrald, J., Karwan, K. R., & Wallace, W. A. 1976. A goal interval programming model for resource allocation in a marine environmental protection program. *Journal of Environmental Economics and Management*, **3**(4), 347–362.
- Charnes, A., Cooper, W. W., Karwan, K. R., & Wallace, W. A. 1979. A chance-constrained goal programming model to evaluate response resources for marine pollution disasters. *Journal of Environmental Economics and Management*, **6**(3), 244–274.

- Chen, Huey-Kuo, Hsueh, Che-Fu, & Chang, Mei-Shiang. 2006. The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, **42**(5), 383–408.
- Chen, Ping, Dong, Xingye, & Niu, Yanchao. 2012. An Iterated Local Search Algorithm for the Cumulative Capacitated Vehicle Routing Problem. *Pages 575–581 of: Tan, Honghua (ed), Technology for Education and Learning. Advances in Intelligent Systems and Computing*, vol. 136. Springer Berlin Heidelberg.
- Choi, Eunjeong, & Tcha, Dong-Wan. 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, **34**(7), 2080 – 2095.
- Chou, Jui-Sheng, Tsai, Chih-Fong, Chen, Zong-Yao, & Sun, Ming-Hui. 2014. Biological-based genetic algorithms for optimized disaster response resource allocation. *Computers & Industrial Engineering*, **74**(0), 52–67.
- Christofides, N., & Eilon, S. 1969. An Algorithm for the Vehicle-Dispatching Problem. *Operational Research Quarterly*, **20**(3), 309–318.
- Christofides, N., Mingozzi, A., & Toth, P. 1979. The vehicle routing problem. *Pages 315–338 of: Christofides, N., Mingozzi, A., Toth, P., & Sandi, L. (eds), Combinatorial optimization. Wiley, Chichester.*
- Christofides, N., Mingozzi, A., & Toth, P. 1981a. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, **20**(1), 255–282.
- Christofides, Nicos. 1976. The vehicle routing problem. *RAIRO - Operations Research - Recherche Opérationnelle*, **10**(V1), 55–70.
- Christofides, Nicos, Mingozzi, A., & Toth, P. 1981b. State-space relaxation procedures for the computation of bounds to routing problems. *Networks*, **11**(2), 145–164.
- Chu, Feng, Labadi, Nacima, & Prins, Christian. 2006. A Scatter Search for the periodic capacitated arc routing problem. *European Journal of Operational Research*, **169**(2), 586–605.
- Clarke, G., & Wright, J. W. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, **12**(4), 568–581.

- Coelho, Leandro C., Cordeau, Jean-François, & Laporte, Gilbert. 2012. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, **24**(0), 270–287.
- Cordeau, J-F, Laporte, G., & Mercier, A. 2001. A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. *The Journal of the Operational Research Society*, **52**(8), 928–936.
- Cordeau, Jean-François, & Maischberger, Mirko. 2012. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, **39**(9), 2033–2050.
- Cordeau, Jean-François, Gendreau, Michel, & Laporte, Gilbert. 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, **30**(2), 105–119.
- Crainic, Teodor Gabriel, Perboli, Guido, Mancini, Simona, & Tadei, Roberto. 2010. Two-Echelon Vehicle Routing Problem: A satellite location analysis. *Procedia - Social and Behavioral Sciences*, **2**(3), 5944–5955.
- Current, John R., & Schilling, David A. 1994. The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, **73**(1), 114–126.
- Das, Rubel, & Hanaoka, Shinya. 2014. Relief inventory modelling with stochastic lead-time and demand. *European Journal of Operational Research*, **235**(3), 616–623.
- Davis, Lauren B., Samanlioglu, Funda, Qu, Xiuli, & Root, Sarah. 2013. Inventory planning and coordination in disaster relief efforts. *International Journal of Production Economics*, **141**(2), 561–573. Special Issue on Service Science.
- Dayarian, Iman, Crainic, Teodor Gabriel, Gendreau, Michel, & Rei, Walter. 2014. A branch-and-price approach for a multi-period vehicle routing problem. *Computers & Operations Research*, –.
- De Angelis, Vanda, Mecoli, Mariagrazia, Nikoi, Chris, & Storchi, Giovanni. 2007. Multiperiod integrated routing and scheduling of World Food Programme cargo planes in Angola. *Computers & Operations Research*, **34**(6), 1601–1615.
- Dell, Robert F., Batta, Rajan, & Karwan, Mark H. 1996. The Multiple Vehicle TSP with Time Windows and Equity Constraints over a Multiple Day Horizon. *Transportation Science*, **30**(2), 120–133.

- Derigs, U., & Kaiser, R. 2007. Applying the attribute based hill climber heuristic to the vehicle routing problem. *European Journal of Operational Research*, **177**(2), 719–732.
- Dhahri, Amine, Zidi, Kamel, & Ghedira, Khaled. 2014. Variable Neighborhood Search based Set Covering ILP Model for the Vehicle Routing Problem with Time Windows. *Procedia Computer Science*, **29**(0), 844–854. 2014 International Conference on Computational Science.
- Drexler, Michael. 2013. Applications of the vehicle routing problem with trailers and transshipments. *European Journal of Operational Research*, **227**(2), 275–283.
- Dueck, Gunter. 1993. New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel. *Journal of Computational Physics*, **104**(1), 86–92.
- Dueck, Gunter, & Scheuer, Tobias. 1990. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, **90**(1), 161–175.
- Duhamel, Christophe, Lacomme, Philippe, Prins, Christian, & Prodhon, Caroline. 2010. A GRASP \times ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, **37**(11), 1912–1923.
- Duhamel, Christophe, Lacomme, Philippe, Quilliot, Alain, & Toussaint, H el ene. 2011. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, **38**(3), 617–640.
- Duhamel, Christophe, Lacomme, Philippe, & Prodhon, Caroline. 2012. A hybrid evolutionary local search with depth first search split procedure for the heterogeneous vehicle routing problems. *Engineering Applications of Artificial Intelligence*, **25**(2), 345–358.
- Eilon, S., Watson-Gandy, C., & Christofides, N. 1971. *Distribution Management: Mathematical Modeling and Practical Analysis*. Griffin.
- Ergun,  ozlem, Orlin, James B., & Steele-Feldman, Abran. 2006. Creating very large scale neighborhoods out of smaller ones by compounding moves. *Journal of Heuristics*, **12**(1-2), 115–140.
- Fakcharoenphol, J., Harrelson, C., & Rao, S. 2003. The k -traveling repairman problem. *In: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*.

Fakcharoenphol, Jittat, Harrelson, Chris, & Rao, Satish. 2007. The k -traveling Repairmen Problem. *ACM Trans. Algorithms*, **3**(4).

Falasca, Mauro, & Zobel, Christopher. 2012. An optimization model for volunteer assignments in humanitarian organizations. *Socio-Economic Planning Sciences*, **46**(4), 250–260.

Fallahi, Abdellah El, Prins, Christian, & Calvo, Roberto Wolfer. 2008. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, **35**(5), 1725–1741.

Farahani, Reza Zanjirani, Asgari, Nasrin, Heidari, Nooshin, Hosseininia, Mahtab, & Goh, Mark. 2012. Covering problems in facility location: A review. *Computers & Industrial Engineering*, **62**(1), 368–407.

Feillet, Dominique, Dejax, Pierre, Gendreau, Michel, & Gueguen, Cyrille. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, **44**(3), 216–229.

Fischer III, H.W. 2003. *A Proposed Disaster Scale*. Tech. rept. Millersville University of Pennsylvania.

Fischetti, Matteo, Laporte, Gilbert, & Martello, Silvano. 1993. The delivery man problem and cumulative matroids. *Operations Research*, **41**(6), 1055–1064.

Fischetti, Matteo, Toth, Paolo, & Vigo, Daniele. 1994. A Branch-And-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs. *Operations Research*, **42**(5), pp. 846–859.

Fisher, Marshall L. 1994. Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. *Operations Research*, **42**(4), 626–642.

Fisher, Marshall L., & Jaikumar, Ramchandran. 1981. A generalized assignment heuristic for vehicle routing. *Networks*, **11**(2), 109–124.

Fleischmann, B. 1990. *The Vehicle Routing Problem with Multiple Use of Vehicles*. Working paper. Fachbereich Wirtschaftswissenschaften Universität Hamburg.

Fleury, Gérard, Lacomme, Philippe, Prins, Christian, & Sevaux, Marc. 2008. *A bi-objective stochastic approach for stochastic CARP*. Tech. rept. Research Report LIMOS/ RR-08-06.

- Foster, B. A., & Ryan, D. M. 1976. An Integer Programming Approach to the Vehicle Scheduling Problem. *Operational Research Quarterly (1970-1977)*, **27**(2), 367–384.
- Franceschi, Roberto De, Fischetti, Matteo, & Toth, Paolo. 2006. A new ILP-based refinement heuristic for Vehicle Routing Problems. *Mathematical Programming*, **105**(2-3), 471–499.
- Fu, L., Sun, D., & Rilett, L.R. 2006. Heuristic shortest path algorithms for transportation applications: State of the art. *Computers & Operations Research*, **33**(11), 3324–3343.
- Fukasawa, Ricardo, Longo, Humberto, Lysgaard, Jens, Aragão, Marcus Poggi de, Reis, Marcelo, Uchoa, Eduardo, & Werneck, Renato F. 2006. Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. *Mathematical Programming*, **106**(3), 491–511.
- Furuta, Takehiro, & Tanaka, Ken-ichi. 2010. A maximal covering model for helicopter emergency medical systems. *Pages 324–331 of: The ninth international symposium on operations research and its applications.*
- Galindo, Gina, & Batta, Rajan. 2013. Review of recent developments in OR/MS research in disaster operations management. *European Journal of Operational Research*, **230**(2), 201–211.
- Garg, N. 1996. A 3-approximation for the minimum tree spanning k vertices. *Pages 302–309 of: Proceedings of the 37th Annual Symposium on Foundations of Computer Science.*
- Gauvin, Charles, Desaulniers, Guy, & Gendreau, Michel. 2014. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, **50**(0), 141 – 153.
- Gavish, B., & Graves, S.C. 1979. *The travelling salesman problem and related problems.* Tech. rept. Graduate School of Management. University of Rochester.
- Gendreau, Michel, Hertz, Alain, & Laporte, Gilbert. 1994. A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, **40**(10), 1276–1290.
- Gillett, Billy E., & Miller, Leland R. 1974. A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research*, **22**(2), 340–349.
- Glover, F., & Punnen, A. P. 1997. The Travelling Salesman Problem: New Solvable Cases and Linkages with the Development of Approximation Algorithms. *The Journal of the Operational Research Society*, **48**(5), 502–510.

- Goemans, M., & Kleinberg, J. 1998. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, **82**, 111–124.
- Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. 1977. Implementing vehicle routing algorithms. *Networks*, **7**(2), 113–148.
- Golden, B.L., Wasil, E.A., Kelly, J.P., & Chao, I.M. 1998. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets and computational results. *Pages 33–56 of: Crainic, T.G., & Laporte, G. (eds), Fleet management and logistics.* Kluwer.
- Gong, Qiang, & Batta, Rajan. 2007. Allocation and reallocation of ambulances to casualty clusters in a disaster relief operation. *IIE Transactions*, **39**(1), 27–39.
- Gounaris, Chrysanthos E., Repoussis, Panagiotis P., Tarantilis, Christos D., & Floudas, Christodoulos A. 2011. A Hybrid Branch-and-Cut Approach for the Capacitated Vehicle Routing Problem. *Pages 507 – 511 of: E.N. Pistikopoulos, M.C. Georgiadis, & Kokossis, A.C. (eds), 21st European Symposium on Computer Aided Process Engineering.* Computer Aided Chemical Engineering, vol. 29. Elsevier.
- Gouveia, Luis, & Voss, Stefan. 1995. A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research*, **83**(1), 69–82.
- Green III, Walter G., & McGinnis, Suzanne R. 2002. Thoughts on the Higher Order Taxonomy of Disasters. *Notes on the Science of Extreme Situations*, **7**, 1–6.
- Gribkovskaia, Irina, Gullberg, B. O., Hovden, Karl J., & Wallace, Stein W. 2006. Optimization model for a livestock collection problem. *International Journal of Physical Distribution & Logistics Management*, **36**(2), 136–152.
- Guerrero, W. J., Prodhon, C., Velasco, N., & Amaya, C. A. 2013. Hybrid heuristic for the inventory location-routing problem with deterministic demand. *International Journal of Production Economics*, **146**(1), 359–370.
- Gutiérrez-Jarpa, Gabriel, Desaulniers, Guy, Laporte, Gilbert, & Marianov, Vladimir. 2010. A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows. *European Journal of Operational Research*, **206**(2), 341 – 349.
- Hà, Minh Hoàng, Bostel, Nathalie, Langevin, André, & Rousseau, Louis-Martin. 2013. An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a

constraint on the number of vertices. *European Journal of Operational Research*, **226**(2), 211–220.

Hadjiconstantinou, Eleni, Christofides, Nicos, & Mingozzi, Aristide. 1995. A new exact algorithm for the vehicle routing problem based on q -paths and k -shortest paths relaxations. *Annals of Operations Research*, **61**(1), 21–43.

Hamedi, Masoud, Haghani, Ali, & Yang, Saini. 2012. Reliable Transportation of Humanitarian Supplies in Disaster Response: Model and Heuristic. *Procedia - Social and Behavioral Sciences*, **54**(0), 1205–1219. Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation, September 2012, Paris.

Hart, W.E., Krasnogor, N., & Smith, J.E. 2005. Memetic Evolutionary Algorithms. *Pages 3–27 of: Hart, William E., Smith, J.E., & Krasnogor, N. (eds), Recent Advances in Memetic Algorithms*. Studies in Fuzziness and Soft Computing, vol. 166. Springer Berlin Heidelberg.

Hashimoto, Hideki, Yagiura, Mutsunori, & Ibaraki, Toshihide. 2008. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*, **5**(2), 434–456.

Hauge, Kristian, Larsen, Jesper, Lusby, Richard Martin, & Krappner, Emil. 2014. A hybrid column generation approach for an industrial waste collection routing problem. *Computers & Industrial Engineering*, **71**(0), 10 – 20.

Hellingrath, Bernd, & Widera, Adam. 2011. Survey on major challenges in humanitarian logistics. *In: Proceedings of the th International ISCRAM Conference*.

Hemel, Tim, van Erk, Sebastiaan, & Jenniskens, Pieter. 1996. *The Manhattan Project*.

Hoetmer, G.J. 1991. Introduction. *Pages xvii–xxiv of: Drabek, T.E., Hoetmer G.J. (ed), Emergency Management: Principles and Practice for Local Government*. International City Management Association.

Holguín-Veras, José Brandão, Pérez, Noel, Jaller, Miguel, Wassenhove, Luk N. Van, & Aros-Vera, Felipe. 2013. On the appropriate objective function for post-disaster humanitarian logistics models. *Journal of Operations Management*, **31**(5), 262–280.

Hsueh, C., Chen, H., & Chou, H. 2008. *Vehicle routing problem*. Vol. 1. Chap. Dynamic vehicle routing for relief logistics in natural disasters, pages 71–84.

- Hu, Zhi-Hua. 2011. A container multimodal transportation scheduling approach based on immune affinity model for emergency relief. *Expert Systems with Applications*, **38**(3), 2632–2639.
- Huang, Michael, Smilowitz, Karen, & Balcik, Burcu. 2012. Models for relief routing: Equity, efficiency and efficacy. *Transportation Research Part E: Logistics and Transportation Review*, **48**(1), 2 – 18. Select Papers from the 19th International Symposium on Transportation and Traffic Theory.
- IFRC. 2011. *About disasters*. <http://www.ifrc.org/en/what-we-do/disaster-management/about-disasters/>: IFRC (International Federation of Red Cross and Red Crescent Societies).
- IFRC. 2013. *World disasters report 2013. Focus on technology and the future of humanitarian action*. IFRC (International Federation of Red Cross and Red Crescent Societies).
- Islam, K.M.S. 1998. *Spatial Dynamic Queueing Models for the Daily Deployment of Airtankers for Forest Fire Control*. Ph.D. thesis, University of Toronto.
- Jin, Mingzhou, Liu, Kai, & Eksioglu, Burak. 2008. A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, **36**(2), 265 – 270.
- Jothi, Raja, & Raghavachari, Balaji. 2004. Minimum Latency Tours and the k-Traveling Repairmen Problem. *Pages 423–433 of: Farach-Colton, Martín (ed), LATIN 2004: Theoretical Informatics*. Lecture Notes in Computer Science, vol. 2976. Springer Berlin Heidelberg.
- Jothi, Raja, & Raghavachari, Balaji. 2007. Approximating the k -traveling repairman problem with repair times. *Journal of Discrete Algorithms*, **5**(2), 293–303.
- Jotshi, Arun, Gong, Qiang, & Batta, Rajan. 2009. Dispatching and routing of emergency vehicles in disaster mitigation using data fusion. *Socio-Economic Planning Sciences*, **43**(1), 1–24.
- Jozefowiez, Nicolas, Semet, Frédéric, & Talbi, El-Ghazali. 2007. The bi-objective covering tour problem. *Computers & Operations Research*, **34**(7), 1929–1942.
- Kara, Imdat, Kara, Bahar Y., & Yetis, M. Kadri. 2007. Energy Minimizing Vehicle Routing Problem. *Pages 62–71 of: Dress, Andreas, Xu, Yinfeng, & Zhu, Binhai (eds), Combinatorial Optimization and Applications*. Lecture Notes in Computer Science, vol. 4616. Springer Berlin Heidelberg.

- Kara, Imdat, Kara, Bahar Yetis, & Yetis, M. Kadri. 2008. Cumulative vehicle routing problems. *Pages 85–98 of: Caric, Tonci, & Gold, Hrvoje (eds), Vehicle routing problem.* Intech.
- Karaoglan, Ismail, Altiparmak, Fulya, Kara, Imdat, & Dengiz, Berna. 2011. A branch and cut algorithm for the location-routing problem with simultaneous pickup and delivery. *European Journal of Operational Research*, **211**(2), 318 – 332.
- Ke, Liangjun, & Feng, Zuren. 2013. A two-phase metaheuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, **40**, 633–638.
- Kemball-Cook, David, & Stephenson, Robin. 1984. Lessons in logistics from Somalia. *Disasters*, **8**(1), 57–66.
- Krumke, Sven O., de Paepe, Willem E., Poensgen, Diana, & Stougie, Leen. 2003. News from the online traveling repairman. *Theoretical Computer Science*, **295**(1-3), 279–294.
- Kytöjoki, Jari, Nuortio, Teemu, Bräysy, Olli, & Gendreau, Michel. 2007. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research*, **34**(9), 2743–2757.
- Labadi, Nacima, Prins, Christian, & Reghioui, Mohamed. 2008a. An Evolutionary Algorithm with Distance Measure for the Split Delivery Capacitated Arc Routing Problem. *Pages 275–294 of: Cotta, Carlos, & van Hemert, Jano (eds), Recent Advances in Evolutionary Computation for Combinatorial Optimization.* Studies in Computational Intelligence, vol. 153. Springer Berlin Heidelberg.
- Labadi, Nacima, Prins, Christian, & Reghioui, Mohamed. 2008b. GRASP with Path Relinking for the Capacitated Arc Routing Problem with Time Windows. *Pages 111–135 of: Fink, Andreas, & Rothlauf, Franz (eds), Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management.* Studies in Computational Intelligence, vol. 144. Springer Berlin Heidelberg.
- Labadi, Nacima, Prins, Christian, & Reghioui, Mohamed. 2008c. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO - Operations Research*, **42**(3), 415–431.
- Lacomme, P., Prins, C., & Sevaux, M. 2006. A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers & Operations Research*, **33**(12), 3473–3493.

- Lacomme, Philippe, Prins, Christian, & Ramdane-Chérif, Wahiba. 2001. A Genetic Algorithm for the Capacitated Arc Routing Problem and Its Extensions. *Pages 473–483 of: Boers, Egbert J. W. (ed), Applications of Evolutionary Computing*. Lecture Notes in Computer Science, vol. 2037. Springer Berlin Heidelberg.
- Lacomme, Philippe, Prins, Christian, & Ramdane-Cherif, Wahiba. 2004. Competitive Memetic Algorithms for Arc Routing Problems. *Annals of Operations Research*, **131**(1-4), 159–185.
- Lacomme, Philippe, Prins, Christian, & Ramdane-Chérif, Wahiba. 2005. Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research*, **165**(2), 535–553.
- Lagaros, Nikos D., & Karlaftis, Matthew G. 2011. A critical assessment of metaheuristics for scheduling emergency infrastructure inspections. *Swarm and Evolutionary Computation*, **1**(3), 147–163.
- Landesman, Linda Young. 1996. *Emergency preparedness in health care organizations*. Oakbrook Terrace, IL : Joint Commission on Accreditation of Healthcare Organizations.
- Landesman, Linda Young. 2001. *Public Health Management of Disasters: The Practice Guide*. APHA Press.
- Laporte, G. and Semet, F. 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications. Chap. Classical heuristics for the capacitated VRP, pages 109–128.
- Laporte, G. 2010. A Concise Guide to the Traveling Salesman Problem. *The Journal of the Operational Research Society*, **61**(1), 35–40.
- Laporte, G., & Nobert, Y. 1983. A branch and bound algorithm for the capacitated vehicle routing problem. *OR Spektrum*, **5**(2), 77–85.
- Laporte, Gilbert. 2009. Fifty Years of Vehicle Routing. *Transportation Science*, **43**(4), 408–416.
- Laporte, Gilbert, Nobert, Yves, & Desrochers, Martin. 1985. Optimal Routing under Capacity and Distance Restrictions. *Operations Research*, **33**(5), 1050–1073.
- Laporte, Gilbert, Mercure, Hélène, & Nobert, Yves. 1986. An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks*, **16**(1), 33–46.

- Laurent, Benoît, & Hao, Jin-Kao. 2009. Iterated local search for the multiple depot vehicle scheduling problem. *Computers & Industrial Engineering*, **57**(1), 277–286.
- Lenstra, J. K., & Kan, A. H. G. Rinnooy. 1981. Complexity of vehicle routing and scheduling problems. *Networks*, **11**(2), 221–227.
- Li, Feiyue, Golden, Bruce, & Wasil, Edward. 2005. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, **32**(5), 1165–1179.
- Li, L. Y. O., & Fu, Z. 2002. The School Bus Routing Problem: A Case Study. *Journal of the Operational Research Society*, **53**(5), 552–558.
- Li, Xueping, Zhao, Zhaoxia, Zhu, Xiaoyan, & Wyatt, Tami. 2011. Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research*, **74**, 281–310.
- Liefooghe, A., Jourdan, L., & Talbi, E.-G. 2010. Metaheuristics and cooperative approaches for the Bi-objective Ring Star Problem. *Computers & Operations Research*, **37**(6), 1033–1044.
- Lin, S., & Kernighan, B. W. 1973. An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research*, **21**(2), 498–516.
- Lin, Shen. 1965. Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, **44**(10), 2245–2269.
- Lin, Yen-Hung, Batta, Rajan, Rogerson, Peter A., Blatt, Alan, & Flanigan, Marie. 2011. A logistics model for emergency supply of critical items in the aftermath of a disaster. *Socio-Economic Planning Sciences*, **45**(4), 132–145.
- Lin, Yen-Hung, Batta, Rajan, Rogerson, Peter A., Blatt, Alan, & Flanigan, Marie. 2012. Location of temporary depots to facilitate relief operations after an earthquake. *Socio-Economic Planning Sciences*, **46**(2), 112–123.
- Lourenço, H.R., Martin, O.C., & Stützle, T. 2010. Iterated local search: framework and applications. *Pages 363–397 of: Gendreau, M., & Potvin, J.Y. (eds), Handbook of metaheuristics*. Springer.
- Lucena, A. 1990. Time-dependent traveling salesman problem: The deliveryman case. *Networks*, **20**, 753–763.

- Luo, Zhixing, Qin, Hu, & Lim, Andrew. 2014. Branch-and-price-and-cut for the multiple traveling repairman problem with distance constraints. *European Journal of Operational Research*, **234**(1), 49 – 60.
- Lysgaard, Jens, & Wøhlk, Sanne. 2014. A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *European Journal of Operational Research*, **236**(3), 800–810. Vehicle Routing and Distribution Logistics.
- Lysgaard, Jens, Letchford, Adam N., & Eglese, Richard W. 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, **100**(2), 423–445.
- Mak, Vicky, & Boland, Natashia. 2000. Heuristic approaches to the asymmetric travelling salesman problem with replenishment arcs. *International Transactions in Operational Research*, **7**(4-5), 431–447.
- Malandraki, Chryssi, & Dial, Robert B. 1996. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operational Research*, **90**(1), 45 – 55.
- Marianov, V., & ReVelle, C.S. 1995. *Facility Location. A Survey of Applications and Methods*. Springer. Chap. Siting Emergency Services, pages 199–223.
- Martinelli, Rafael, Pecin, Diego, Poggi, Marcus, & Longo, Humberto. 2011. A Branch-Cut-and-Price Algorithm for the Capacitated Arc Routing Problem. *Pages 315–326 of: Pardalos, Panos M., & Rebennack, Steffen (eds), Experimental Algorithms. Lecture Notes in Computer Science*, vol. 6630. Springer Berlin Heidelberg.
- Maya, Pablo, & Sörensen, Kenneth. 2011. A GRASP metaheuristic to improve accessibility after a disaster. *OR Spectrum*, **33**, 525–542.
- Mendoza, Jorge E., Castanier, Bruno, Guéret, Christelle, Medaglia, Andrés L., & Velasco, Nubia. 2010. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, **37**(11), 1886–1898.
- Mester, David, & Bräysy, Olli. 2005. Active guided evolution strategies for large-scale vehicle routing problems with time windows. *Computers & Operations Research*, **32**(6), 1593–1614.
- Mester, David, & Bräysy, Olli. 2007. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research*, **34**(10), 2964–2975.

- Michallet, Julien, Prins, Christian, Amodeo, Lionel, Yalaoui, Farouk, & Vitry, Grégoire. 2014. Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & Operations Research*, **41**(0), 196–207.
- Miller, C. E., Tucker, A. W., & Zemlin, R. A. 1960. Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*, **7**(4), 326–329.
- Minieka, Edward. 1989. The delivery man problem on a tree network. *Annals of Operations Research*, **18**(1), 261–266.
- Mladenović, N., & Hansen, P. 1997. Variable neighborhood search. *Computers & Operations Research*, **24**(11), 1097–1100.
- Mohaymany, Afshin Shariat, Hosseini, Mahmood, & Habibi, Hossein Motevalli. 2003. Obtaining the emergency transportation network for rescue and relief activities in large cities based on the life loss mitigation criteria. *Technical Council on Lifeline Earthquake Engineering Monograph*, **25**, 231–240.
- Naddef, D., & Rinaldi, G. 2002. *The Vehicle Routing Problem*. Siam Monographs on Discrete Mathematics and Applications. Chap. Branch-and-cut algorithms for the capacitated VRP, pages 53–81.
- Nagata, Yuichi. 2007. Edge Assembly Crossover for the Capacitated Vehicle Routing Problem. *Pages 142–153 of: Cotta, Carlos, & Hemert, Jano (eds), Evolutionary Computation in Combinatorial Optimization*. Lecture Notes in Computer Science, vol. 4446. Springer Berlin Heidelberg.
- Naji-Azimi, Z., Renaud, J., Ruiz, A., & Salari, M. 2012a. A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. *European Journal of Operational Research*, **222**(3), 596–605.
- Naji-Azimi, Zahra, & Salari, Majid. 2014. The time constrained maximal covering salesman problem. *Applied Mathematical Modelling*, –.
- Naji-Azimi, Zahra, Salari, Majid, & Toth, Paolo. 2010. A heuristic procedure for the Capacitated m -Ring-Star problem. *European Journal of Operational Research*, **207**(3), 1227–1234.
- Naji-Azimi, Zahra, Salari, Majid, & Toth, Paolo. 2012b. An Integer Linear Programming based heuristic for the Capacitated m -Ring-Star Problem. *European Journal of Operational Research*, **217**(1), 17–25.

- Nelson, Marvin D., Nygard, Kendall E., Griffin, John H., & Shreve, Warren E. 1985. Implementation techniques for the vehicle routing problem. *Computers & Operations Research*, **12**(3), 273–283.
- Neri, Ferrante, & Cotta, Carlos. 2012. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, **2**(0), 1–14.
- Ngueveu, Sandra Ulrich, Prins, Christian, & Wolfler-Calvo, Roberto. 2009. An Effective Evolutionary Algorithm for the Cumulative Capacitated Vehicle Routing Problem. *Pages 778–787 of: Giacobini, Mario, Brabazon, Anthony, Cagnoni, Stefano, Di Caro, Gianni A., Ekárt, Anikó, Esparcia-Alcázar, Anna Isabel, Farooq, Muddassar, Fink, Andreas, Machado, Penousal, McCormack, Jon, O’Neill, Michael, Neri, Ferrante, Preuss, Mike, Rothlauf, Franz, Tarantino, Ernesto, & Yang, Shengxiang (eds), Applications of Evolutionary Computing. Lecture Notes in Computer Science, vol. 5484. Springer Berlin Heidelberg.*
- Ngueveu, Sandra Ulrich, Prins, Christian, & Calvo, Roberto Wolfler. 2010. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, **37**(11), 1877–1885.
- Nguyen, Viet-Phuong, Prins, Christian, & Prodhon, Caroline. 2010. A Multi-Start Evolutionary Local Search for the Two-Echelon Location Routing Problem. *Pages 88–102 of: Blesa, María J., Blum, Christian, Raidl, Günther, Roli, Andrea, & Sampels, Michael (eds), Hybrid Metaheuristics. Lecture Notes in Computer Science, vol. 6373. Springer Berlin Heidelberg.*
- Nguyen, Viet-Phuong, Prins, Christian, & Prodhon, Caroline. 2012a. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, **25**(1), 56–71.
- Nguyen, Viet-Phuong, Prins, Christian, & Prodhon, Caroline. 2012b. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. *European Journal of Operational Research*, **216**(1), 113–126.
- Nishi, Tatsushi, & Izuno, Tsukasa. 2014. Column generation heuristics for ship routing and scheduling problems in crude oil transportation with split deliveries. *Computers & Chemical Engineering*, **60**(0), 329 – 338.
- Olivera, Alfredo, & Viera, Omar. 2007. Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research*, **34**(1), 28–47.

- Oppen, Johan, Løkketangen, Arne, & Desrosiers, Jacques. 2010. Solving a rich vehicle routing and inventory problem using column generation. *Computers & Operations Research*, **37**(7), 1308 – 1317. Algorithmic and Computational Methods in Retrial Queues.
- Or, I. 1976. *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking*. Ph.D. thesis, Department of Industrial Engineering and Management Sciences, Northwestern University.
- Özdamar, Linet. 2011. Planning helicopter logistics in disaster relief. *OR Spectrum*, **33**, 655–672.
- Özdamar, Linet, Ekinçi, Ediz, & Küçükyazıcı, Beste. 2004. Emergency Logistics Planning in Natural Disasters. *Annals of Operations Research*, **129**, 217–245.
- Ozsoydan, Fehmi Burcin, & Sipahioglu, Aydin. 2013. Heuristic solution approaches for the cumulative capacitated vehicle routing problem. *Optimization*, **62**(10), 1321–1340.
- Paessens, H. 1988. The savings algorithm for the vehicle routing problem. *European Journal of Operational Research*, **34**(3), 336–344.
- Palhazi Cuervo, Daniel, Goos, Peter, Sörensen, Kenneth, & Arráiz, Emely. 2014. An iterated local search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, **237**(2), 454–464.
- Park, Junhyuk, & Kim, Byung-In. 2010. The school bus routing problem: A review. *European Journal of Operational Research*, **202**(2), 311–319.
- Penna, Puca Huachi Vaz, Subramanian, Anand, & Ochi, Luiz Satoru. 2013. An Iterated Local Search heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *Journal of Heuristics*, **19**(2), 201–232.
- Petch, R. J, & Salhi, S. 2004. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics*, **133**, 69–92.
- Pettit, S., & Beresford, A. 2005. Emergency relief logistics: an evaluation of military, non-military and composite response models. *International Journal of Logistics*, **8**(4), 313–331.
- Picard, Jean-Claude, & Queyranne, Maurice. 1978. The Time-Dependent Traveling Salesman Problem and Its Application to the Tardiness Problem in One-Machine Scheduling. *Operations Research*, **26**(1), 86–110.

- Pisinger, David, & Ropke, Stefan. 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research*, **34**(8), 2403–2435.
- Pop, Petrică C., Kara, Imdat, & Marc, Andrei Horvat. 2012. New mathematical models of the generalized vehicle routing problem and extensions. *Applied Mathematical Modelling*, **36**(1), 97–107.
- Pop, Petrică C., Matei, Oliviu, & Sitar, Corina Pop. 2013. An improved hybrid algorithm for solving the generalized vehicle routing problem. *Neurocomputing*, **109**(0), 76–83.
- Prins, Christian. 2002. Efficient heuristic for the heterogeneous fleet multitrip VRP with application to a large-scale real case. *Journal of Mathematical Modelling and Algorithms*, **1**, 135–150.
- Prins, Christian. 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, **31**(12), 1985–2002.
- Prins, Christian. 2009a. A GRASP \times evolutionary local search hybrid for the vehicle routing problem. *Pages 35–53 of: Pereira, F.B., Tavares J. (ed), Bio-inspired Algorithms for the Vehicle Routing Problem*. Springer.
- Prins, Christian. 2009b. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, **22**(6), 916–928.
- Prins, Christian, Lacomme, Philippe, & Prodhon, Caroline. 2014. Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies*, **40**(0), 179–200.
- Prodhon, Caroline, & Prins, Christian. 2014. A survey of recent research on location-routing problems. *European Journal of Operational Research*, **238**(1), 1–17.
- Psaraftis, Harilaos N., & Ziozas, Babis O. 1985. A Tactical Decision Algorithm for the Optimal Dispatching of Oil Spill Cleanup Equipment. *Management Science*, **31**(12), 1475–1491.
- Psaraftis, Harilaos N., Tharakan, Geverghese G., & Ceder, Avishai. 1986. Optimal Response to Oil Spills: The Strategic Decision Case. *Operations Research*, **34**(2), 203–217.
- Reghioui, Mohamed, Prins, Christian, & Labadi, Nacima. 2007. GRASP with Path Relinking for the Capacitated Arc Routing Problem with Time Windows. *Pages 722–731 of: Giacobini, Mario (ed), Applications of Evolutionary Computing*. Lecture Notes in Computer Science, vol. 4448. Springer Berlin Heidelberg.

- Reimann, Marc, Doerner, Karl, & Hartl, Richard F. 2004. D-Ants: Savings Based Ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, **31**(4), 563–591.
- Renaud, Jacques, Boctor, Fayed F., & Laporte, Gilbert. 1996a. A Fast Composite Heuristic for the Symmetric Traveling Salesman Problem. *INFORMS Journal on Computing*, **8**(2), 134–143.
- Renaud, Jacques, Boctor, Fayed F., & Laporte, Gilbert. 1996b. An Improved Petal Heuristic for the Vehicle Routing Problem. *The Journal of the Operational Research Society*, **47**(2), 329–336.
- Repoussis, P. P., Tarantilis, C. D., Bräysy, O., & Ioannou, G. 2010. A hybrid evolution strategy for the open vehicle routing problem. *Computers & Operations Research*, **37**(3), 443–455.
- Ribeiro, Glaydston Mattos, & Laporte, Gilbert. 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, **39**(3), 728–735.
- Riera-Ledesma, Jorge, & Salazar-González, Juan-José. 2012. Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Computers & Operations Research*, **39**(2), 391 – 404.
- ROADEF. 2014. *ROADEF: Société Française de Recherche Opérationnelle et d'Aide à la Décision*. <http://www.roadef.org/>: ROADEF (Société Française de Recherche Opérationnelle et d'Aide à la Décision).
- Rochat, Yves, & Taillard, Éric D. 1995. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, **1**(1), 147–167.
- Rodríguez-Martín, Inmaculada, Salazar-González, Juan-José, & Yaman, Hande. 2014. A branch-and-cut algorithm for the hub location and routing problem. *Computers & Operations Research*, **50**(0), 161 – 174.
- Ronald, S. 1998. More distance functions for order-based encodings. *Pages 558–563 of: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*.
- Ropke, Stefan, & Pisinger, David. 2006. A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research*, **171**(3), 750–775.

Ryan, David M., Hjorring, Curt, & Glover, Fred. 1993. Extensions of the Petal Method for Vehicle Routing. *The Journal of the Operational Research Society*, **44**(3), 289–296.

Salani, Matteo, & Vacca, Ilaria. 2011. Branch and price for the vehicle routing problem with discrete split deliveries and time windows. *European Journal of Operational Research*, **213**(3), 470 – 477.

Salari, Majid, & Naji-Azimi, Zahra. 2012. An integer programming-based local search for the covering salesman problem. *Computers & Operations Research*, **39**(11), 2594–2602.

Salari, Majid, Toth, Paolo, & Tramontani, Andrea. 2010a. An ILP improvement procedure for the Open Vehicle Routing Problem. *Computers & Operations Research*, **37**(12), 2106–2120.

Salari, Majid, Naji-Azimi, Zahra, & Toth, Paolo. 2010b. A Variable Neighborhood Search and its Application to a Ring Star Problem Generalization. *Electronic Notes in Discrete Mathematics*, **36**(0), 343–350.

Salehipour, A., Sörensen, K., Goos, P., & Bräysy, O. 2008. *An efficient GRASP + VND metaheuristic for the traveling repairman problem*. Working paper. University of Antwerp, Faculty of Applied Economics.

Salhi, S., & Petch, R. 2007. A GA-Based Heuristic for the Vehicle Routing Problem with Multiple Trips. *Journal of Mathematical Modelling and Algorithms*, **6**, 591–613.

Salhi, Said, Wassan, Niaz, & Hajarat, Mutaz. 2013. The Fleet Size and Mix Vehicle Routing Problem with Backhauls: Formulation and Set Partitioning-based Heuristics. *Transportation Research Part E: Logistics and Transportation Review*, **56**(0), 22–35.

Santos, Fernando Afonso, Mateus, Geraldo Robson, & da Cunha, Alexandre Salles. 2011. A Branch-and-price algorithm for a Vehicle Routing Problem with Cross-Docking. *Electronic Notes in Discrete Mathematics*, **37**(0), 249 – 254. LAGOS'11 - VI Latin-American Algorithms, Graphs and Optimization Symposium.

Schilling, D., Jayaraman, V., & Barkhi, R. 1993. A review of covering problems in facility location. *Location Science*, **1**(1), 25–55.

Schittekat, Patrick, Kinable, Joris, Sörensen, Kenneth, Sevaux, Marc, Spieksma, Frits, & Springael, Johan. 2013. A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, **229**(2), 518–528.

- Scott, Peter D., & Rogova, Galina L. 2004. Crisis Management in a Data Fusion Synthetic Task Environment. *Pages 330–337 of: 7th International Conference on Information Fusion.*
- Sevaux, M., & Sörensen, K. 2005. Permutation distance measures for memetic algorithms with population management. *Pages 832–838 of: Proceedings of 6th Metaheuristics International Conference, MIC 2005.*
- Shaw, Paul. 1998. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. *Pages 417–431 of: Maher, Michael, & Puget, Jean-Francois (eds), Principles and Practice of Constraint Programming – CP98. Lecture Notes in Computer Science, vol. 1520. Springer Berlin Heidelberg.*
- Sheu, Jiuh-Biing. 2007. An emergency logistics distribution approach for quick response to urgent relief demand in disasters. *Transportation Research Part E: Logistics and Transportation Review*, **43**(6), 687–709.
- Silva, Marcos Melo, Subramanian, Anand, Vidal, Thibaut, & Ochi, Luiz Satoru. 2012. A simple and effective metaheuristic for the Minimum Latency Problem. *European Journal of Operational Research*, **221**(3), 513–520.
- Simchi-Levi, David, & Berman, Oded. 1991. Minimizing the Total Flow Time of n Jobs on a Network. *IIE Transactions*, **23**(3), 236–244.
- Simonetti, L., Frota, Y., & de Souza, C.C. 2011. The ring-star problem: A new integer programming formulation and a branch-and-cut algorithm. *Discrete Applied Mathematics*, **159**(16), 1901–1914.
- Sniedovich, Moshe, & Voß, Stefan. 2006. The corridor method: a dynamic programming inspired metaheuristic. *Control and Cybernetics*, **35**(3), 551–578.
- Sörensen, Kenneth. 2007. Distance measures based on the edit distance for permutation-type representations. *Journal of Heuristics*, **13**(1), 35–47.
- Sörensen, Kenneth, & Sevaux, Marc. 2006. MA|PM: memetic algorithms with population management. *Computers & Operations Research*, **33**(5), 1214–1225.
- Stepanov, Alexander, & Smith, James MacGregor. 2009. Multi-objective evacuation routing in transportation networks. *European Journal of Operational Research*, **198**(2), 435–446.
- Stålhane, Magnus, Andersson, Henrik, Christiansen, Marielle, Cordeau, Jean-François, & Desaulniers, Guy. 2012. A branch-price-and-cut method for a ship routing and scheduling problem with split loads. *Computers & Operations Research*, **39**(12), 3361 – 3375.

- Swersey, Arthur J. 1994. The deployment of police, fire, and emergency medical units. *Pages 151–200 of: S.M. Pollock, M.H. Rothkopf, & Barnett, A. (eds), Operations Research and the Public Sector. Handbooks in Operations Research and Management Science, vol. 6. Elsevier.*
- Taş, D., Gendreau, M., Dellaert, N., van Woensel, T., & de Kok, A.G. 2014. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, **236**(3), 789 – 799.
- Taillard, É. 1993. Parallel iterative search methods for vehicle routing problems. *Networks*, **23**(8), 661–673.
- Taillard, Éric D., Laporte, Gilbert, & Gendreau, Michel. 1996. Vehicle Routing with Multiple Use of Vehicles. *Journal of the Operational Research Society*, **47**(8), 1065–1070.
- Talbi, El-Ghazli. 2009. *Metaheuristics. From design to implementation.* Wiley. Chap. Common Concepts for Metaheuristics, pages 1–86.
- Tarantilis, C. D. 2005. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research*, **32**(9), 2309–2327.
- Tarantilis, C. D., & Kiranoudis, C. T. 2002. BoneRoute: An Adaptive Memory-Based Method for Effective Fleet Management. *Annals of Operations Research*, **115**(1-4), 227–241.
- The Operational Research Society. 2014. *Operational Research. The science of better.* <http://www.scienceofbetter.co.uk/>: The Operational Research Society.
- Thomas, Anisya S., & Kopczak, Laura Rock. 2005. *From logistics to supply chain management: the path forward in the humanitarian sector.* Tech. rept. Fritz Institute.
- Thomas, Marlin U. 2002. Supply chain reliability for contingency Operations. *Pages 61–67 of: Proceedings of the annual reliability and maintainability symposium.*
- Thompson, Paul M., & Psaraftis, Harilaos N. 1993. Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems. *Operations Research*, **41**(5), 935–946.
- Tomasini, Rolando, & van Wassenhove, Luk. 2009. *Humanitarian Logistics.* Palgrave Macmillan.
- Toth, Paolo, & Tramontani, Andrea. 2008. An Integer Linear Programming Local Search for Capacitated Vehicle Routing Problems. *Pages 275–295 of: Golden, Bruce, Raghavan, S., &*

- Wasil, Edward (eds), *The Vehicle Routing Problem: Latest Advances and New Challenges*. Operations Research/Computer Science Interfaces, vol. 43. Springer US.
- Toth, Paolo, & Vigo, Daniele. 2003. The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. *INFORMS Journal on Computing*, **15**(4), 333–346.
- Tsitsiklis, John N. 1992. Special cases of traveling salesman and repairman problems with time windows. *Networks*, **22**, 263–282.
- Turkensteen, Marcel, Ghosh, Diptesh, Goldengorin, Boris, & Sierksma, Gerard. 2008. Tolerance-based Branch and Bound algorithms for the ATSP. *European Journal of Operational Research*, **189**(3), 775 – 788.
- Ukkonen, Esko. 1985. Finding approximate patterns in strings. *Journal of Algorithms*, **6**(1), 132–137.
- UNISDR. 2004. *Living with risk: a global review of disaster reduction initiatives*. UNISDR (United Nations Office for Disaster Risk Reduction).
- Valle, Cristiano Arbex, da Cunha, Alexandre Salles, Mateus, Geraldo Robson, & Martinez, Leonardo C. 2009. Exact algorithms for a selective Vehicle Routing Problem where the longest route is minimized. *Electronic Notes in Discrete Mathematics*, **35**(0), 133–138.
- Valle, Cristiano Arbex, Martinez, Leonardo Conegundes, da Cunha, Alexandre Salles, & Mateus, Geraldo R. 2011. Heuristic and exact algorithms for a min-max selective vehicle routing problem. *Computers & Operations Research*, **38**(7), 1054–1065.
- van Wassenhove, L. N. 2006. Humanitarian Aid Logistics: Supply Chain Management in High Gear. *The Journal of the Operational Research Society*, **57**(5), 475–489.
- Vander Wiel, Russ J., & Sahinidis, Nikolaos V. 1995. Heuristic Bounds and Test Problem Generation for the Time-Dependent Traveling Salesman Problem. *Transportation Science*, **29**(2), 167–183.
- Vander Wiel, Russ J., & Sahinidis, Nikolaos V. 1996. An exact solution approach for the time-dependent traveling-salesman problem. *Naval Research Logistics (NRL)*, **43**(6), 797–820.
- Vansteenwegen, Pieter, & Mateo, Manuel. 2014. An iterated local search algorithm for the single-vehicle cyclic inventory routing problem. *European Journal of Operational Research*, **237**(3), 802–813.

Vidal, Thibaut, Crainic, Teodor Gabriel, Gendreau, Michel, Lahrichi, Nadia, & Rei, Walter. 2012. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research*, **60**(3), 611–624.

Vidal, Thibaut, Crainic, Teodor Gabriel, Gendreau, Michel, & Prins, Christian. 2013. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, **40**(1), 475–489.

Vidal, Thibaut, Crainic, Teodor Gabriel, Gendreau, Michel, & Prins, Christian. 2014. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, **234**(3), 658–673.

Villegas, Juan G., Prins, Christian, Prodhon, Caroline, Medaglia, Andrés L., & Velasco, Nubia. 2010. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, **23**(5), 780–794.

Villegas, Juan G., Prins, Christian, Prodhon, Caroline, Medaglia, Andrés L., & Velasco, Nubia. 2011. A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, **38**(9), 1319–1334.

Villegas, Juan G., Prins, Christian, Prodhon, Caroline, Medaglia, Andrés L., & Velasco, Nubia. 2013. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, **230**(2), 231–244.

Walker, James D., Ochoa, Gabriela, Gendreau, Michel, & Burke, Edmund K. 2012. Vehicle Routing and Adaptive Iterated Local Search within the HyFlex Hyper-heuristic Framework. *Pages 265–276 of: Hamadi, Youssef, & Schoenauer, Marc (eds), Learning and Intelligent Optimization*. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Wark, Peter, & Holt, John. 1994. A Repeated Matching Heuristic for the Vehicle Routing Problem. *The Journal of the Operational Research Society*, **45**(10), 1156–1167.

Wolf, Steffen, & Merz, Peter. 2007. Evolutionary Local Search for the Super-Peer Selection Problem and the p-Hub Median Problem. *Pages 1–15 of: Bartz-Beielstein, Thomas, Blesa Aguilera, MaríaJosé, Blum, Christian, Naujoks, Boris, Roli, Andrea, Rudolph, Günter, & Sampels, Michael (eds), Hybrid Metaheuristics*. Lecture Notes in Computer Science, vol. 4771. Springer Berlin Heidelberg.

Yagiura, Mutsunori, & Ibaraki, Toshihide. 1996. The use of dynamic programming in genetic algorithms for permutation problems. *European Journal of Operational Research*, **92**(2), 387–401.

Yan, Liang, Jinsong, Bao, Xiaofeng, Hu, & Ye, Jin. 2009. A heuristic project scheduling approach for quick response to maritime disaster rescue. *International Journal of Project Management*, **27**(6), 620–628.

Yi, Wei, & Özdamar, Linet. 2007. A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research*, **179**(3), 1177–1193.

Zhang, Zizhen, Qin, Hu, Zhu, Wenbin, & Lim, Andrew. 2012. The single vehicle routing problem with toll-by-weight scheme: A branch-and-bound approach. *European Journal of Operational Research*, **220**(2), 295 – 304.

Juan Carlos RIVERA AGUDELO

Doctorat : Optimisation et Sûreté des Systèmes

Année 2014

Optimisation de la logistique dans des opérations en cas de catastrophes

Les problèmes de tournées de véhicules cumulatives avec capacité (CCVRP) sont étudiés dans cette thèse, où la minimisation de la somme des temps d'arrivée reflète mieux les objectifs stratégiques de la logistique humanitaire.

Dans le problème de multiples tournées d'un véhicule cumulatif avec capacité (mt-CCSVRP), un seul véhicule est disponible et il peut effectuer plusieurs voyages. Un algorithme du plus court chemin avec contrainte de ressources est proposé pour résoudre ce problème, dans lequel les tournées deviennent des nœuds et les sites sont des ressources. Le réseau est orienté et acyclique en raison des propriétés particulières du mt-CCSVRP.

Le problème de multiples tournées de véhicules cumulatives avec capacité (mt-CCVRP) est introduit, où plusieurs véhicules peuvent effectuer multiples voyages. Quatre programmes linéaires en nombre entiers (PLNE) sont proposés pour résoudre le CCVRP. Un PLNE pour le mt-CCVRP est proposé ainsi que trois métaheuristiques : une recherche locale itérée à démarrages multiples (MS-ILS), un algorithme mémétique avec gestion de la population (MA|PM) et une recherche locale évolutive à démarrages multiples (MS-ELS), qui appellent un algorithme de recherche local à voisinages variables (VND). Une méthode split à deux phases permet MA|PM et MS-ELS d'alterner entre deux espaces de solutions.

Le problème de tournées de véhicules cumulatif avec capacité et des livraisons indirectes (CCVRP-ID) permet aux sites non visités si leurs demandes sont fournies par un véhicule auxiliaire. Un PLNE et un MS-ELS sont développés.

Mots clés : recherche opérationnelle – optimisation combinatoire – métaheuristiques – transport – aide humanitaire – logistique (organisation).

Logistic Optimization in Disaster Response Operations

The cumulative capacitated vehicle routing problems (CCVRP) are studied in this thesis, where the minimization of the sum of arrival times better reflects the strategic objectives of humanitarian logistics.

In the multitrip cumulative capacitated single-vehicle routing problem (mt-CCSVRP), only one vehicle is available and it can perform multiple trips. An exact resource constrained shortest path algorithm is proposed for this problem, in which trips become nodes and sites are resources. The resulting network is proven to be directed and acyclic due to the special properties of the mt-CCSVRP.

The multitrip cumulative capacitated vehicle routing problem (mt-CCVRP) is introduced, where several vehicles can do multiple trips. Four mixed integer linear programs (MILP) are proposed to solve the CCVRP. For the mt-CCVRP an MILP is also given as well as three metaheuristics: a multi-start iterated local search (MS-ILS), a memetic algorithm with population management (MA|PM) and a multi-start evolutionary local search (MS-ELS), which call a variable neighborhood descent algorithm (VND). A two phases split method allows MA|PM and MS-ELS to alternate between two solution spaces.

The cumulative capacitated vehicle routing problem with indirect deliveries (CCVRP-ID) allows unvisited sites if their demands are provided by an auxiliary vehicle. An MILP and an MS-ELS are developed.

Keywords: operations research – combinatorial optimization – metaheuristics – transportation – humanitarian assistance – business logistics.