



HAL
open science

Optimization of reverse supply chain with take into account the disassembly process

Muhammad Khoirul Khakim Habibi

► **To cite this version:**

Muhammad Khoirul Khakim Habibi. Optimization of reverse supply chain with take into account the disassembly process. Business administration. Université de Lyon, 2017. English. NNT : 2017LY-SEM005 . tel-02956465

HAL Id: tel-02956465

<https://theses.hal.science/tel-02956465v1>

Submitted on 2 Oct 2020

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.



N° d'ordre NNT : 2017LYSEM005

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'École des Mines de Saint-Etienne

Ecole Doctorale N° 488
Sciences, Ingénierie, Santé

Spécialité de doctorat : Génie Industriel

Soutenue publiquement/à huis clos le 10/02/2017, par :
Muhammad Khoirul Khakim HABIBI

**Optimisation intégrée de la collecte de
produits en fin de vie et de leur
désassemblage dans une chaîne
logistique inverse**

Devant le jury composé de :

AMODEO, Lionel	Professeur	Université de Technologie de Troyes	Président
PRINS, Christian	Professeur	Université de Technologie de Troyes	Rapporteur
REKIK, Yacine	Professeur associé	EMLyon Business School	Rapporteur
DAUZERE-PERES, Stéphane	Professeur	École des Mines de Saint-Etienne	Examineur
DOLGUI, Alexandre	Professeur	Institut Mines-Télécom Atlantique, Nantes	Directeur de thèse
CUNG, Van-Dat	Professeur	Université Grenoble Alpes	Co-directeur de thèse
BATTAIA, OLGA	Professeur	ISAÉ-SUPAERO Toulouse	Co-encadrante

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX MECANIQUE ET INGENIERIE GENIE DES PROCEDES SCIENCES DE LA TERRE SCIENCES ET GENIE DE L'ENVIRONNEMENT	K. Wolski Directeur de recherche S. Drapier, professeur F. Gruy, Maître de recherche B. Guy, Directeur de recherche D. Graillot, Directeur de recherche	MATHEMATIQUES APPLIQUEES INFORMATIQUE IMAGE, VISION, SIGNAL GENIE INDUSTRIEL MICROELECTRONIQUE	O. Roustant, Maître-assistant O. Boissier, Professeur J.C. Pinoli, Professeur X. Delorme, Maître assistant Ph. Lalevée, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	CR	Génie industriel	CMP
AUGUSTO	Vincent	CR	Image, Vision, Signal	CIS
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BADEL	Pierre	MA(MDC)	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BEIGBEDER	Michel	MA(MDC)	Informatique	FAYOL
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	MA(MDC)	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR1	Génie Industriel	FAYOL
CHRISTIE	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
DOUCE	Sandrine	PR2	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FOREST	Valérie	MA(MDC)	Génie des Procédés	CIS
FOURNIER	Jacques	Ingénieur chercheur CEA	Microélectronique	CMP
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Sciences de la Terre	SPIN
GAVET	Yann	MA(MDC)	Image Vision Signal	CIS
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFORST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche	Mécanique et ingénierie	CMP
NORTIER	Patrice	PR1		SPIN
OWENS	Rosin	MA(MDC)	Microélectronique	CMP
PERES	Véronique	MR	Génie des Procédés	SPIN
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROBISSON	Bruno	Ingénieur de recherche	Microélectronique	CMP
ROUSSY	Agnès	MA(MDC)	Microélectronique	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

INTEGRATED OPTIMISATION OF THE COLLECTION OF END-OF-LIFE
PRODUCTS AND THEIR DISASSEMBLY IN REVERSE SUPPLY CHAIN

Muhammad Khoirul Khakim HABIBI

Doctoral Dissertation of the Université de Lyon operated by
the École Nationale Supérieure des Mines de Saint-Étienne

2017

*To my Parents
To my Wife
To my Family*

Résumé

Contexte et motivations

De nos jours, nous constatons une évolution du cadre économique des entreprises de l'économie linéaire vers l'économie circulaire. L'économie linéaire est ainsi appelée à cause du caractère linéaire du cycle de vie du produit : les matières premières sont transformées en produits finaux qui sont vendus et acheminés aux clients et enfin sont jetés en fin de vie. Le rapport du forum économique mondial ([Forum, 2014](#)) a exposé l'obsolescence de cette économie. De ce fait, la mise en œuvre de l'économie circulaire est encouragée et soutenue par des gouvernements, à cause de ses avantages en terme financiers, sociaux et environnementaux sous le nom de triple performance (*angl. Triple-bottom-lines*). Cette mise en œuvre est effectuée en intégrant les activités des chaînes logistiques inverses telles que la réutilisation, la réparation, le réemploi et le recyclage, aux chaînes logistiques directes existantes. À cause de la complexité posée par les chaînes logistiques inverses, généralement les entreprises collaborent avec des entreprises spécialisées dans la gestion des chaînes logistiques inverses comme la société ENVIE en France.

Dans cette thèse, nous nous intéressons aux processus de collecte des produits en fin de vie ainsi qu'à leur désassemblage dans les chaînes logistiques inverses, car ils possèdent des rôles pivots. Cependant, les deux processus sont actuellement optimisés séparément. Ceci peut mener à des décisions sous-optimales pour les chaînes logistiques. Au contraire, l'intégration des décisions d'au moins deux processus dans les chaînes logistiques directes a été prouvée comme menant à une meilleure gestion de stock, une meilleure agilité à répondre aux demandes des clients et une augmentation de l'efficacité des chaînes. Dans cette thèse, nous souhaitons montrer l'intérêt d'une approche intégrée des deux processus (collecte et désassemblage) par rapport à une approche non-intégrée.

Cette thèse soutient non seulement la mise en œuvre de l'économie circulaire mais propose aussi une meilleure façon de gérer des chaînes logistiques inverses. Le manuscrit s'articule de la manière suivante :

- Le chapitre 1 présente l'introduction générale et l'état de l'art.
- Le chapitre 2 présente une comparaison entre deux formulations déterministes sous forme programmation linéaire en nombre entier : la première intégrant les deux

processus (collecte et désassemblage) et la seconde ne les intégrant pas. Un solveur commercial CPLEX est utilisé pour résoudre les problèmes correspondants.

- Le chapitre 3 présente plusieurs méthodes approchées développées pour traiter les instances de grande taille du problème intégré, puisque le solveur CPLEX n'est pas capable de fournir les solutions optimales en un temps de calcul raisonnable.
- Le chapitre 4 décrit une formulation basée sur la programmation stochastique du problème. Elle vise à considérer des paramètres sous incertitude qui existent fréquemment dans les chaînes logistiques inverses. Deux méthodes de résolution sont développées.
- Le chapitre 5 propose un problème qui se focalise sur les décisions de la collecte des produits en fin de vie et l'équilibrage de ligne de désassemblage. Une formulation déterministe du problème est proposée.
- Nous présentons également les conclusions et les perspectives de cette thèse dans le chapitre 6.

Les détails de ces chapitres sont présentés dans les prochaines sections.

Le problème de collecte et désassemblage

Ce chapitre est consacré à prouver que l'intégration des décisions des processus de collecte de produits en fin de vie et de leur désassemblage dans les chaînes logistiques inverses mène à une amélioration des décisions prises. Dans le problème présenté dans la section 2.2, nous considérons un seul type de produits en fin de vie, ramassés dans les centres de collecte dans une quantité déterminée, un seul véhicule disponible pour leur ramassage et il y a un dépôt pour les produits collectés où les processus de désassemblage sont effectués (un centre / site de désassemblage). Les variables de décision considérées sont (i) la quantité de produits en fin de vie à désassembler pour chaque période, (ii) les périodes quand le véhicule visite les centres de collecte, (iii) la quantité de produits en fin de vie à collecter depuis les centres de collecte, (iv) la tournée du véhicule pour chaque période et (v) le niveau de stockage des produits en fin de vie au dépôt.

Une formulation du problème qui intègre des décisions concernant les deux processus est développée dans la sous-section 2.3.1. Sa fonction objectif vise à minimiser le coût total tenant compte du coût de collecte, du coût de stock, du coût de désassemblage et de la pénalité des demandes de composant non-satisfaites. Le coût de collecte est constitué des coûts de setup et de voyage de véhicule. Le coût de stock concerne le niveau de stockage de produits en fin de vie. Le coût de désassemblage est en rapport à la quantité de produits en fin de vie à désassembler et à la pénalité due aux demandes de composant non-satisfaites.

Les contraintes dans cette formulation sont (i) les contraintes imposant que chaque centre de collecte soit visité au maximum une fois pour chaque période, (ii) les contraintes

de la conservation des flux de véhicule à chaque sommet, (iii) les contraintes d'élimination de sous-tours, (iv) les contraintes de la conservation des flux de produits en fin de vie dans le stock, (v) les contraintes déterminant les demandes non-satisfaites et (vi) l'ensemble des contraintes fixant les domaines des variables de décision à travers l'horizon.

La sous-section 2.3.2 présente une autre formulation du problème dans le cas non-intégré. Ne disposant pas d'instances appropriées pour tester les deux formulations, la section 2.4 présente la procédure de génération d'instances pour les tests. Les deux formulations sont comparées grâce au solveur commercial CPLEX. Selon l'analyse numérique présentée dans la section 2.5, les résultats obtenus vérifient notre hypothèse qu'une telle intégration mène à de meilleures décisions par rapport au cas non-intégré.

Ces travaux ont aboutis aux articles suivants :

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Collection-Disassembly Problem in Reverse Supply Chain, *International Journal of Production Economics*, vol. 183, 2017, p. 334-344. doi:10.1016/j.ijpe.2016.06.025
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Coordination of Collection and Disassembly Planning for End-of-Life Product. *IFAC-PapersOnLine*, vol. 48, no. 3, 2015, p. 76–80, doi:10.1016/j.ifacol.2015.06.061
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Dealing with the collection and the disassembly planning for simple end-of-life product, *Actes des Journées Nationales/ Doctorales MACS*, Du 16 au 19 Juin 2015, Bourges, France, 5 pages (USB)
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Combining Procurement and Disassembly Decisions for End-of-Life Product. Le 16ème congrès de la société française de recherche opérationnelle et aide à la décision (ROADEF). Du 25 au 26 Février 2015, Marseille, France
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. La coordination de la collecte des produits en fin de vie et de leur désassemblage dans les chaînes logistiques inversées. La 20ème journée du pôle Sciences et Techniques de la Production (STP) du GdR MaCS, Le 5 Février 2015, Troyes, France.

Le solveur commercial CPLEX nous permet de valider notre modèle ainsi que prouver notre hypothèse. Cependant, il n'est pas capable de fournir les solutions optimales en un temps raisonnable, notamment pour les instances de grande taille. Afin de trouver des solutions de bonne qualité en un temps de calcul raisonnable, nous avons développé plusieurs méthodes approchées, présentées dans le chapitre suivant.

Méthodes approchées pour le problème de collect et désassemblage

Six méthodes sont développées dans ce chapitre : la *Two-Phase Iterative Heuristic* (sous-

section 3.2.1), sa version avec des améliorations (sous-section 3.2.1), une heuristique basée sur la recherche locale (sous-section 3.2.2), une méthode d'optimisation par essais (sous-section 3.2.3), une méthode d'optimisation par essais particulières avec recherche locale (sous-section 3.2.3) et un *Imperialist Competitive Algorithm* (sous-section 3.2.4). Elles sont en fait basées sur une procédure itérative et des métaheuristiques.

La première méthode approchée décompose le problème en deux sous-problèmes: le problème de dimensionnement de lot de produits en fin de vie avec les coûts approximatifs de transport et le problème de voyageur de commerce. Les coûts approximatifs de transport sont obtenus en résolvant le problème de transport et appliquant l'heuristique de LKH de [Lin and Kernighan \(1973\)](#).

La deuxième méthode approchée est une amélioration de la première en ajoutant une étape supplémentaire. Elle vise à réduire le nombre de périodes servies et s'imposer une procédure adaptative afin d'améliorer les solutions et de réduire le temps de calcul sans sacrifier la qualité des solutions obtenues.

Dans la troisième méthode approchée, des opérateurs de recherche locale sont ajoutés pour améliorer la solution initiale.

Afin d'améliorer la solution initiale, la quatrième méthode approchée imite le comportement social des organismes sociaux. Dans cette méthode, le comportement relationnel d'un organisme avec les autres organismes est influencé par son propre comportement et celui des autres organismes du groupe auquel il appartient.

Nous avons également proposé une méthode basée sur la quatrième méthode en ajoutant les opérateurs de recherche locale utilisés dans la troisième méthode.

La sixième méthode approchée imite l'impérialisme, c'est-à-dire la situation où un pays souhaite étendre son pouvoir et son influence sur les autres. Chaque pays représente une solution, appelé une colonie, et celui qui a la valeur minimale devient un impérialiste.

Les résultats des méthodes proposées sont comparés à ceux obtenus par le solveur CPLEX en fonction de leurs gaps (bornes inférieures si leurs solutions optimales ne sont pas obtenues) et leurs temps de calcul en sous-section 3.3.2. Les résultats montrent que la meilleure méthode est l'*Two-Phase Iterative Heuristic* avec ses améliorations en proposant un gap moyen inférieure de 1,68 % et un temps de calcul moyen très vite.

Une partie de ce chapitre est l'objet de l'article suivant :

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. An Efficient Two-Phase Iterative Heuristic for Collection-Disassembly Problem, *Computers & Industrial Engineering*, vol. 110, 2017, p. 505-514.

Dans la réalité, les chaînes logistiques inverses font souvent à des incertitudes tels

que la quantité et la qualité de produits en fin de vie, ainsi que les demandes de leurs composantes. Pour cela, une formulation consacrée à ces incertitudes basée sur la programmation stochastique est nécessaire. En outre, plusieurs véhicules pourraient être disponibles pour collecter les produits en fin de vie depuis les centres de collecte. Dans le chapitre suivant, une formulation considérant ces incertitudes et le cas multi-véhicule est présentée.

Méthodes de résolutions pour le problème stochastique de collecte et désassemblage dans le cas multi-véhicule

Ce chapitre introduit le problème de collecte et désassemblage dans le cas multi-véhicule et mono-produit sous incertitude. Les paramètres incertains concernent (i) la quantité de produits en fin de vie dans centres de collecte, (ii) la quantité de leurs composants représentant la qualité de produits en fin de vie et (iii) les demandes des composants pour chaque période. La section 4.3 présente une formulation basée sur la programmation stochastique en bi-niveau où nous admettons que :

- les paramètres sous incertitude suivent des lois de distribution connues,
- leurs valeurs sont connues après la période de planification,
- plusieurs véhicules sont disponibles pour collecter les produits en fin de vie à chaque période.

Dans cette formulation, la fonction objectif a pour but de minimiser le coût concernant les décisions du premier niveau et l'espérance du coût des décisions du deuxième niveau. Le premier niveau correspond aux coûts de setup et de voyage de véhicules utilisés. Le deuxième niveau correspond aux décisions du niveau de stockage de produits en fin de vie, de la quantité produits en fin de vie à désassembler et de la pénalité des demandes de composant non-satisfaites.

Les contraintes sont (i) les contraintes imposant que chaque centre de collecte soit visité au maximum une fois dans chaque période, (ii) les contraintes de la conservation de flux de véhicule dans chaque période, (iii) les contraintes d'élimination de sous-tours dans chaque période selon un scénario, (iv) les conservations de flux de produits en fin de vie dans le stock dans chaque période selon un scénario, (v) les contraintes déterminant les demandes de composant non-satisfaites dans chaque période selon un scénario et (vi) les contraintes de définition des variables de décision.

Deux méthodes de résolution sont développées pour fournir les solutions du problème traité : la *Two-Phase Iterative Heuristic* et sa version avec des améliorations, décrites en sous-section 4.4.1. Nous avons choisi de ré-adapter ces méthodes pour le problème stochastique car elles ont montré une flexibilité et de très bonnes performances pour le problème déterministe. Pour fournir une solution avec bonne qualité pour chaque

instance testée, les deux méthodes sont mises en œuvre avec la procédure d'*approximation moyenne par échantillonnage* (angl. *Sample Average Approximation*). Cette procédure se base sur l'échantillonnage de la méthode de simulation Monte Carlo afin de résoudre un problème avec un très grand nombre de scénarios, qui est intraitable, en le divisant en de plus petits scénarios qui sont résolubles (Adulyasak et al., 2015a; Ghilas et al., 2016; Kleywegt et al., 2002).

D'après les résultats obtenus dans la sous-section 4.5.2, la *Two-Phase Iterative Heuristic* avec des améliorations montre une meilleure performance que l'autre méthode proposée pour deux ensembles de données. Cette conclusion est faite en analysant les bornes inférieures et supérieures pour chaque méthode.

Un article basé sur ce chapitre est actuellement en préparation comme suivant :

- Habibi, M. K. K., Battaïa, O., Cung, V.-D., Dolgui, A., Tiwari, M. K. Sample Average Approximation for Multi-Vehicle Collection-Disassembly Problem under Uncertainty (soumis à la revue de *International Journal of Productions Research*).

Il présente un travail qui a été fait en partenariat avec l'institut indien de technologie de Kharagpur.

Le problème intégré d'approvisionnement et désassemblage

Un autre problème appelé *le problème intégré d'approvisionnement et désassemblage* est présenté dans le chapitre 5. Contrairement aux problèmes traités dans les chapitres précédents, ce problème met l'accent sur les décisions conjointes de la collecte de produits en fin de vie et planification de la ligne de désassemblage. Nous admettons que la ligne de désassemblage peut s'adapter aux changements de décisions sur le processus de collecte. Cette hypothèse se base sur le fait que les processus de désassemblage sont toujours effectués manuellement.

Une formulation est développée pour formaliser le problème. La fonction objectif vise à minimiser les coûts totaux d'approvisionnement et de setup à travers l'horizon de planning. Les contraintes associées sont (i) les contraintes de la conservation de flux de stockage, (ii) les contraintes imposant que chaque centre de collecte soit visité au maximum une fois dans chaque période, (iii) les contraintes d'élimination de sous-tours, (iv) les contraintes des relations de précédence entre les tâches de désassemblage, (v) les contraintes de l'affectation d'une tâche sur un poste de travail et (vi) l'ensemble des contraintes qui définissent les variables de décision.

Ce chapitre a abouti à l'article suivant:

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Integrated procurement–disassembly problem, En: *Advances in Production Management Systems*:

Innovative and Knowledge-Based Production Management in a Global-Local World, Part II, B. Grabot, B. Vallespir, S. Gomes, A. Bouras, D. Kiritsis (Eds.), Series: IFIP Advances in Information and Communication Technology, Springer, vol. 439, 2014, ISSN: 1868-4238, p. 382–390. (IFIP WG 5.7 International Conference, APMS 2014, Ajaccio, France, Du 20 au 24 Septembre 2014).

Conclusions générales

L'intégration des décisions des processus de collecte de produits en fin de vie et de leur désassemblage mène à une meilleure performance des chaînes logistiques. Afin de traiter les instances de grande taille, pour lesquelles le solveur CPLEX n'est pas capable de fournir les solutions optimales en un temps raisonnable, nous avons développé des méta-heuristiques. La *Two-Phase Iterative Heuristic* avec des améliorations est la méthode la plus performante parmi les autres méthodes développées, selon le gap moyen et le temps de calcul. Nous avons également développé une formulation basée sur la programmation stochastique en bi-niveau pour traiter l'incertitude concernant la quantité de produits en fin de vie dans centres de collecte, la quantité de leurs composants et les demandes des composants pour chaque période. La *Two-Phase Iterative Heuristic* et ses améliorations avec la procédure d'approximation moyenne par échantillonnage est capable de proposer de meilleures solutions que l'autre méthode. Un autre problème se focalisant sur la collecte de produits en fin de vie et la planification de leur ligne de désassemblage est également introduit.

Des perspectives de notre recherche sont présentées. Concernant le problème déterministe, le cas multi-véhicule et multi-produit en fin de vie pourraient être étudié conjointement. Pour traiter ces deux facteurs, deux indices supplémentaires seraient nécessaires dans la formulation présentée dans le chapitre 2. Un autre type des méthodes de résolution de [Absi et al. \(2014\)](#) appelée l'*Iterative Method-Vehicle Routing Problem* (IM-VRP) pourrait être implémentée et comparée avec la meilleure méthode proposée pour ce problème déterministe. Cette méthode de décomposition IM-VRP propose que l'affectation des centres de collecte à visiter par les véhicules soit déterminée pendant la construction de leurs tournées et exclue du problème de dimensionnement de lot avec les coûts approximatifs de transport.

Dans le *problème stochastique de collecte et désassemblage dans le cas multi-véhicule* en section 4.3, nous admettons que toutes les valeurs des paramètres incertains soient réalisées juste après la période de planification (la période zéro). Leurs valeurs pourraient en plus être réalisées dans chaque période tout au long de l'horizon de planification. Dans ce cas, la programmation stochastique en multi-niveau pourrait être utilisée dans ce problème en ajoutant les contraintes qui assurent la consistance des variables de décision telles que le niveau de stockage, la quantité de produits en fin de vie à désassembler et les demandes non-satisfaites.

Dans les chaînes logistiques inverses, il y a souvent plusieurs types de produits. Cela signifie qu'une formulation dans le cas multi-produit basée sur la programmation stochastique soit également nécessaire. Par conséquent, un indice supplémentaire associé aux produits doit être intégré dans la formulation. La méthode approchée *Two-Phase Iterative Heuristic* pourrait être adaptée et combinée avec le *rollout algorithm* de Bertsekas et al. (1997) avec la procédure d'approximation moyenne par échantillonnage afin de résoudre ce problème.

Summary

Due to the drawbacks of the linear economy, the implementation of the circular economy is more and more encouraged by several governments due to its advantages in terms of financial, social and environmental aspects. It is performed by incorporating reverse supply chains into existing forward supply chains. However, many companies collaborate with third-party reverse logistics providers that own competencies to manage the complexity of reverse supply chains. Furthermore, managing supply chain by integrating two or more functions leads to better inventory management, better response to market challenges and higher efficiency.

This dissertation supports not only the implementation of the circular economy but also proposes better management by integrating functions in reverse supply chains. The hypothesis of this dissertation is that integrating functions in reverse supply chain leads to propose better decisions. The functions concerned are the collection process of End-of-Life products and their disassembly process since both processes hold important roles in reverse supply chains.

First, a deterministic problem integrating both two processes is introduced and called *Collection-Disassembly Problem*. A corresponding formulation of the problem is developed and some instances are generated accordingly due to lack of available instance in the literature. Another non-integrated formulations are also developed and solved for the generated instances using a commercial solver namely CPLEX. The obtained results show that the integrated formulation proposes better decisions than the non-integrated formulations in terms of the optimal cost.

However, the commercial solver CPLEX are unable to provide optimal solutions under acceptable CPU times notably for large size instances. Therefore, some approximate methods are developed to propose (near) optimal solutions under shorter CPU times. According to the obtained results, the *Two-Phase Iterative Heuristic* with Enhancements offers the best performance compared to the other proposed methods.

Second, reverse supply chains frequently deal with the uncertainty of some parameters such as the quantity and the quality of End-of-Life products as well as the demands of their components. An extended version of the Collection-Disassembly Problem under uncertainty of the concerning parameters is introduced. Furthermore, there are often more than one vehicle available to collect the products. Thus, the problem called *Stochastic Multi-Vehicle Collection-Disassembly Problem* and its formulation are also developed.

The problem is formalised as a two-stage stochastic programming where the parameters under uncertainty follow some known probability distribution and their realisation comes after the planning stage. The first-stage variables correspond to the routing of vehicles dispatched and the second-stage variables correspond to the decisions of disassembly lot-sizing of End-of-Life products, their inventory and the unmet demands of component.

Accordingly, two methods combined with an algorithmic framework of *Sample Average Approximation* are developed to provide high quality solutions of the stochastic problem. The obtained results show that *Two-Phase Iterative Heuristic* with Enhancements combined with the framework outperforms the other developed method in the tested instances.

Third, another problem called *Integrated Procurement-Disassembly Problem* is also studied. Unlike the previous problem, this problem emphasises on the decisions of disassembly line balancing problem.

Finally, some prospects of future work are also provided.

Acknowledgements

My deepest gratitude goes to Professor Alexandre DOLGUI for his supervision, supports, guidances and motivations for my doctoral studies. I also gratefully thank Professor Van-Dat CUNG as the co-supervisor of this dissertation. I would like to extend my sincere gratitude to Professor Olga BATAÏA for all her time, dedication, insights and motivations. Thanks to them, I had opportunities to attend and meet exceptional scientists in many occasions.

I deeply thank the government of Auvergne-Rhône-Alpes region for their support to finance my doctoral studies under the ARC 8 doctoral scholarship. My sincere gratitude is also addressed to the government of France and India that awarded me as fellow of the Raman-Charpak Doctoral Exchange Programme 2015. Thanks to this fellowship, I had an opportunity to perform an exchange in India for two months.

It has been a great honour to be part of the department of *Génie Mathématique et Industriel* of the Henri Fayol Institute of the École Nationale Supérieure des Mines de Saint-Étienne. My personal gratitude goes to Dr. Paolo GIANESSI who has helped, taught and inspired me to develop critical parts of this dissertation as well as personal consultation during my doctoral studies. I deeply thank Dr. Audrey CERQUES for her helps in my dissertation. My sincere appreciation goes to Professor Xavier DELORME for his valuable suggestions to improve this dissertation. I also gratefully thank to my fellows at the department: Amine, Oussama, Andrés, Espèranc, Hassan, Hossein, Mickaël, Lounès, Sergey, Afafe, Akram and Jean-Charles. I am deeply indebted to Professor Mireille BATTON-HUBERT for her helps, suggestions and supports. I owe Madame Christine EXBRAYAT for her hospitality, administrative supports and personal helps.

My gratitude also goes to all members of the Indonesian Students' Association in France and Franco-Indonesian families in Saint-Étienne. I also thank all my friends and colleagues at IIT Kharagpur for all memories during my exchange programme.

Finally, I would like to profoundly thank my family, especially my beloved mother and brother, for their unconditional love and encouragements throughout all these years. My greatest thank and deepest love go to my beloved wife for her love, patience and understanding although the distance is currently against us. I dedicate this dissertation to them.

Contents

List of Figures	xvii
List of Tables	xix
1 Decisions in Forward and Reverse Supply Chains	1
1.1 Circular Economy	1
1.2 Forward and Reverse Supply Chains	3
1.2.1 Network Design	5
1.2.2 Routing Transportation	10
1.3 Disassembly in Supply Chains	11
1.4 Production-Distribution Problem	12
1.4.1 Formulations	13
1.5 Research Gaps	21
2 Collection-Disassembly Problem	23
2.1 Introduction	23
2.2 Problem Description	23
2.3 Formulations	25
2.3.1 Case I (with integration)	25
2.3.2 Case II (without integration)	26
2.4 Instance Generation	28
2.5 Numerical Analysis	29
2.6 Conclusions	34
2.7 Publications	35
3 Approximate Methods for Collection-Disassembly Problem	37
3.1 Introduction	37
3.2 Approximate Methods	37
3.2.1 Two-Phase Iterative Heuristic	38
3.2.2 Local Search-based Heuristic	45
3.2.3 Particle Swarm Optimisation	49
3.2.4 Imperialist Competitive Algorithm	52

CONTENTS

3.3	Numerical Experiments	55
3.3.1	Experimental Setup	55
3.3.2	Results	56
3.4	Conclusions	59
3.5	Publication	60
A	Average Gaps and CPU Times	61
B	Detail Results of Data Set I	65
C	Details Results of Data Set I within 100 seconds	69
4	Stochastic Multi-Vehicle Collection-Disassembly Problem	73
4.1	Introduction	73
4.2	Problem Description	74
4.3	Formulations	75
4.4	Solution Methods	78
4.4.1	Two-Phase Iterative Heuristic	78
4.4.2	Sample Average Approximation	83
4.5	Numerical Experiments	84
4.5.1	Experimental Setup	84
4.5.2	Results	85
4.6	Conclusions	86
4.7	Publication	87
5	Integrated Procurement-Disassembly Problem	93
5.1	Introduction	93
5.2	Problem Definition	94
5.3	Formulation	95
5.4	Instance Generation	97
5.5	Numerical Experiments	97
5.6	Conclusions	98
5.7	Publication	98
6	General Conclusions	101
A	Publications	105
	Bibliography	107

List of Figures

1	Circular Economy (Forum, 2014)	2
2	Basic Activities and Flows in Closed-Loop Supply Chain (Gupta, 2013) .	3
3	Network Representations of Production-Distribution Problem (Adulyasak et al., 2015b)	12
4	Network Representations of Our Problem	22
5	Relation of Disassembly Lot-Sizing and Collection Routing	26
6	Data Sets	28
7	Results of Data Set I	32
8	CPU Times of Data Set I (in seconds)	33
9	Results of Data Set II	34
10	CPU Times of Data Set II (in seconds)	35
11	Results of Data Set III	35
12	CPU Times of Data Set III (in seconds)	36
13	Disassembly Lot-Sizing Problem with Approximate Visiting Costs	40
14	Solution Representation at Period t	45
15	1-1 Exchange Operator	46
16	2-Opt Operator	46
17	1-0 Exchange Operator	46
18	Change $route_t$, Keep $route - bin_t$	47
19	Change $route-bin$, Keep $route$	47
20	Keep $route$ and $route-bin$, Change Orders	47
21	Diversification	48
22	Representation of Variable $random-number_{kt}$	51
23	Gap When Optimal Solution Found at First Time and Its Lower Bound .	57
24	Results of Data Set - Random 1 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)	58
25	Results of Data Set - Random 2 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)	58
26	Results of Data Set - Cluster 1 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)	59
27	Results of Data Set - Cluster 2 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)	59
28	Two-Stage Stochastic Problem	74

LIST OF FIGURES

29	Two-Stage SMCDP	75
30	Average CPU Times (in seconds) for All \mathcal{M}	85
31	Average CPU Times (in seconds) for All \mathcal{K}	86
32	Multi-Stage Stochastic Problem	103
33	Multi-Stage SMCDP with Two Scenarios	103

List of Tables

1.1	Differences between Forward and Reverse Supply Chains (Gupta, 2013)	6
1.2	Major Reverse Supply Chains Decisions (Gupta, 2013; Lambert et al., 2011)	7
2.1	Location of Collection Centres in Cluster Category	29
2.2	Parameters of Data Set I	30
2.3	Parameters of Data Set II	30
2.4	Parameters of Data Set III	31
3.1	Global Average Gaps (in %)	55
3.2	Global Average CPU Times (in seconds)	56
3.3	Global Average Gaps (in %) within 100 seconds	57
A.1	Average Gaps (in %) based on Parameters of Data Set I	61
A.2	Average CPU Times (in seconds) based on Parameters of Data Set I	62
A.3	Average Gaps (in %) based on Parameters of Data Set II	62
A.4	Average CPU Times (in seconds) based on Parameters of Data Set II	62
A.5	Average Gaps (in %) based on Parameters of Data Set III	62
A.6	Average CPU Times (in seconds) based on Parameters of Data Set III	63
B.1	Average Gaps (in %) of Data Set I - Random 1	65
B.2	Average CPU Times (in seconds) of Data Set I - Random 1	65
B.3	Average Gaps (in %) of Data Set I - Random 2	66
B.4	Average CPU Times (in seconds) of Data Set I - Random 2	66
B.5	Average Gaps (in %) of Data Set I - Cluster 1	67
B.6	Average CPU Times (in seconds) of Data Set I - Cluster 1	67
B.7	Average Gaps (in %) of Data Set I - Cluster 2	68
B.8	Average CPU Times (in seconds) of Data Set I - Cluster 2	68
C.1	Average Gaps in Data Set - Random 1 (in %) within 100 seconds	69
C.2	Average Gaps in Data Set - Random 2 (in %) within 100 seconds	70
C.3	Average Gaps in Data Set - Cluster 1 (in %) within 100 seconds	70
C.4	Average Gaps in Data Set - Cluster 2 (in %) within 100 seconds	71

LIST OF TABLES

4.1	Instances	84
4.2	Results of All Data Sets	85
4.3	Results of Random 1	88
4.4	Results of Random 2	89
4.5	Results of Cluster 1	90
4.6	Results of Cluster 2	91
5.1	Part and Demand Data	97
5.2	Supplier Data	97
5.3	Vehicle Trip	98
5.4	Task Assignment	98

List of Algorithms

1	Two-Phase Iterative Heuristic (IM-MultiTSP) for PDP	39
2	Two-Phase Iterative Heuristic for CDP	42
3	Update of Approximate Visiting Costs SC_{it}	42
4	Two-Phase Iterative Heuristic with Enhancements for CDP	44
5	Local Search-based Heuristics	48
6	Initialisation of Local Search-based Heuristic	49
7	Initialisation of Particle Swarm Optimisation	50
8	Particle Swarm Optimisation	52
9	Imperialist Competitive Algorithm	53
10	Initialisation of Countries	54
11	Two-Phase Iterative Heuristic for SMCDP	80
12	Update of Approximate Visiting Costs SC_{it}^k	81
13	Two-Phase Iterative Heuristic with Enhancements for SMCDP	82

LIST OF ALGORITHMS

Chapter 1

Decisions in Forward and Reverse Supply Chains

1.1 Circular Economy

The linear economy has been commonly employed as traditional "take-make-dispose" pattern to fulfil consumers' needs. Companies extract raw materials from the nature, transforms them into final product, distribute and sell them to consumers. If the product reaches its end-use phase, commonly called End-of-Life (EOL) product, it is often discarded as disposal.

According to the report of World Economic Forum ([Forum, 2014](#)), the linear economy is reaching its limits by looking on the following facts:

- higher resource prices and supply disruptions,
- price volatility of metals, foods and non-agriculture outputs achieves higher points since the start of 21th century,
- opportunity to increase efficiency exists but it is not possible to create sufficient competitive advantage or differentiation,
- unpredicted consequences of the improvement of energy and resource efficiency drives the increase of amount of materials and energy used,
- growth of agriculture productivity is slower followed by the decline of the soil fertility and the nutrition of agricultural product,
- risk of global supply chain's supply security and safety increases,
- number of production sites faces problem to get virgin resources (water, land and atmosphere).

Furthermore, the linear economy also leads to the burden of waste, notably waste of electrical and electronic equipment (WEEE). However, WEEE often contains hazardous materials (e.g. mercury, etc.) and also precious metals (e.g. gold, silver, etc). United

1.1. CIRCULAR ECONOMY

Nations Environment Programme reported that 20 to 50 millions tonnes of WEEE are generated worldwide annually. In particular, the annual waste of European Union (EU) member states is around 4 million tonnes.

To tackle these issues, transforming the linear economy into the circular one (Figure 1) is more and more encouraged by various governments through directives such as EU Directive 2002/96/EC and 2011/65/EU, and inter-governmental agreements such as Paris Agreement (*Accord de Paris sur le Climat*) of the 21st session of the Conference of the Parties (COP 21) in 2015.

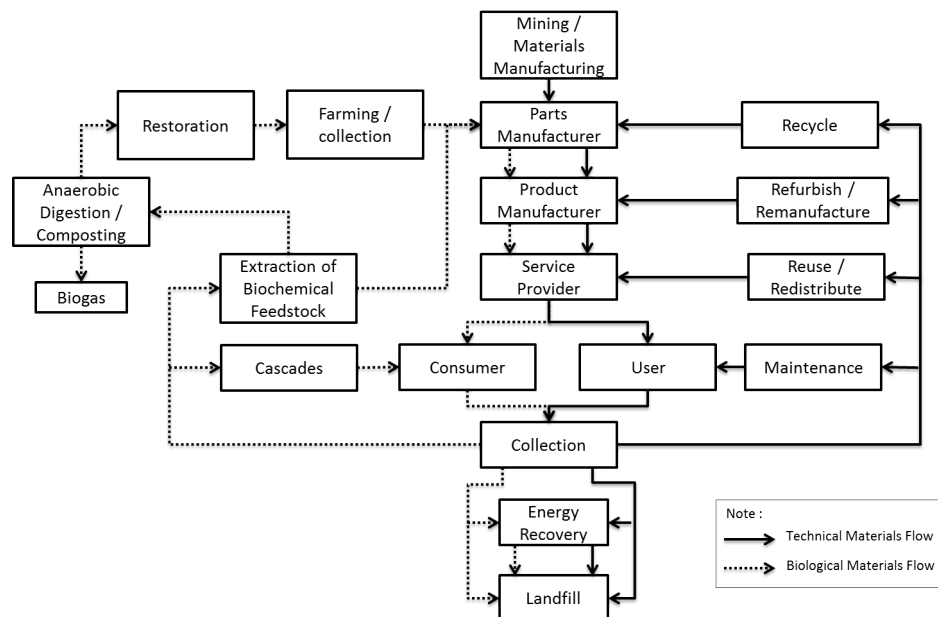


Figure 1 – Circular Economy (Forum, 2014)

This transformation is important since the circular economy is proven to yield benefits not only for business stakeholders but also consumers and the whole society both operationally and strategically. In detail, it yields net annual material cost savings in manufacturing sector up to US\$ 630 billion. It is also likely to mitigate price volatility and supply risks. The demand of creating reverse logistics network will evoke innovation and potentially create new jobs. It improves land productivity and soil health since it emphasizes on the anaerobic digestion or composting process and back into soil. It permits to reduce the replenishment of additional nutrients. It leads to reduce of the dependency on resource markets and, thus, supports more resilient economy (Forum, 2014). Thus, the circular economy complies the Triple-Bottom-Lines since it comes up with positive impacts on economic, social and environmental aspects. Renault, Phillips, Xerox, Hewlett-Packard and Caterpillar are among success stories regarding the imple-

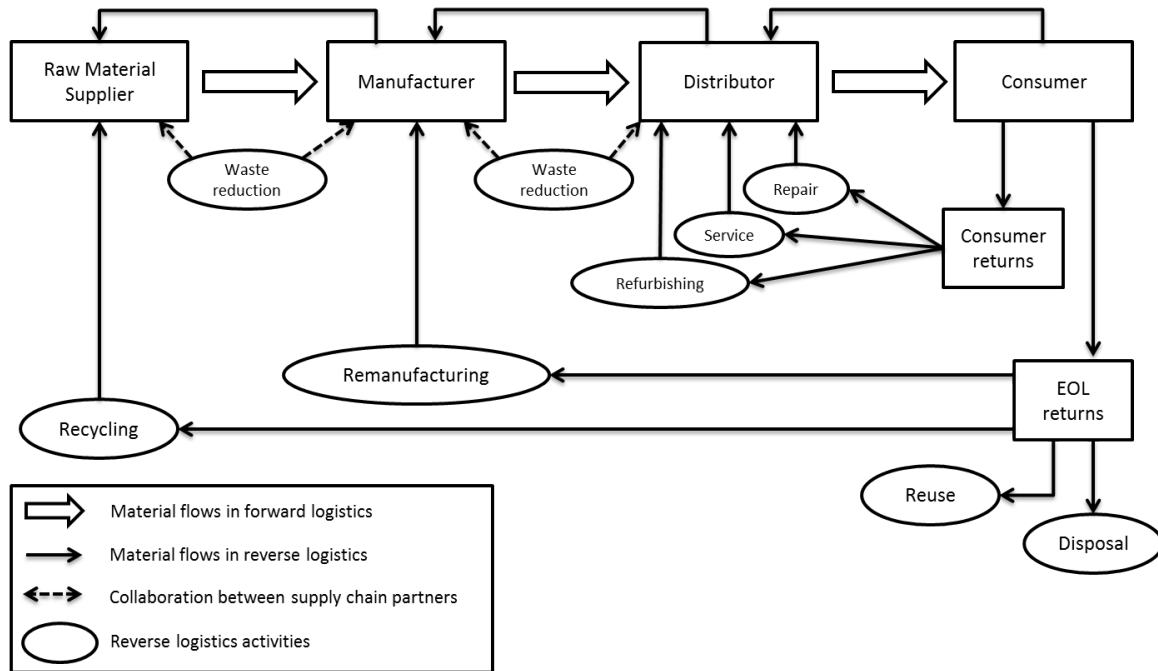


Figure 2 – Basic Activities and Flows in Closed-Loop Supply Chain (Gupta, 2013)

mentation of circular economy (Alumur et al., 2012; Forum, 2014; Kumar and Putnam, 2008; Pishvae et al., 2010).

In order to comply with the circular economy, the business stakeholder needs to re-design their forward supply chains (CLSC) for resource efficiency and circularity by incorporating reverse supply chains to form so called closed-loop supply chains as depicted in Figure 2. The reverse supply chains principally deal with the activities of reusing, repairing, refurbishing and recycling existing materials and products.

However, reverse supply chains differs from forward one in various aspects (Gupta, 2013; Tibben-Lembke and Rogers, 2002). The following section details the differences between the forward and reverse supply chains.

1.2 Forward and Reverse Supply Chains

This part presents the differences between forward and reverse supply chains (RSC) in terms of various aspects as well as duration-based decisions. Existing works related to RSC are also reviewed. We also suggest the reader to see Dolgui and Proth (2010) for deeper insights regarding forward supply chains.

For aspects distinguishing both supply chains as provided in Table 1.1, we divide them into for set of aspects; planning, production, distribution and revenue.

Planning Aspects

In forward supply chains, the objective is traditionally to optimise the profit and/or cost. Apart from profit/cost optimisation, RSC is mainly triggered by environmentally conscious principles and laws of governments. Unlike forward supply chains, the common forecasting techniques may need to be adjusted in RSC due to high level of uncertainty related to product returns.

Production Aspects

The new product quality manufactured in forward supply chains is highly controlled to meet the quality standard. In contrast, the quality of returned products in RSC vary. They may be highly degraded due to consumer usage or even have higher quality due to modification during the usage. The processing times and steps for a new product are well defined. However, the returned product often has various conditions led to various processing times and steps. The packaging of new products is highly available for the reason of protection, handling and identification. In RSC, returned products are rarely well packaged. New products have fixed nomenclature to pass the quality inspection and standard before being delivered. Returned products, particularly EOL products, have many missing, modified or damaged parts due to the usage. Therefore, the nomenclature of returned products is not fixed. Since product returns often have a high level of uncertainty, the inventory models in forward supply chains cannot be properly applied in RSC. The reader is suggested to see [Dolgui and Proth \(2006\)](#) for further interests in production systems.

Distribution Aspects

The new products are transported from the production site to many other locations e.g. distribution centres, retailers, customers, etc. Whilst, the returned products are collected from many locations (e.g. collection centres, customers) to one processing facility. It indicates that the flow from collection centres to the processing facility is complex and depends on the number of collection centres as well as the quantity and the quality of returned products. The speed of delivery holds critical role since the customer can refuse to buy undelivered or delayed new products from the firm. In RSC, the returned products are received by the firm itself. Hence, there is no urgency to receive them in fast delivery mode.

Revenue Aspects

In forward supply chains, the final objective is the sale of the product to customers. In RSC, it is not clear since it depends on the product type and condition of returned product. For manufacturing companies, their primary importance is forward supply chains since their revenue is generated from new products that are distributed through forward supply chains. For remanufacturing or recycling companies, the primary importance is RSC since they recover parts of materials from EOL products.

Thus, the condition of returned EOL products has big impact on how to manage RSC. Particularly, the disassembly process as a mandatory step for RSC activities is costly and labour intensive and has high level of uncertainty due to the condition of

returned products.

As shown in Table 1.2, the decision of integrating forward and reverse flows in supply chains is categorised as strategic decision because of its long-term impacts as well as high cost. In current reality, manufacturers have opted to collaborate with other specialised companies to manage this reverse flow such as ENVIE (France) since RSC requires new facilities and its activities are typically costly and manual labour intensive. This fact has motivated this dissertation to propose decision support tools to decision makers in such companies. Furthermore, this dissertation has been put in place in the particular interest of the management of EOL products from the points of collection until re-manufacturers and/or recyclers. In the following part, some researches related to the aim of this dissertation are presented.

1.2.1 Network Design

As strategic level decision, this issue deals with high cost decisions due to its long duration impact. Once the decision makers decide to conduct the recovery process of used products, the recovery system requires some facilities such as the sites of collection/sorting, remanufacturing, refurbishing, recycling or disposal in order to reprocess the returned products. This decision requires facilities incurring enormous cost. The latter cause is the main reason for categorizing this research area into strategic issues. Pishvae et al. (2009b) argue that the considered decisions are to determine the numbers, locations and capacities of facilities and the material flows between them. Since some researches disregard the uncertainty factors of the problems, the following description are divided into deterministic and stochastic models.

Deterministic Models. In this part, some researches assuming that the necessary data are available and have no uncertainty. They consider the materials flow from users until recovery centres.

Lee et al. (2009) focus on minimisation the total cost of transportation and fixed opening facilities and took into consideration of multi-stage, multi-product and some specific conditions for disassembly as well as processing centres. Respecting the parts types, the returned products are delivered to either the processing center or the disassembly center.

A model determining the number, the location and allocation of facilities as well as the flow of used products for its market is proposed in Mutha and Pokharel (2009). The products are modular with different disposal and recycling fraction so that the returned modules are resold in the spare-part market. The warehouses, remanufacturing centre and manufacturer have a portion for processing the returned products.

A multi-period and multi-product reverse logistics network design problem is proposed in Alumur et al. (2012). The model takes into account modular capacities, capacity expansion of the facilities, reverse bill of materials, minimum throughput at the facilities, variable operational costs and finite demands in the secondary market. The decision

1.2. FORWARD AND REVERSE SUPPLY CHAINS

Table 1.1 – Differences between Forward and Reverse Supply Chains (Gupta, 2013)

Aspects	Forward	Reverse
Objective	Based on profit and cost optimisation (traditionally)	Based on environmentally conscious principles and laws as well as profit and cost optimisation
Forecasting	Relatively easier and straightforward for product demand	More difficult for product returns
Product Quality	Less variation	Highly stochastic
Marketing	Traditional marketing techniques can be applied	There are factors complicating marketing
Processing Times and Steps	Well defined	Depend on the condition of the returned product
Transportation	Goods are transported from one location to many other locations	Returned products collected from many locations arrive in one processing facility
Speed	Competitive advantage	Not a critical factor
Packaging	Standard	Highly variable
Product Structure	Standard	Modified
Cost Estimation	Easier due to accounting systems	Determination and visualisation is complicated
Disposition Options	Clear	Depend on the condition of returned product
Inventory Management	Consistent	Inconsistent
Financial Implications	Clear	Not clear
Process Visibility	Highly visible due to real-time product tracking	Less visible due to lack of information system capabilities for product tracking
Product Life Cycle Management	Relatively easier	More difficult
Deterministic / Stochastic	Relatively more deterministic	Relatively more stochastic
Primary Importance	To manufacturers	To EOL processors (i.e. remanufacturers, recyclers)

Table 1.2 – Major Reverse Supply Chains Decisions (Gupta, 2013; Lambert et al., 2011)

Strategic	Tactical	Operational
Whether or not to integrate reverse flow with the forward flow in supply chains	Decide transportation means and establish transportation routes	Logistics and operations scheduling
Allocate adequate financial resources	Establish operational policies (production and inventory)	Emphasise cost control
Categorise and define return policies	Define return policies for each item	Return acquisition activities
Determine reasons, stakeholders and issues related to reverse supply chains (RSC)	Define technical support to offer (in-store, subcontractors, etc.)	Consider time value of returns
Evaluate internal expertise in RSC and decide about outsourcing a few / all RSC activities	Do the RSC activities (transportation, warehousing, remanufacturing, etc., in-house or subcontract)	Train personnel on RSC concepts and practices
Implement environmental management systems and acquire knowledge of directives, laws and environmental rules	Develop a planning system for various RSC activities and establish quality standards for them	Manage information
Choose activities (repair / rework, reuse, etc.) and identify potential locations	Decide the location and allocation of capacities for RSC facilities	Determine level of disassembly
Risk assessment (value of information and uncertainties)	Define performance measures; optimise policies	Analyse returns in order to improve disposition

1.2. FORWARD AND REVERSE SUPPLY CHAINS

variables correspond to the location and the capacity of inspection and remanufacturing facilities, the capacity expansion of the existing facilities, the materials flow through the network, the amount of holding materials and purchased product from the suppliers for the remanufacturing plants.

A model minimising the total processing costs of WEEEs as well as the costs of collection, treatment and transportation is proposed in [Dat et al. \(2012\)](#). The sales income of returned products is taken into account. The model has four stages processing facility i.e. collection, disassembly, treatment (recycling facility and repairing facility) and final sites (disposal, primary market and secondary market).

The presence of second market hold significant role since it differs to the main market i.e. the price sensitive and the form of offered products (refurbished product, component/sub-assemblies or raw material). Taking into account multi-product factor affects the problem's complexity while gives more applicable solutions. Furthermore, most of reviewed researches employ a dedicated collection centre for the network whereas there is an opportunity to consider a hybrid site as both the collection centre of the used products and distribution center, simultaneously.

A location problem where the objectives are to determine the number and the location of collection centres as well as the incentive offered by the firms to consumers is studied in [Aras and Aksen \(2008\)](#). The number and location objective is important because the willingness of consumers to return the EOL product is based on the offered incentive beside their distance with the collection centres. Using the same objectives, a model investigated in [Aras et al. \(2008\)](#) takes into account the pick-up strategy that deals with capacitated vehicles picking up used products from consumers to collection centres. The distance between collection centres and customer zones incurs transportation cost.

[Hanafi et al. \(2008\)](#) investigate a case study of mobile phones in Australia. This work proposes an effective collection strategy by considering the economic and environmental impacts.

[Grunow and Gobbi \(2009\)](#) investigate new network of reverse logistics due to the altered environment of WEEE collection in Denmark. It determines the location points for collection centres. The model proposed takes into account the problem of collection point assignment precisely.

A decision support system (DSS) is studied in [Achillas et al. \(2010\)](#). It permits private sectors in Greece as well as public regulators to examine and determine the optimal locations of recycling facilities in Greece. The DSS takes into account both economical criteria (local population, population served, distance from existing recycle facilities, land value, land condition, distance from the capital of the region, distance from nearest port) and social criteria (unemployed population, financial status of local population). These criteria are combined by employing the multi-criteria methodology.

An evaluation of the existing facilities for processing EOL products is conducted in [Pochampally and Gupta \(2009\)](#). It aims to maximise the demand satisfaction for customers, the fulfilment of local government requirement concerning environmental con-

sciousness and the profit of supply chain actors.

The next issues is about structure of reverse channel for collecting the used products. A good reverse logistics structure allows to gain the efficiency of material flow.

A model proposed in [Lee et al. \(2011\)](#) incorporates pricing, production, and inventory decisions in reverse production system with retailer collection using continuous time and differential game framework. This study identifies the profitability of the entire supply chain under time-varying situation of post-consumer products.

Based on the case study of a manufacturer of EEEs in France, a framework is provided in [El korchi and Millet \(2011\)](#) to generate and assess different reverse logistics channel structures. This framework allows to analyse current structure and propose alternative structures in terms of its environmental and economic impacts by using a number of generic structures of reverse logistics through varying the location of treatment centres.

Stochastic Models. The data often contain uncertainty. The following part presents some researches in CLSC that deal with uncertainty by employing robust optimization, stochastic programming and fuzzy logic.

A robust optimisation model in [Pishvae et al. \(2011\)](#) takes into consideration the uncertainty of several parameters i.e. the supply of return products, the demand from second markets and transport costs between facilities in CLSC.

A stochastic model in [Chouinard et al. \(2008\)](#) determines the location of service and processing centres and warehouses of valorised products as well as materials flow. The uncertain parameters encompass the recovery, processing and demand volumes.

A stochastic model dealing with single-period single-product multi-stage closed-loop supply chain is proposed in [Pishvae et al. \(2009a\)](#). The uncertainty parameters deal with the quantity and quality of returned products, demands and variable costs.

A study of two-stage multi-period stochastic reverse logistics network design is investigated in [Lee and Dong \(2009\)](#) to deal with the option of close or open for every facility at the beginning of every period of time horizon. The demands of new products and supply of returned products at consumers are considered uncertain.

A multi-period multi-echelon network design of CLSC is proposed in [El-Sayed et al. \(2010\)](#) for a single type product. The objective is to maximise the expected profit. The studied types of markets are first market with stochastic demand and second market with deterministic demand.

A multi-objective possibilistic optimization model for CLSC is studied in [Pishvae and Torabi \(2010\)](#). It concerns about the uncertainties of demands, returns, delivery times, costs and capacities as well as the presence of second market of EOL products. The objective functions minimise the total cost and the total tardiness of delivered products.

Three optimisation models of network design in RSC is studied in [Qin and Ji \(2010\)](#) namely expected value model, chance constrained programming and dependent-chance programming. The models attempt to determine the location of collection centres and

returned products flow. The uncertainty of the models correspond to the quantity of products collected. They consider single-period and single-product.

A multi-objective CLSC model is proposed in [Zarandi et al. \(2011\)](#). Three models are presented for comparing the forward flow, the reverse flow and the integration of both flows considering the imprecision in decision makers' aspiration levels of the goals. The aspiration level corresponds to the degree of importance of the models' objectives.

1.2.2 Routing Transportation

It is well-known that transportation cost in distribution or collection contributes significantly in total cost of supply chains particularly in tactical-strategical decisions such as vehicle routes. Henceforth, this part provide several recent articles about vehicle routing problem (VRP) for collection process and in reverse supply chain.

A study of the American Red Cross' blood collection is performed in [Alshamrani et al. \(2007\)](#). It focuses on dynamic logistics planning problem with multi-period based on insulation-line distribution. It is formalised as VRP with pickup and delivery using single-vehicle. The objectives are to minimise the travelling cost and total expected penalty cost by designing an efficient routes in planned time horizon. The penalty cost is due to customer dissatisfaction associated with delay in the return of blood boxes.

Moreover, the capacity, speed and cost of vehicles often vary that impose challenging problem. Effort has been performed to deal with the problem so called VRP with heterogeneous fleet (VRP-HF).

A VRP-HF with multi-depot as collection center is studied in [Brandão \(2009\)](#). Working on the same VRP variant, the parameter of time windows for each pickup node is considered in [Xu et al. \(2012\)](#). The limit of vehicle journey is taken into account in [Salhi et al. \(2013\)](#). The minimisation of the number of vehicles dispatched is investigated in [Rafiei et al. \(2013\)](#). A VRP-HF taking into account splitted delivery with single-depot is investigated in [Belfiore and Yoshizaki \(2013\)](#).

Often, the depot functions as starting point instead of collection centre. Therefore, intermediate facilities as collection centres between customer nodes and depots are required. Herein, the collection area is divided into several zone with one depot. Such case where a VRP-HF with the limitation of available vehicles as well as time windows is studied in [Jiang et al. \(2014\)](#).

[Zhao and Zhu \(2015\)](#) investigate a multi depot VRP for the collection of explosive waste. This work is to minimise the transportation cost of explosive waste and the risk for explosive waste recycling during the transportation.

In the literature, different actors of supply chain are often considered separately. However, managing supply chain by integrating two or more functions leads to better inventory management, better response to market challenges and higher efficiency ([Chandra, 1993](#)). By extension, integrating two or more functions in RSC may also lead to gain efficiency. In contrast with the rising interest in the studies of decision integrations in

forward supply chains, only few works has targeted such integration in RSC.

1.3 Disassembly in Supply Chains

As aforementioned, disassembly process has important role in material and component recovery as well as is mandatory step in remanufacturing and recycling activities of RSC. According to [McGovern and Gupta \(2011\)](#), disassembly process is defined as a set of systematic activities which aim to extract the components, subassemblies and/or other grouping of recovered products. It interacts with all phases namely *before life* (the phase of design and life cycle analysis), *useful period*(the phase between the beginning of manufacturing and the end of in use) and *end-of-life*(the phase in which product completes is useful period and is ready for further processing such recovery and/or disposal).

Based on our knowledge, the majority of researches in disassembly process focuses on single problem such as sequencing ([Adenso-Díaz et al., 2008](#); [ElSayed et al., 2012](#); [Gupta and Imtanavanich, 2009](#); [Tripathi et al., 2009](#); [Yeh, 2012](#)), scheduling ([Barba-Gutiérrez et al., 2008](#)), line balancing ([Altekin et al., 2008](#); [Bentaha et al., 2014a,b,c,d, 2015](#); [Duta et al., 2008](#); [Gungor and Gupta, 1999](#); [Kalayci and Gupta, 2011, 2013a,b](#); [Koc et al., 2009](#)), etc.

Only few works have been conducted to deal with integration considering the disassembly process notably for decisions in strategic and tactical level. [Özceylan and Paksoy \(2013\)](#) investigate a network design optimisation of RSC by considering disassembly for EOL products. The objective is to minimise the transportation cost and the fixed cost of workstations of the disassembly line, simultaneously. The work of [Özceylan et al. \(2014\)](#) aims to determine the materials flow in forward and reverse flows of closed-loop supply chain as well as to balance the disassembly lines in the reverse chain in order to minimise the costs of transportation, purchasing, refurbishing and operations in the condition of multi-period and subassemblies. The decision variables encompass the number of disassembly workstations, the cycle time and the quantity of purchased subassemblies from supplier. To deal with the uncertainty of cost coefficients, capacity levels, market demands and reverse rates of the problem, an interactive fuzzy programming approach is proposed in [Özceylan and Paksoy \(2014\)](#). In these studies, disassembly process is taken into consideration as disassembly line balancing problem.

Since the integration of cross-function decisions in forward supply chains leads to gain more efficiency, our hypothesis is that such integration may also lead to better management and efficiency in RSC. Particularly, we interest on the decisions of collection process of EOL products and the choice to their components disassembled due to following motivations:

- The collection of EOL products in RSC triggers high transportation cost,
- The disassembly process is mandatory step in RSC,

- Integrating the decisions of these two processes may lead to better proposition to the decision makers.

In the next part, some related studies of integration in forward supply chains are presented to support our hypothesis.

1.4 Production-Distribution Problem

After some industrial practices of Vendor-Managed Inventory / Distribution e.g. Kellogg Company (Brown et al., 2001) and Frito-Lay’s North America (Çetinkaya et al., 2009), the integrated logistical planning is favourable for proposing a supply chain with better performance. Particularly, the coordinated management of production and distribution processes leads to the reduction of the total cost. It may take various configurations such as (i) integrated lot-sizing with direct shipment, (ii) inventory routing problem and (iii) production-distribution problem (PDP). The first configuration minimises the total cost of setup, production, inventory and direct shipment while disregarding the routing aspect. The second configuration exposes the decisions on routing aspect but ignores on production detail. Whereas, the third configuration focuses on both production and distribution aspects by incorporating the production decision and routing part in operational level decision as depicted in Figure 3.

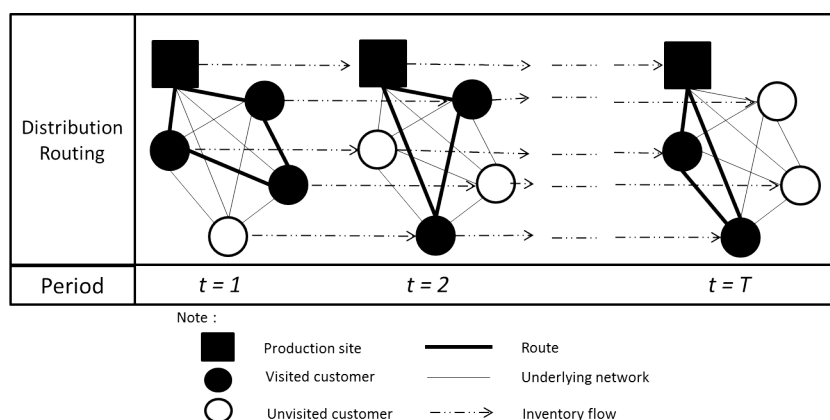


Figure 3 – Network Representations of Production-Distribution Problem (Adulyasak et al., 2015b)

Based on the existing literature of PDP that mostly deals with continuous products, it aims to minimise the total cost of production, inventory and routing, simultaneously, by respecting the demands of retailers, their inventory capacities, the production facility’s capacity and its inventory limit. In general, PDP is a combination of two well-known and hard combinatorial problems : lot-sizing problem and VRP.

The decisions of PDP throughout the planning horizon consist of:

- when and how much products to produce (continuous products),
- when to visit,
- how much to deliver to retailers,
- routing for each vehicle,
- inventory level of each retailer (Armentano et al., 2011; Bard and Nananukul, 2009, 2010; Boudia et al., 2007).

With respect to the inventory level due to its lot-sizing problem, there are two types of inventory policy considered namely *order-up-to level* (OU) and *maximum level* (ML). The OU policy imposes that a visited customer receives the amount of products equal to their predefined inventory level. The ML policy expresses that the quantity of products delivered is positive at most as the capacity of inventory (Archetti et al., 2011).

1.4.1 Formulations

This part provides four existing models of Production-Distribution Problem with Multi-Vehicle under ML policy where they are the main references for other researches in PDP. We put them because some ideas are reused to construct the formulations of our problems in the next chapters.

First, the formulation of Boudia et al. (2007) is completed with vehicle index. Second, the formulation proposed by Bard and Nananukul (2009, 2010) has no index regarding the vehicle. Third, the formulation of Armentano et al. (2011) deals with Multi-Product where the available vehicles are indexed. Finally, the formulation of Adulyasak et al. (2015a) deals with PDP under the uncertainty of demands of retailers.

The parameters and decision variables in these formulations are as follows:

Parameters :

- N set of nodes: $i, j \in \{0, 1, \dots, |N|\}$
- N_c set of customers : $i, j \in \{1, 2, \dots, |N|\}$
- T set of time periods: $t \in \{1, 2, \dots, |T|\}$
- K set of identical vehicles: $m \in \{1, \dots, |K|\}$
- u unit production cost
- f fixed production setup cost
- h_i unit inventory holding cost at node i
- c_{ij} transportation cost from node i to node j
- d_{it} demand at customer i in period t
- C production capacity
- Q vehicle capacity
- L_i maximum or target inventory level at node i
- I_{i0} initial inventory available at node i .

1.4. PRODUCTION-DISTRIBUTION PROBLEM

Let $M_t = \min\{C, \sum_{j=t}^l \sum_{t \in N_c} d_{ij}\}$ as a large number corresponding to the decision variables of the production quantity and the production setup. Let $\bar{M}_{it} = \min\{L_i, Q, \sum_{j=t}^l d_{ij}\}$ as a large number corresponding to the decision variables of the quantity delivered to customers and the vehicle routing.

Decision variables :

- p_t production quantity in period t
- I_{it} inventory at node i at the end of period t
- y_t $\begin{cases} 1 & \text{iff production takes place in period } t \\ 0 & \text{otherwise.} \end{cases}$
- z_{0t} number of vehicles leaving the plant in period t
- z_{it} $\begin{cases} 1 & \text{iff node } i \text{ is visited in period } t \\ 0 & \text{otherwise.} \end{cases}$
- x_{ijt} $\begin{cases} 1 & \text{iff a vehicle travels directly from node } i \text{ to node } j \text{ is visited in period } t \\ 0 & \text{otherwise.} \end{cases}$
- q_{it} quantity delivered to customer i in period t
- w_{it} load of vehicle before making a delivery to customer i in period t .

Formulation of PDP **Bard and Nananukul (2009, 2010)**

This formulation considers the multi-vehicle case in which the available vehicles are unindexed.

$$\min \sum_{t \in T} (up_t + fy_t + \sum_{i \in N} h_i I_{it} + \sum_{(i,j) \in A} c_{ij} x_{ijt}) \quad (1.1)$$

Subject to:

$$I_{0,t-1} + p_t = \sum_{i \in N_c} q_{it} + I_{0t}, \forall t \in T \quad (1.2)$$

$$I_{i,t-1} + q_{it} = d_{it} + I_{it}, \forall i \in N_c, \forall t \in T \quad (1.3)$$

$$p_t \leq M_t y_t, \forall t \in T \quad (1.4)$$

$$I_{0t} \leq L_0, \forall t \in T \quad (1.5)$$

$$I_{i,t-1} + q_{it} \leq L_i, \forall i \in N_c, \forall t \in T \quad (1.6)$$

$$q_{it} \leq \bar{M}_{it} z_{it}, \forall i \in N_c, \forall t \in T \quad (1.7)$$

$$\sum_{j \in N} x_{ijt} = z_{it}, \forall i \in N_c, \forall t \in T \quad (1.8)$$

$$\sum_{j \in N} x_{jit} + \sum_{j \in N} x_{ijt} = 2z_{it}, \forall i \in N, \forall t \in T \quad (1.9)$$

$$z_{0t} \leq m, \forall t \in T \quad (1.10)$$

$$w_{it} - w_{jt} \geq q_{it} - \bar{M}_{it}(1 - x_{ijt}), \forall (i, j) \in A, \forall t \in T \quad (1.11)$$

$$0 \leq w_{it} \leq Qz_{it}, \forall i \in N_c, \forall t \in T \quad (1.12)$$

$$p_t, I_{it}, q_{it} \geq 0, \forall i \in N, \forall t \in T \quad (1.13)$$

$$y_t, x_{ijt} \in \{0, 1\}, \forall i, j \in N, \forall t \in T \quad (1.14)$$

$$z_{it} \in \{0, 1\}, \forall i \in N_c, \forall t \in T \quad (1.15)$$

$$z_{0t} \in \mathbb{Z}^+, \forall t \in T \quad (1.16)$$

The objective function (1.1) minimises the total cost of production, setup, inventory and travelling cost. Constraints (1.2) and (1.3) balance inventory levels in the depot and customers, respectively. Constraints (1.4) force the setup and the capacity limitation of production level. Inventory levels are assured by constraints (1.5) and (1.6). Respecting the ML Policy, the number of delivered products is limited by constraints (1.7). Constraints (1.8) state that a node has to be visited once it is travelled. Constraints (1.9) keep the flow entering and leaving a node. Constraints (1.10) ensure the number of vehicles used cannot exceed the number of vehicles available. Constraints (1.11) are subtour elimination constraints based on [Desrochers and Laporte \(1991\)](#). Although these constraints provide weak lower bound, they are easily adapted to new problem. Constraints (1.12) impose the capacity of vehicle after visiting a node.

Formulation of PDP [Boudia et al. \(2007\)](#)

This formulation differs from the previous one due to the vehicle index.

$$\min \sum_{t \in T} (up_t + fy_t + \sum_{i \in N} h_i I_{it} + \sum_{(i,j) \in A} c_{ij} \sum_{k \in K} x_{ijkt}) \quad (1.17)$$

Subject to:

$$I_{0,t-1} + p_t = \sum_{i \in N} \sum_{k \in K} q_{ikt} + I_{0t}, \forall t \in T \quad (1.18)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} = d_{it} + I_{it}, \forall i \in N_c, \forall t \in T \quad (1.19)$$

$$p_t \leq M_t y_t, \forall t \in T \quad (1.20)$$

$$I_{0t} \leq L_0, \forall t \in T \quad (1.21)$$

$$I_{i,t-1} + \sum_{k \in K} q_{kit} \leq L_i, \forall i \in N_c, \forall t \in T \quad (1.22)$$

$$q_{ikt} \leq \bar{M}_{it} z_{ikt}, \forall k \in K, \forall i \in N_c, \forall t \in T \quad (1.23)$$

$$\sum_{k \in K} z_{ikt} \leq 1, \forall i \in N_c, \forall t \in T \quad (1.24)$$

$$\sum_{j \in N} x_{jik} + \sum_{j \in N} x_{ijkt} = 2z_{ikt}, \forall k \in K, \forall i \in N, \forall t \in T \quad (1.25)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijkt} \leq |S| - 1, \forall S \subset N_c : |S| \geq 2, \forall k \in K, \forall t \in T \quad (1.26)$$

$$\sum_{i \in N_c} q_{ikt} \leq Q z_{0kt}, \forall k \in K, \forall t \in T \quad (1.27)$$

$$p_t, I_{it}, q_{ikt} \geq 0, \forall i \in N, \forall k \in K, \forall t \in T \quad (1.28)$$

$$y_t, z_{ikt}, x_{ijkt} \in \{0, 1\}, \forall i, j \in N, \forall k \in K, \forall t \in T \quad (1.29)$$

The objective function (1.17) and constraints (1.18) - (1.25) are equivalent to (1.1) and (1.2) - (1.9). Subtour elimination constraints in (1.26) consider the vehicle index $k \in K$. These constraints provide better lower bounds than constraints (1.11). Constraints (1.27) are the capacity constraints corresponding to the load of vehicle after visiting a node.

According to Archetti et al. (2011) and Adulyasak et al. (2014b), the following constraints are proved being more efficient rather than constraints (1.26):

$$\sum_{i \in S} \sum_{j \in S} x_{ijkt} \leq \sum_{i \in S} z_{ikt} - z_{ekt}, \forall S \subset N_c : |S| \geq 2, \forall e \in S, \forall k \in K, \forall t \in T \quad (1.30)$$

Formulation of PDP Armentano et al. (2011)

This formulation uses vehicle index and deals with multi-products. The objective function considers also fix vehicle cost. Apart from aforementioned parameters, it also requires some following parameters:

Parameters :

- P set of products : $p \in \{1, 2, \dots |P|\}$
- d_{pit} demand of product p at node i during period t
- b_p time required to produce one unit of product p
- u_p unitary manufacturing cost of product p
- f_p setup cost of product p
- h_{pi} holding cost of product p at node i
- fv fixed cost per used vehicle
- C production capacity at the manufacturing plant (in time units)
- L_{pi}^{max} inventory capacity for product p at node i
- L_{pi}^{min} minimum inventory level for product p at node i
- L maximum length of each route
- M large number = $\sum_{p \in P} \sum_{i \in N} \sum_{t \in T} d_{pit}$

Decision variables :

- p_{pt} quantity of p produced at period t
- I_{pit} inventory level of p at node i at the end of period t
- q_{pikt} quantity of p delivered to i with vehicle k at period t
- w_{pijkt} quantity of p transported from i to j with vehicle k at period t
- y_{ipt} $\begin{cases} 1 & \text{iff a setup is performed for } p \text{ at period } t \\ 0 & \text{otherwise.} \end{cases}$
- $x_{ijk t}$ $\begin{cases} 1 & \text{iff vehicle } k \text{ travels from } i \text{ to } j \text{ at period } t \\ 0 & \text{otherwise.} \end{cases}$

$$\min \sum_{t \in T} \left(\sum_{p \in P} (u_p p_{pt} + f_p y_{ipt} + \sum_{i \in N} h_{pi} I_{pit}) + \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} c_{ij} x_{ijk t} + \sum_{k \in K} \sum_{j \in N} fv x_{0jkt} \right) \quad (1.31)$$

Subject to:

$$I_{p0t} = I_{p0,t-1} + p_{pt} - \sum_{i \in N_c} \sum_{k \in K} q_{pikt}, \forall p \in P, \forall t \in T \quad (1.32)$$

$$I_{pit} = I_{pi,t-1} + \sum_{k \in K} q_{pikt} - d_{pit}, \forall p \in P, \forall i \in N_c \quad (1.33)$$

$$\sum_{p \in P} b_p p_{pt} \leq C, \forall t \in T \quad (1.34)$$

$$p_{pt} \leq M y_{pt}, \forall p \in P, \forall t \in T \quad (1.35)$$

$$\sum_{i \in N, i \neq k} w_{pijkt} - \sum_{m \in N, m \neq k} w_{pjmkt} = q_{pikt}, \forall p \in P, \forall j \in N, \forall k \in K, \forall t \in T \quad (1.36)$$

$$\sum_{i \in N} \sum_{k \in K} w_{pi0kt} - \sum_{m \in N} \sum_{k \in K} w_{p0mkt} = - \sum_{i \in N} \sum_{k \in K} q_{pikt}, \forall p \in P, \forall t \in T \quad (1.37)$$

$$\sum_{p \in P} w_{pijkt} \leq Q x_{ijkt}, \forall i, j \in N, \forall k \in K, \forall t \in T \quad (1.38)$$

$$\sum_{i \in N} \sum_{j \in J} c_{ij} x_{ijkt} \leq L, \forall k \in K, \forall t \in T \quad (1.39)$$

$$\sum_{i \in N} x_{0jkt} \leq 1, \forall k \in K, \forall t \in T \quad (1.40)$$

$$\sum_{i \in N, i \neq k} x_{ijkt} - \sum_{m \in N, m \neq k} x_{jmkt} = 0, \forall j \in N, \forall k \in K, \forall t \in T \quad (1.41)$$

$$\sum_{i \in N} \sum_{k \in K} x_{ijkt} \leq 1, \forall j \in N, \forall t \in T \quad (1.42)$$

$$L_{pi}^{min} \leq I_{pit} \leq L_{pi}^{max}, \forall p \in P, \forall i \in N, \forall t \in T \quad (1.43)$$

$$p_{pt}, I_{pit}, q_{pikt}, w_{pijkt} \geq 0, \forall p \in P, \forall i, j \in N, \forall k \in K, \forall t \in T \quad (1.44)$$

$$y_{pt}, x_{ijkt} \in \{0, 1\}, \forall p \in P, \forall i, j \in N, \forall k \in K, \forall t \in T \quad (1.45)$$

The objective function (1.31) minimises the total cost of production, setup, inventory, fix vehicle and mileage. Constraints (1.32) and (1.33) balance the inventory level at depot and customers, respectively. The production capacity for each product is assured in constraints (1.34). Constraints (1.35) force that setup iff production is conducted. Constraints (1.36) and (1.37) conserve the production flow at customers and depot, respectively. Constraints (1.38) and (1.39) express the maximum capacity and the maximum length of vehicle, respectively. Constraints (1.42) ensure that each vehicle passes only single route. The flow conservation is guaranteed by constraints (1.41). Constraints (1.42) state that each node is visited at most once by at most one vehicle. The bound of inventory is declared by constraints (1.43). Constraints (1.44) define the lower bound

and the integrality of production, inventory, delivery and transport. Constraints (1.45) declare the nature of binary variables.

Formulation of Stochastic PDP [Adulyasak et al. \(2015a\)](#)

This formulation assumes that the demands follow some known probability distributions. Note that this formulation is equivalent with those in [Adulyasak et al. \(2014b\)](#); [Archetti et al. \(2011\)](#) except the decision variables related to production are associated with the scenario index. Also, this work proposes two-stage and multi-stage stochastic formulations.

Parameters :

Ω	finite set of scenario: $\omega = \{1, 2, \dots, \Omega \}$
ρ_ω	probability of scenario ω
N	set of nodes
N_c	set of customers
$E(S)$	set of edges $(i, j) \in E$
S	given set of nodes, $S \subset N$
$\delta(S)$	set of edges incident of a node set S
T	set of time periods: $t = \{1, \dots, T\}$
$d_{it\omega}$	demand of customer i in period t under scenario ω
σ	unit penalty cost of customer i if some demand is unmet at the end of period
L_i	inventory level of node i
$M_{t\omega}$	$\min\{C, \sum_{j=t}^l \sum_{i \in N_c}\}$
$M'_{it\omega}$	$\min\{L_i, Q, \sum_{j=t}^l d_{ij\omega}\}$

Decision variables :

y_t	$\begin{cases} 1 & \text{iff production takes place in period } t \\ 0 & \text{otherwise.} \end{cases}$
z_{ikt}	$\begin{cases} 1 & \text{iff node } i \text{ is visited by vehicle } k \text{ in period } t \\ 0 & \text{otherwise.} \end{cases}$
x_{ijkt}	number of times vehicle k travels directly between node i and node j in period t
y_{it}	vehicle load after visiting i at period t
$p_{t\omega}$	production quantity in period t under scenario $\omega \in \Omega$
$I_{t\omega}$	inventory at node i at the end of period t under scenario ω
$q_{ikt\omega}$	quantity delivered to customer i with vehicle k in period t under scenario ω
$e_{it\omega}$	unmet demand at customer i in period t associated with scenario ω .

First, we presents the formulation of 2-Stage SPDP as follows:

$$\min \sum_{t \in T} (f y_t + \sum_{(i,j) \in E} \sum_{k \in K} c_{ij} x_{ijkt} + \sum_{\omega \in \Omega} \rho_\omega (u p_{t\omega} + \sum_{i \in N} h_i I_{it\omega} + \sum_{i \in N_c} \sigma_i e_{it\omega})) \quad (1.46)$$

1.4. PRODUCTION-DISTRIBUTION PROBLEM

Subject to:

$$I_{0,t-1,\omega} + p_{t\omega} = \sum_{i \in N_c} \sum_{k \in K} q_{ikt\omega} + I_{0t\omega}, \forall t \in T, \forall \omega \in \Omega \quad (1.47)$$

$$I_{i,t-1,\omega} + \sum_{k \in K} q_{ikt\omega} + e_{it\omega} = d_{it\omega} + I_{it\omega}, \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (1.48)$$

$$I_{0t\omega} \leq L_0, \forall t \in T, \forall \omega \in \Omega \quad (1.49)$$

$$I_{it\omega} + d_{it\omega} \leq L_i, \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (1.50)$$

$$p_{t\omega} \leq M_{t\omega} y_t, \forall t \in T, \forall \omega \in \Omega \quad (1.51)$$

$$\sum_{i \in N_c} q_{ikt\omega} \leq Q z_{0kt}, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (1.52)$$

$$q_{ikt\omega} \leq M'_{it\omega} z_{ikt}, \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (1.53)$$

$$\sum_{k \in K} z_{ikt} \leq 1, \forall i \in N_c, \forall t \in T \quad (1.54)$$

$$\sum_{(j,j') \in \delta(i)} x_{jj'kt} = 2z_{ikt}, \forall i \in N, \forall k \in K, \forall t \in T \quad (1.55)$$

$$\sum_{(i,j) \in E(S)} x_{ijkt} \leq \sum_{i \in S} z_{ikt} - z_{ekt}, \forall S \subset N_c : |S| \geq 2, \forall e \in S, \forall k \in K, \forall t \in T \quad (1.56)$$

$$e_{it\omega}, p_{t\omega}, I_{it\omega}, q_{ikt\omega} \geq 0, \forall i \in N, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (1.57)$$

$$y_t, z_{ikt} \in \{0, 1\}, \forall i \in N, \forall k \in K, \forall t \in T \quad (1.58)$$

$$x_{ijkt} \in \{0, 1\}, \forall (i, j) \in E : i \neq 0, \forall k \in K, \forall t \in T \quad (1.59)$$

$$x_{ojkt} \in \{0, 1, 2\}, \forall j \in N_c, \forall k \in K, \forall t \in T \quad (1.60)$$

The following valid inequalities are used to strengthen the model:

$$z_{ikt} \leq z_{0kt}, \forall i \in N_c, \forall k \in K, \forall t \in T \quad (1.61)$$

$$x_{ijkt} \leq z_{ikt} \text{ and } x_{ijkt} \leq z_{jkt}, \forall (i, j) \in E(N_c), \forall k \in K, \forall t \in T \quad (1.62)$$

$$z_{0kt} \geq z_{0,k+1,t}, \forall 1 \leq k \leq m-1, \forall t \in T \quad (1.63)$$

$$\sum_{i=1}^j 2^{(j-i)} z_{ikt} \geq \sum_{i=1}^j 2^{(j-i)} z_{i,k+1,t}, \forall j \in N_c, \forall 1 \leq k \leq m-1, \forall t \in T \quad (1.64)$$

The following non-anticipativity constraints are used for formalising multi-stage stochastic formulation:

$$p_{t\omega} = p'_{t,\mathcal{H}'\omega}, \forall t \in T, \forall \omega \in \Omega \quad (1.65)$$

$$I_{it\omega} = I'_{it,\mathcal{H}'\omega}, \forall i \in N, \forall t \in T, \forall \omega \in \Omega \quad (1.66)$$

$$e_{it\omega} = e'_{it,\mathcal{H}'\omega}, \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (1.67)$$

$$q_{ikt\omega} = q'_{ikt,\mathcal{H}'\omega}, \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (1.68)$$

Constraints (1.65 - 1.68) ensure the consistency of the decisions throughout the scenario tree.

Another recent formulation considering time windows and carbon emission is developed in Kumar et al. (2015). Extensive reviews on PDP are presented in Adulyasak et al. (2015b) and Díaz-Madroño et al. (2015).

PDP is a challenging problem because it integrates two well-known and hard combinatorial problems: lot sizing and vehicle routing (Absi et al., 2014). Therefore, it requires efficient solving methods to provide solutions.

1.5 Research Gaps

Based on the works on network design (subsection 1.2.1), routing (subsection 1.2.2, disassembly process (section 1.3), few researches have investigated the integrated decisions in RSC. In particular, no research focuses on PDP-liked problem for RSC as in Section 1.4. To fulfil this gap, this dissertation has been conducted to contribute on the integration of decisions in collection and disassembly processes in RSC as depicted in Figure 4.

Based on practical and scientific points of view, this dissertation mainly contributes to support the implementation of the circular economy since it allows the decision makers to optimise their RSC. It also contributes to fulfil the gaps in the domain of decision integration in RSC. This dissertation also extends the existing researches in PDP into the context of RSC.

To achieve these contributions, an integer linear programming formulation of integrated decisions concerning both collection of single type EOL product and its disassembly process with single capacitated vehicle is provided in Chapter 2. The advantage of such an integration is pointed out through the comparison with the non-integrated optimisation of both processes under deterministic condition. To deal with large size instances of the problem that the commercial solver is commonly unable to provide the optimal solutions, some approximate methods are developed and provided in Chapter 3. A formulation based on stochastic integer linear programming as well as some solution methods are presented in Chapter 4 as a further effort to extend the formulation presented in Chapter 2 in order to propose better applicability. This stochastic formulation deals with the uncertainty corresponding to the quantity and the quality of single type EOL products, the demand of its components and multi-vehicle. Following the two formulations, another formulation dealing with collection routing and disassembly line balancing

1.5. RESEARCH GAPS

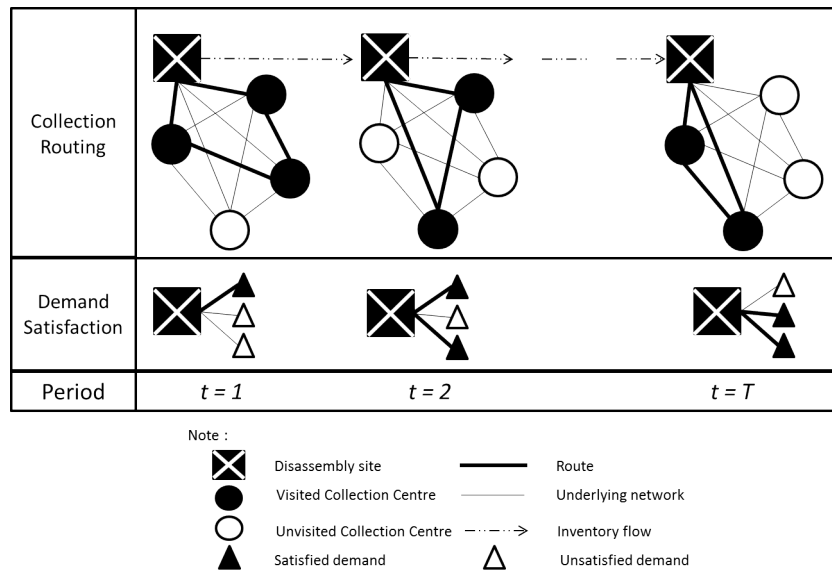


Figure 4 – Network Representations of Our Problem

problems is proposed in Chapter 5. The general conclusions of this dissertation and some prospects of future work are presented in Chapter 6.

Chapter 2

Collection-Disassembly Problem

Contents

2.1	Introduction	23
2.2	Problem Description	23
2.3	Formulations	25
2.3.1	Case I (with integration)	25
2.3.2	Case II (without integration)	26
2.4	Instance Generation	28
2.5	Numerical Analysis	29
2.6	Conclusions	34
2.7	Publications	35

2.1 Introduction

This chapter introduces the Collection-Disassembly Problem. This problem is to integrate the decisions concerning the collection of End-of-Life (EOL) products and their disassembly process. A formulation based on integer linear programming is proposed. Another formulation treating both processes separately is also introduced. Since there is no instance benchmark for this problem, few data sets are generated to test the problem. A comparison between this integrated problem and those separately optimised is provided to show the value of the integrated model.

2.2 Problem Description

The Collection-Disassembly Problem (CDP) is a version of Production-Distribution Problem (PDP) in reverse supply chain context. It integrates decisions on collection vehicle routing and disassembly lot-sizing. In this chapter, CDP under deterministic

2.2. PROBLEM DESCRIPTION

condition is provided. Since CDP is based on an integrated approach, a non-integrated approach regarding the decisions concerned in CDP is also provided to prove the benefit of such integration.

This chapter provides case I as depicted in Figure 4 (in Chapter 1) and case II as non-integrated approach of collection routing and disassembly lot-sizing focusing on EOL product.

Suppose that a single disassembly site is responsible for gathering a single type of EOL product available at dispersed collection centres. A capacitated vehicle is available for gathering the products. Once a collection centre visited, all items are picked up by the vehicle.

It is assumed that the nomenclature is known. Each item has several components $a \in A$ where each component has a quantity n_a . The collected items will be disassembled on the disassembly site in order to release the components requested for satisfying the demands. The disassembly site has a fixed capacity $DisCap$ corresponding to the cycle time of the disassembly line. The items are disassembled in order to satisfy a given demands of component. The unmet demand of components results a penalty cost for each unit CP_a . The problem is multi-period since it concerns with inventory having capacity $InvCap$. There is no salvage value or disposal cost for any leftover components. The parameters and the decision variables are provided as follows:

Parameters :

- A set of components: $a \in \{1, 2, \dots, |A|\}$
- N set of nodes: $i, j \in \{1, 2, \dots, |N|\}$
- N_c set of collection centres: $i, j \in \{2, \dots, |N|\}$
- T planning horizon: $t \in \{1, 2, \dots, |T|\}$
- n_a quantity of component a in product
- S_{it} quantity of items available at collection centre i at period t
- q_{at} demand of component a at period t
- Q vehicle capacity
- $InvCap$ inventory capacity
- $DisCap$ disassembly capacity
- CF fixed vehicle dispatch cost
- c_{ij} mileage cost from node i to j
- CD unit disassembly cost
- CH unit holding cost
- CP_a unit penalty cost of component a .

Decision variables :

x_{ijt}	$\begin{cases} 1 & \text{if } j \text{ is visited right after } i \text{ at period } t \\ 0 & \text{otherwise.} \end{cases}$
y_{it}	vehicle load after visiting i at period t
I_t	inventory level at period t
P_t	quantity disassembled at period t
SO_{at}	unmet demand of component a at period t .

2.3 Formulations

2.3.1 Case I (with integration)

The following formulation refers to CDP. It deals with the decisions on routing, inventory and disassembly.

Integer Linear Programming (ILP) model

$$\text{Min} \sum_{t \in T} \left(\sum_{j \in N_c} CF \cdot x_{1jt} + \sum_{i \in N} \sum_{j \in N} c_{ij} \cdot x_{ijt} + CH \cdot I_t + CD \cdot P_t + \sum_{a \in A} CP_a \cdot SO_{at} \right) \quad (2.1)$$

Subject to:

$$\sum_{j \in N, i \neq j} x_{ijt} \leq 1 \quad \forall i \in N_c, \forall t \in T \quad (2.2)$$

$$\sum_{i \in N, i \neq v} x_{ivt} = \sum_{j \in N, j \neq v} x_{vjt} \quad \forall v \in N, \forall t \in T \quad (2.3)$$

$$y_{it} + (Q - S_{it}) \cdot x_{1it} \leq Q \quad \forall i \in N_c, \forall t \in T \quad (2.4)$$

$$y_{it} - y_{jt} + Q \cdot x_{ijt} + (Q - S_{jt} - S_{it}) \cdot x_{jit} \leq Q - S_{jt} \quad i \neq j, \forall i, j \in N_c, \forall t \in T \quad (2.5)$$

$$I_t = I_{t-1} + \sum_{i \in N} \sum_{j \in N, i \neq j} S_{it} \cdot x_{ijt} - P_t \quad \forall t \in T; \quad (2.6)$$

$$n_a \cdot P_t + SO_{at} \geq q_{at} \quad \forall a \in A, \forall t \in T \quad (2.7)$$

$$\sum_{j \in N, i \neq j} S_{it} \cdot x_{ijt} \leq y_{it} \leq \sum_{j \in N, i \neq j} Q \cdot x_{ijt} \quad \forall i \in N, \forall t \in T \quad (2.8)$$

$$I_t \leq InvCap \quad \forall t \in T \quad (2.9)$$

$$P_t \leq DisCap \quad \forall t \in T \quad (2.10)$$

$$x_{ijt} \in \{0, 1\} \quad \forall i, j \in N, \forall t \in T \quad (2.11)$$

$$SO_{at}, y_{it}, I_t, P_t \in \mathbb{Z}^+ \quad \forall a \in A, \forall i \in N, \forall t \in T. \quad (2.12)$$

The objective function (2.1) minimises the total cost summing the costs of collection routing, holding, disassembly and penalty. The collection routing consists of the dispatch

2.3. FORMULATIONS

and mileage costs. The holding cost concerns about the quantity of products stored at inventory. The disassembly cost depends on the number of products disassembled. The penalty cost corresponds to the unmet component demands.

Constraints (2.2) state that each collection centre is visited at most once during a period. The idea of these constrained is taken from PDP formulations presented in Subsection 1.4.1. The flow balance of each collection centre is assured by constraints (2.3). Constraints (2.4) determine the load of vehicle for the first node visited. The subtour elimination (2.5) are based on lifting method proposed in Desrochers and Laporte (1991). These constraints are also adapted in PDP formulations of Bard and Nananukul (2009, 2010) in constraints (1.11). Constraints (2.6) are the inventory balance of disassembly site for all periods. Constraints (2.7) impose the demand fulfilment. Constraints (2.8 - 2.10) limit the decisions of vehicle load (adapted from constraints (1.12)), inventory and disassembly, respectively. Constraints (2.11) and (2.12) define the nature of decision variables.

As shown by its constraints, CDP also integrates two well-known and hard combinatorial problem: VRP and lot-sizing.

Constraints (2.2 - 2.5), (2.8) and (2.11) correspond to the collection of EOL products. Constraints (2.6) link between the collection and the disassembly process. Constraints (2.6), (2.10) and (2.12) correspond to the disassembly process. In the next part, each constraints are reassembled to the process they belong to.

2.3.2 Case II (without integration)

The formulations presented in this chapter is based on an approach assuming that the decisions on collection and disassembly are optimised independently. The problem is deployed into two subproblems: (i) disassembly lot-sizing and (ii) collection routing. As depicted in Figure 5, the disassembly lot-sizing problem concerns with the decisions on the quantity of EOL products disassembled for satisfying the component demands for all periods. The disassembly lot-sizing problem does not consider the available quantity of EOL products in collection centres. Based on this decision, the collection routing attempts to fulfil by gathering the products available at collection centres. The penalty cost is occurred when the demands of component are unmet. The variable $Collection_t$ is introduced to denote the quantity of products intended.

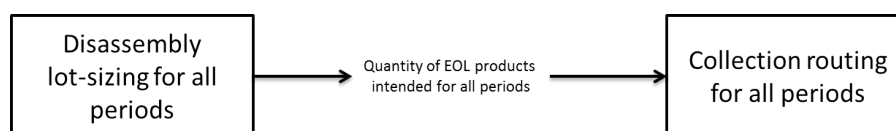


Figure 5 – Relation of Disassembly Lot-Sizing and Collection Routing

ILP of disassembly lot-sizing

$$Min \sum_{t \in T} \{CD \cdot P_t + CH \cdot I_t + \sum_{a \in A} CP_a \cdot SO_{at}\} \quad (2.13)$$

Subject to:

$$I_t = I_{t-1} + Collection_t - P_t, \forall t \in T \quad (2.14)$$

Constraints (2.7), (2.9), (2.10) and (2.12)

$$Collection_t \in \mathbb{Z}^+, \forall t \in T \quad (2.15)$$

The objective function (2.13) minimizes the total cost of disassembly, inventory and penalty. The penalty cost incurs when the capacity of disassembly site is unable to meet the demands of component. Constraints (2.14) balance the number of products in inventory for all periods. Constraints (2.15) are the nature of variable $Collection_t$.

Using the value of $Collection_t$ obtained from the previous problem, the collection routing is dedicated to yield the route of vehicle as follows:

ILP model of collection routing

$$Min \sum_{t \in T} \left\{ \sum_{i \in N} \sum_{j \in N} C_{ij} \cdot x_{ijt} + \sum_{j \in N_c} CF \cdot x_{1jt} + \sum_{a \in A} CP_a \cdot SO_{at} \right\} \quad (2.16)$$

Subject to:

$$Collection_t \geq \sum_{j \in N} \sum_{i \in N, i \neq j} S_{it} \cdot x_{ijt}, \forall t \in T \quad (2.17)$$

$$n_a \cdot \sum_{i \in N, i \neq j} S_{it} \cdot x_{ijt} + SO_{at} \geq q_{at}, \forall a \in A, \forall t \in T \quad (2.18)$$

Constraints (2.2) - (2.5), (2.8) and (2.11)

The objective function (2.16) minimises the dispatch and mileage costs corresponding to the vehicles used as well as the penalty cost emerged by the unmet component demands. Note that the penalty cost occurs when the quantity of products intended $Collection_t$ is unmet. Constraints (2.17) assure that the number of products collected is lower than $Collection_t$ for preventing any excessive mileage cost. Constraints (2.18) impose the satisfaction of component demands.

2.4 Instance Generation

Due to lack of benchmark instances available for CDP, the instances were generated in following fashion. The data sets diagram is given in Figure 6. The data set I varies the location of collection centres, the set of components A , the set of nodes N , the set of periods T , the demand of component q_{at} and the disassembly capacity $DisCap$. The data set II focuses on the quantity of EOL products available at collection centres S_{it} , the vehicle capacity Q and the starting inventory level I_0 . The data set III is used to evaluate the impact of the different costs between disassembly process and collection routing involving the unit disassembly cost CD , the unit holding cost CH , the unit penalty cost of component CP_a , the fixed vehicle dispatch cost CF and the travelling cost c_{ij} .

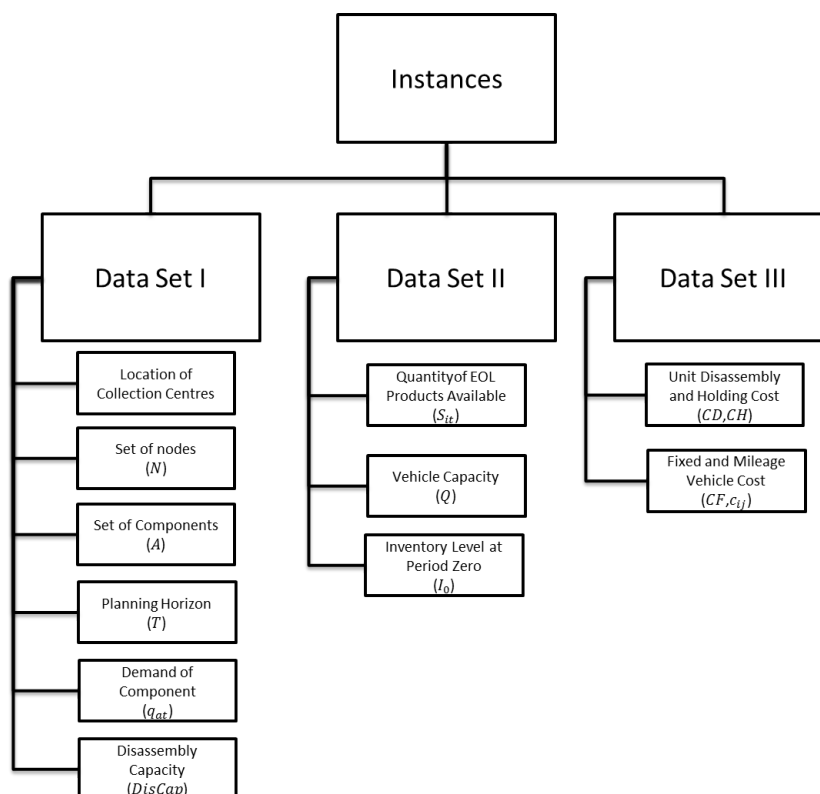


Figure 6 – Data Sets

In data set I, the collection centres' location is generated into either at random or by cluster. In the random category, the location is generated uniformly with $U(0 : 100)$ corresponding to ordinates and axis. In the cluster category, their location is uniformly generated as shown in Table 2.1. Both categories are generated twice. Initially, the

set of nodes N is generated as 25 including the depot (the disassembly site). After, we generated instances with 10 nodes using the characteristics of instances with 25. Finally, we generated instances with 5 nodes based on the characteristics of instances with 10 instances. This manner is adapted from Boudia et al. (2006). The set of components A is set to 5 and 10. The set of periods T is fixed to 5, 10 and 25. The demand of component q_{at} is generated with $U(40\%; 60\%) \cdot \mathbf{S}$ and $U(90\%; 110\%) \cdot \mathbf{S}$ where $\mathbf{S} = \frac{\sum_{i \in N_c} \sum_{t \in T} S_{it}}{N_c \cdot T}$. The disassembly site capacity $DisCap$ is relative to $\frac{\sum_{a \in A} \sum_{t \in T} q_{at}}{T}$ by 85%, 118%, 200% and infinite in which we call them as *under constrained*, *very constrained*, *constrained* and infinite, respectively. The other values are shown in Table 2.2.

In data set II, the quantity of EOL products S_{it} is generated as $U(9 : 11)$ and $U(40 : 60)$. The vehicle capacity Q is generated with 2,3 and 4 times $\frac{\sum_{i \in N_c} \sum_{t \in T} S_{it}}{T}$. The remaining parameters are provided in Table 2.3

In data set III, the value of the fixed cost of vehicle CF is 5, 10 and 25. The value of the unit disassembly cost CD is 50%, 100% and 200% times CF . The value of holding cost CH is fixed to 10% $\cdot CD$. The remaining parameters are shown in Table 2.4.

Table 2.1 – Location of Collection Centres in Cluster Category

Collection Centres	Distribution Parameters	
	Ordinate	Axis
2 nd -7 th	$U(0 : 25)$	$U(0 : 25)$
8 th -14 th	$U(75 : 100)$	$U(30 : 50)$
15 th -19 th	$U(75 : 100)$	$U(75 : 100)$
20 th -25 th	$U(0 : 25)$	$U(75 : 100)$

$U(a : b)$ indicates that the corresponding parameter was generated following uniform distribution with parameter a and b .

2.5 Numerical Analysis

The formulations were implemented in java JDK 7 using ILOG CPLEX 12.6 on a PC with processor Intel®Core™i7 CPU 2.9 GHz and 4 GB of RAM under Windows 7 Professional. The first, second and third data sets contain 488, 18 and 9 instance, respectively. The first data set contains 4 sub-data sets (122 instances per each sub-data set) and were executed within 10 minutes. The second and third data sets containing 18 and 9 instances, respectively, were executed under 100 minutes of execution.

Our findings based on the comparison between the integrated formulation (case I) and the independently solved problems (case II) are presented. The analysis on managerial factor of each interpretation is also available. TC , TDC , TCC and TPC correspond to the average difference of total cost, of total disassembly cost, of total collection cost

Table 2.2 – Parameters of Data Set I

Parameters	Value
$S_{it}, \forall i \in N_c, \forall t \in T$	$U(9:11)$
Q	$2 \cdot \mathbf{S}$
I_0	0
$InvCap$	∞
CD	10
CH	1
$CP_a, \forall a \in A$	4
CF	10

$U(a : b)$ indicates that the corresponding parameter was generated following uniform distribution with parameter a and b .
 \mathbf{S} is the average of supply of EOL products for all collection centres and all periods.

Table 2.3 – Parameters of Data Set II

Parameters	Value
A	5
N	10
T	10
$q_{at}, \forall a \in A, \forall t \in T$	$U(90\% : 110\%) \cdot \mathbf{S}$
I_0	$2 \cdot \mathbf{q}$
$DisCap$	∞
CD	10
CH	1
$CP_a, \forall a \in A$	4
CF	10
Collection Centres Location	Random

$U(a : b)$ indicates that the corresponding parameter was generated following uniform distribution with parameter a and b .
 \mathbf{S} is the average of supply of EOL products for all collection centres and all periods
 \mathbf{q} is the average of demand of component of EOL products for all component centres and all periods.

and of total penalty cost, respectively, between case I and II. For clarity, the following equation computes TC value with corresponding parameters:

$$TC = \frac{\text{average total cost}_{case II} - \text{average total cost}_{case I}}{\text{average total cost}_{case II}}$$

The other average cost differences (TDC , TCC and TPC) are calculated using similar formulas based on related costs.

Table 2.4 – Parameters of Data Set III

Parameters	Value
A	5
N	10
T	10
$S_{it}, \forall i \in N_c, \forall t \in T$	$U(9 : 11)$
q_{at}	$U(40\% : 60\%) \cdot \mathbf{S}$
$InvCap$	∞
$DisCap$	∞
I_0	0
SO_{at}	4
Collection Centres Location	Random

$U(a : b)$ indicates that the corresponding parameter was generated following uniform distribution with parameter a and b .
 \mathbf{S} is the average of supply of EOL products for all collection centres and all periods.

Data Set I According to Figure 7, the value of TC is always non zero indicating that lower cost is always obtained for case I. In other words, the integrated approach permits RSC to have better performance. Whilst the values of TDC are nearly zero showing that the number of products disassembled is almost similar between the two cases.

While TCC alternates on the axis line, TPC is near 1. It indicates that the elevation of collection cost affects the decrease on unmet demand. Henceforth, the satisfaction of customers will be elevated along with the reduction of penalty cost as long as the travelling cost is relatively cheaper.

Concerning to the collection process, we note that for higher values of number of nodes N , number of periods T and number of components A , TCC is increased as depicted in Figure 7(b), Figure 7(c) and Figure 7(d), respectively. It is natural since their elevation requests a higher number of products to be collected for avoiding higher penalty cost. Correspondingly, the value of component demand q_{at} alternates TCC proportionally. In other words, the increase of demand naturally requires a higher number of products to be collected incurring higher collection cost.

Disassembly capacity $DisCap$ has no significant influence except for instances with under constrained disassembly capacity $DisCap$ (the instances having disassembly capacity lower than the average demand of component). It can be concluded that $DisCap$ is not a sensitive parameter for influencing the result as long as its number is higher than q_{at} . Consequently, setting up disassembly system with slightly higher time cycle leads to more efficient TCC since the collection process permits optimising more products gathered.

Regarding the CPU times (in seconds) as shown in Figure 8, it is directly proportional to either N or T and inversely proportional to A and q_{at} . $DisCap$ has a particular effect since *constrained* instances require more CPU time due to the trade off between penalty

2.5. NUMERICAL ANALYSIS

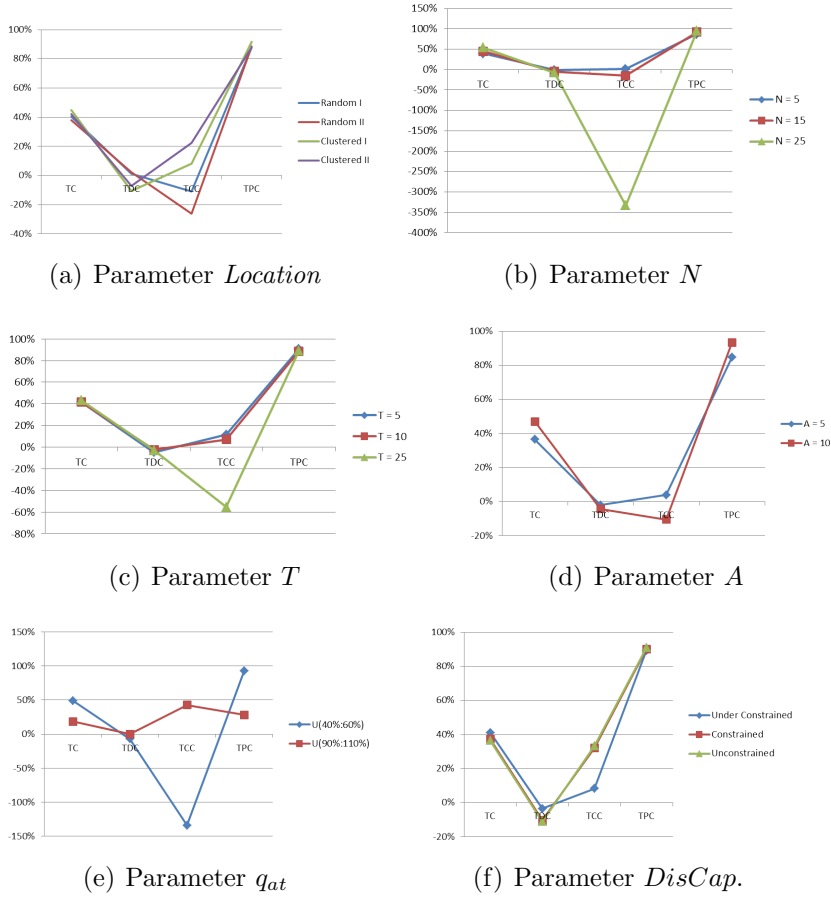


Figure 7 – Results of Data Set I

cost and collection cost.

Data Set II Corresponding to Figure 9(a) and Figure 9(b), parameter supply S_{it} affects the costs slightly rather than vehicle capacity Q . Meanwhile parameter inventory level at period zero I_0 shows that providing higher inventory leads to the decrease of TPC since the demand will be more satisfied. Consequently, it raises the efficiency on TC . Thus, the inventory of components required at period zero reduces the total cost.

According to Figure 10, the associated CPU time is proportional to S_{it} and I_0 . Since S_{it} is high, it naturally results to less collection centres visited. As a consequence, it reduces the CPU time.

Data Set III Observing Figure 11(a), the value of unit disassembly cost CD is proportional to all costs except TPC . When $CD = 50\%.CF$ as fixed vehicle dispatch cost,

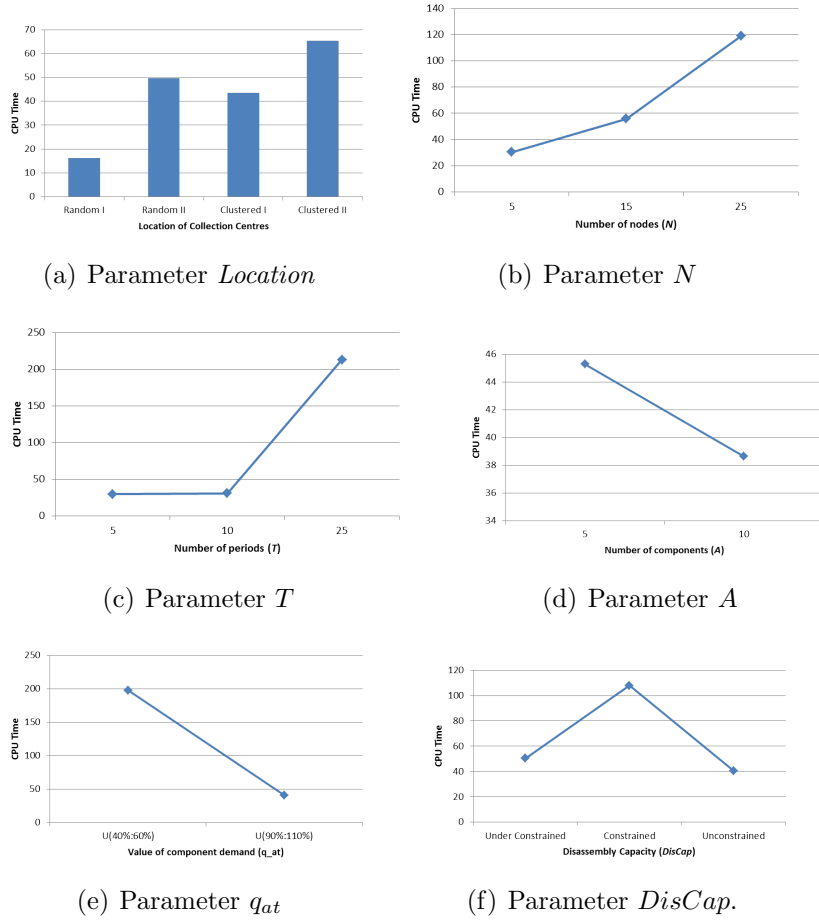


Figure 8 – CPU Times of Data Set I (in seconds)

TPC is zero showing that no difference between case I and II. Whilst, TDC is lower indicating that the disassembly process deals with less products.

Looking on Figure 11(b), a particular behaviour is revealed when $CF = 25$ since both the values of TDC and TCC are zero. In this case, CF is six times of unit penalty cost CP_a . It marks clearly that paying penalty cost allows to avoid expensive collection cost. In application, CF covers costs e.g. cost of maintenance, assurance, driver salary. Thus, the ratio between fixed vehicle has to be considered compared to the unit penalty cost.

Figure 12 shows that the CPU time declines along with the increase of CD and CF . Since CD reflects an expensive unit disassembly cost, the collection process gathers less products. To this point, it reduces the permutation of routing vehicle yielding lower CPU time. Whilst expensive fixed vehicle cost is reflected by CF . It yields less vehicles used resulting less CPU time as well.

2.6. CONCLUSIONS

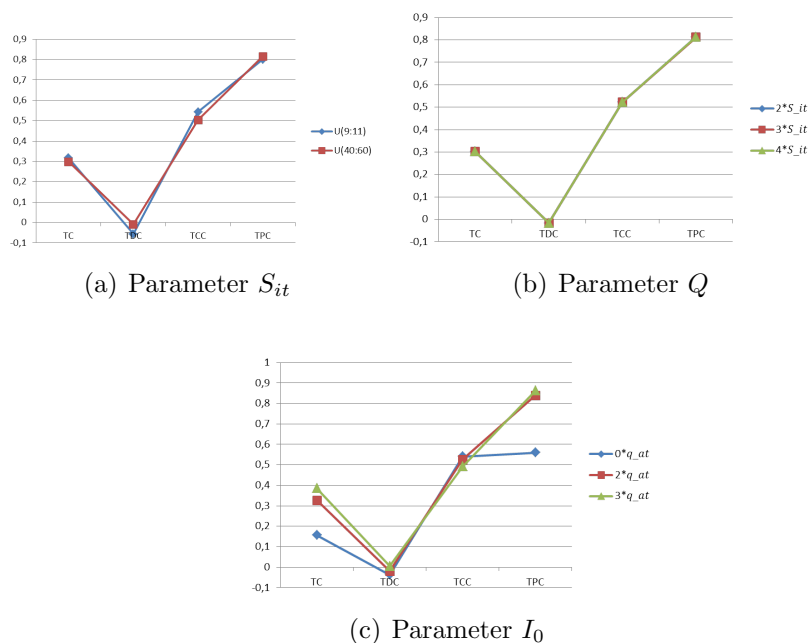


Figure 9 – Results of Data Set II

2.6 Conclusions

In this chapter, a new problem, called Collection-Disassembly Problem, incorporating collection of EOL products and their disassembly process is introduced. It is an integration of vehicle routing and lot-sizing problems in reverse supply chain. Particularly, this chapter points out the advantage of integration of the two problems by comparing it with the two problems being optimised independently.

Since there is no benchmark toward this problem, three data sets were generated. All parameters associated with the data sets have been investigated by changing their values. Accordingly, some parameters have significant impact on average total cost TC and CPU time. They are set of nodes N , planning horizon T , set of component A , demand of component q_{at} , unit disassembly cost CD , inventory level at period zero I_0 and unit fixed vehicle cost CF . The disassembly capacity $DisCap$ and fixed vehicle cost CF alter on CPU time in some instances.

As mentioned, all instances were executed in very limited CPU times. However, some were not solved optimally. Consequently, the instances of Data Set I were executed by CPLEX without any limit of CPU times. We found that the solver CPLEX is unable to obtain the optimal solutions for large size instances (those having size larger than 5 nodes, 25 periods and 5 components) under acceptable CPU times (less than 2 hours).

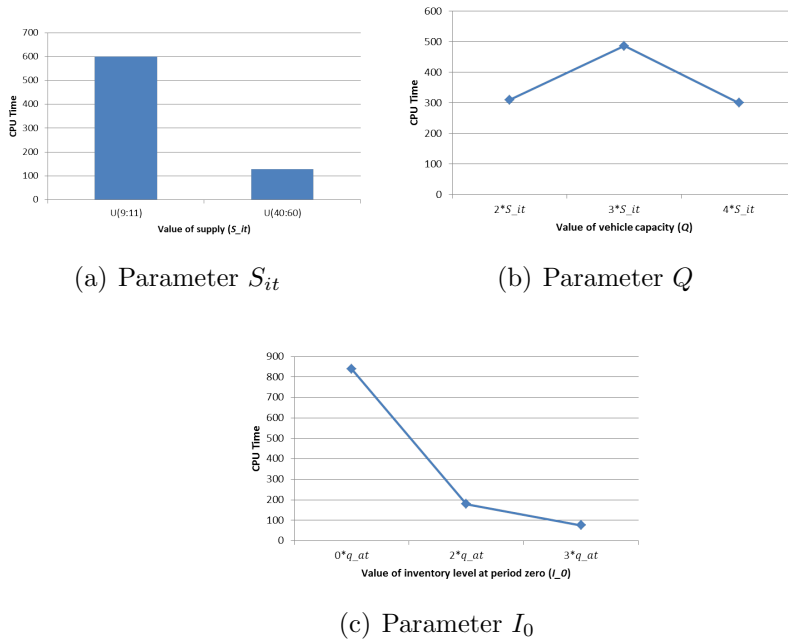


Figure 10 – CPU Times of Data Set II (in seconds)

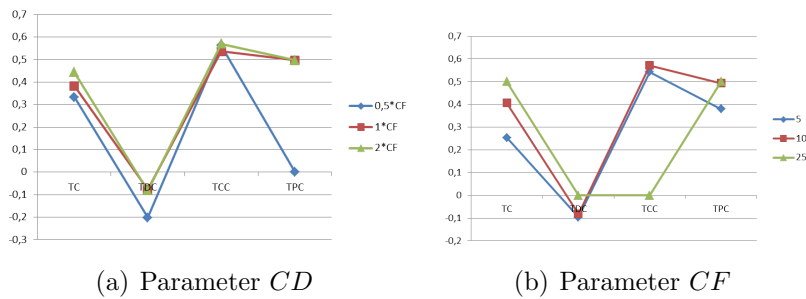


Figure 11 – Results of Data Set III

We also found that this setting also induced memory issue. Therefore, some approximate methods are proposed in the next chapter.

2.7 Publications

The results in this chapter are presented in the following articles:

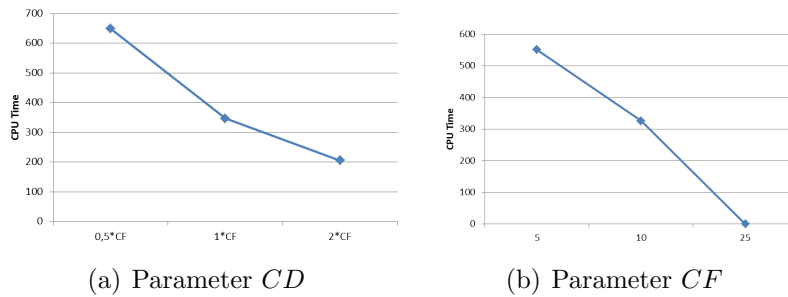


Figure 12 – CPU Times of Data Set III (in seconds)

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Collection-Disassembly Problem in Reverse Supply Chain, *International Journal of Production Economics*, vol. 183, 2017, p. 334-344. doi:10.1016/j.ijpe.2016.06.025
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Coordination of Collection and Disassembly Planning for End-of-Life Product. *IFAC-PapersOnLine*, vol. 48, no. 3, 2015, p. 76–80, doi:10.1016/j.ifacol.2015.06.061
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Dealing with the collection and the disassembly planning for simple end-of-life product, *Actes des Journées Nationales/ Doctorales MACS*, June 18th-19th, 2015, Bourges, France, 5 pages (USB)
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. La coordination de la collecte des produits en fin de vie et de leur désassemblage dans les chaînes logistiques inversées. *La 20ème journée du pôle Sciences et Techniques de la Production (STP) du GdR MaCS*, February 5th, 2015, Troyes, France
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Combining Procurement and Disassembly Decisions for End-of-Life Product. *Le 16ème congrès de la société française de recherche opérationnelle et aide à la décision (ROADEF)*. February 25th – 26th, 2015, Marseille, France.

Chapter 3

Approximate Methods for Collection-Disassembly Problem

Contents

3.1	Introduction	37
3.2	Approximate Methods	37
3.2.1	Two-Phase Iterative Heuristic	38
3.2.2	Local Search-based Heuristic	45
3.2.3	Particle Swarm Optimisation	49
3.2.4	Imperialist Competitive Algorithm	52
3.3	Numerical Experiments	55
3.3.1	Experimental Setup	55
3.3.2	Results	56
3.4	Conclusions	59
3.5	Publication	60

3.1 Introduction

In Chapter 2, it was mentioned that the commercial solver CPLEX was unable to provide optimal solutions for large size instances of CDP under acceptable CPU times (less than 2 hours). Hence, some approximate methods are required to provide (near) optimal solutions within shorter CPU times.

3.2 Approximate Methods

A number of methods has been recently developed for PDP. Their efficiency are evaluated on available data sets provided in Archetti et al. (2011), Boudia et al. (2007)

and Boudia and Prins (2009). Some methods were developed such as exact methods (Amorim et al., 2013), Branch & Price (Bard and Nananukul, 2010), Branch & Cut (Adulyasak et al., 2014b; Archetti et al., 2011), Mathematical Programming-based Heuristics (Archetti et al., 2011), Lagrangian Relaxation (Fumero and Vercellis, 1999; Solyali et al., 2009), Decomposition Heuristics (Bertazzi et al., 2005; Çetinkaya et al., 2009; Chandra and Fisher, 1994; Chen et al., 2009) and L-Shaped (Benders) Decomposition (Adulyasak et al., 2015a). Some metaheuristics and heuristics were also developed such as Tabu Search (Shiguemoto and Armentano, 2010), Genetic Algorithm (Buer et al., 1999), Greedy Randomized Adaptive Search Procedure (Boudia et al., 2007), Memetic Algorithm (Boudia and Prins, 2009), Ant Colony Optimization (Calvete et al., 2011), Adaptive Large Neighbourhood Search (Adulyasak et al., 2014a), Two-Phase Iterative Heuristics (Absi et al., 2014) and Self-Learning Particle Swarm Optimisation (Kumar et al., 2015).

Based on a comparison provided in Absi et al. (2014), Two-Phase Iterative Heuristics provides the best solutions for all available instances. There are two different heuristics proposed in Absi et al. (2014): *Iterative Method-Multi Travelling Salesman Problem* (IM-MultiTSP) and *Iterative Method-Vehicle Routing Problem* (IM-VRP). Due to its flexibility, IM-MultiTSP is adapted to deal with CDP in this chapter. Some other approximate methods are also proposed here such as:

- Local Search-based Heuristic,
- Particle Swarm Optimisation,
- Imperialist Competitive Algorithm.

Furthermore, an enhanced version of Two-Phase Iterative Heuristic are proposed. Also, an enhanced version of Particle Swarm Optimisation is proposed to provide better solutions.

Particle Swarm Optimisation and Imperialist Competitive Algorithm were carried out during our exchange programme at the Department of Industrial and Systems Engineering of Indian Institute of Technology, Kharagpur, India.

3.2.1 Two-Phase Iterative Heuristic

As described in Chapter 1, PDP is an integrated problem of vehicle routing and lot-sizing. This problem is formalised into mixed integer programming in which the decisions consist of:

1. when and how much products to produce (continuous products),
2. when to visit,
3. how much to deliver to retailers,
4. routing for each vehicle,

5. inventory level of each retailer.

The Two-Phase Iterative Heuristic is originally proposed in [Absi et al. \(2014\)](#) for dealing with PDP with multi-vehicle and single type continuous product. The idea is to decompose the problem into smaller problems and solved them iteratively. The PDP is decomposed into so called lot-sizing problem with approximate visiting costs and routing problem. Since we adapt IM-MultiTSP of this method, the routing problem corresponds to travelling salesman problem.

The lot-sizing problem with approximate visiting costs, also called as the first phase, is to deal with the first three decisions. Consequently, this phase provides the set of retailers served in each period.

Accordingly, the second phase aims to determine the vehicle route for each period. In the case of single vehicle, the second phase is a pure travelling salesman problem (TSP) since the vehicle capacity is already taken into account in the first phase.

The approximate visiting cost of a retailer is initialised by multiplying the go-return running costs and the distance between the retailer and the production facility. For the retailer visited, the savings heuristic is used to update their corresponding approximate visiting cost for all periods.

A diversification mechanism of the approximate visiting costs are required to permit the method exploring the unvisited solution space. It is simply done by multiplying the current value of the costs by the number of retailers visited throughout the planning horizon plus one. One is to avoid zero multiplication when no retailer is visited. The method is provided in Algorithm 1. As aforementioned, this method is also adapted from IM-MultiTSP.

Algorithm 1: Two-Phase Iterative Heuristic (IM-MultiTSP) for PDP

```

solution ← ∅
Initialise approximate visiting costs for all retailers and vehicles
while a stopping criterion is not met do
    while a stopping criterion is not met do
        Solve lot-sizing problem with approximate visiting costs and get the set of
        retailers served
        Construct the route of each vehicle to visit the served retailers
        Update solution (if necessary) and approximate visiting costs
    end
    Diversify approximate visiting costs
end

```

Two-Phase Iterative Heuristic for CDP

In this part, the adaptation of IM-MultiTSP into CDP is provided. The first phase consists of a lot-sizing problem considering the number of components in each product

3.2. APPROXIMATE METHODS

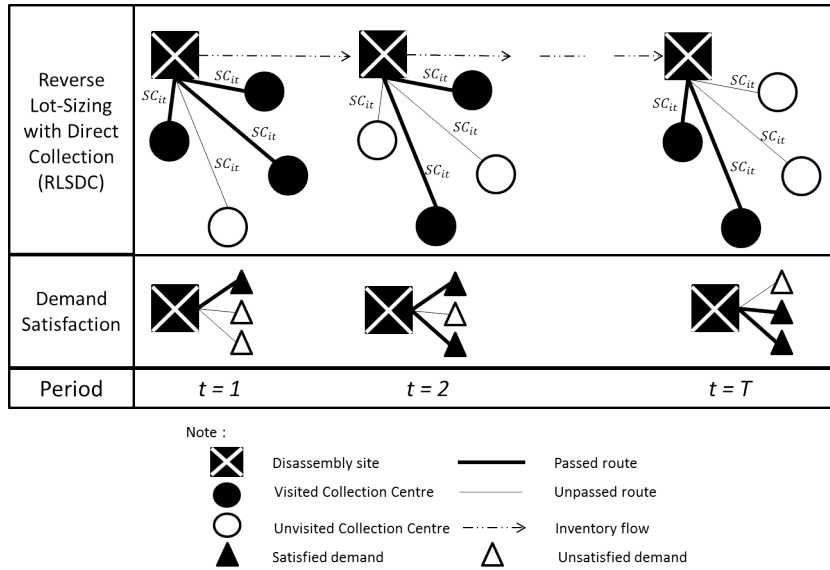


Figure 13 – Disassembly Lot-Sizing Problem with Approximate Visiting Costs

as well as the availability of EOL products in each collection centre at each period. This problem is called as *Disassembly Lot-Sizing Problem with Approximate Visiting Costs* (DLSPA VC) as depicted in Figure 13. This phase determines when and how many products to disassemble P_t , how many products to stock in the warehouse I_t , how many penalty to occur for each component SO_{at} and when to collect EOL products by visiting collection centres. The approximate visiting cost SC_{it} are also considered. The second phase constructs the route of vehicle used in each period. This phase is also TSP because we take into account one vehicle only.

As illustrated in Algorithm 2, the first phase solves the lot-sizing problem containing the quantity of EOL products disassembled P_t , the inventory level of EOL products at disassembly site I_t , the penalty unit of unmet demand SO_{at} and when to collect EOL products by visiting collection centres. Regarding the decision of EOL products collection, we introduce a binary variable γ_{it} and a non-negative variable r_{it} , $\forall i \in N_c, \forall t \in T$, denoting whether a collection centre i is visited at period t and number of EOL products collected from collection centre i at period t , respectively. A vector consisting of these variables is denoted as *solution*.

Instead of using mileage cost c_{ij} and fixed vehicle dispatch cost CF , an approximate visiting cost SC_{it} , $\forall i \in N_c, \forall t \in T$ is introduced. This cost has a prominent role to connect between two phases since it contains an information regarding the cost occurred of visiting a node in a period. This cost is updated at each step of the method. The problem is formalised as follows:

Disassembly Lot-Sizing Problem with Approximate Visiting Costs (DLSPAVC)

$$\text{Min} \sum_{t \in T} (CH \cdot I_t + CD \cdot P_t + \sum_{a \in A} CP_a \cdot SO_{at} + \sum_{i \in N_c} SC_{it} \cdot \gamma_{it}) \quad (3.1)$$

Subject to:

$$(2.7), (2.9), (2.10)$$

$$I_t = I_{t-1} + \sum_{i \in N_c} r_{it} - P_t \quad \forall t \in T; \quad (3.2)$$

$$r_{it} = S_{it} \cdot \gamma_{it} \quad \forall i \in N_c, \forall t \in T; \quad (3.3)$$

$$\sum_{i \in N_c} r_{it} \leq \min \left\{ Q; \max_a \left\{ \frac{\sum_{t'=t}^T q_{at'}}{n_a} \right\} \right\} \quad \forall t \in T; \quad (3.4)$$

$$\gamma_{it} \in \{0, 1\} \quad \forall i \in N_c, \forall t \in T; \quad (3.5)$$

$$SO_{at}, I_t, P_t, r_{it} \in \mathbb{Z}_{\geq 0} \quad \forall a \in A, \forall i \in N_c, \forall t \in T; \quad (3.6)$$

The objective function (3.1) minimises the total cost consisting holding cost, disassembly cost, penalty cost and visiting cost. As previously, constraints (2.7) impose the satisfaction of component demands. The capacity limitation regarding the inventory and disassembly is denoted by constraint (2.9) and (2.10). Constraints (3.2) balance the inventory between periods using variable r_{it} . Constraints (3.3) enforce that if node i is visited, the value of r_{it} is equal to the number of EOL products available at collection centre i . Constraints (3.4) state that the sum of r_{it} at period t is limited by the minimum value between the capacity of vehicle and the biggest remaining demands among components. These constraints are adapted from constraints (1.4), (1.23) and (1.51) in Subsection 1.4.1. Constraints (3.5) and (3.6) imposes the nature of the decision variables.

Initially, the value of SC_{it} is fixed using $c_{0i} + c_{i0}$ to enforce the first phase to serve nearby collection centres to the disassembly site since it imposes low transportation cost. Accordingly, the first phase is solved and we get the value of decision variables consisting P_t , I_t , SO_{at} , γ_{it} and r_{it} denoted as *solution* in Algorithm 2. Using the value of γ_{it} , the set of collection centres visited in each period is obtained. Thus, the routing problem is solved using the Lin-Kernighan Heuristic (LKH) proposed in Lin and Kernighan (1973). After the route in each period constructed, the value of SC_{it} is updated, consecutively.

To update SC_{it} , the following procedure is used. Suppose that $route_t$ is the route of period t constructed by solving *Routing Problem* in the second phase. For $t \in T$ and $i \in route_t$, let denote i^- and i^+ as the predecessor and the successor of node i in $route_t$. For $t \in T$ and $i \notin route_t$, let Δ_{it} as the cheapest insertion of node i into $route_t$. For each step, SC_{it} is updated using Algorithm 3.

3.2. APPROXIMATE METHODS

Algorithm 2: Two-Phase Iterative Heuristic for CDP

```
solution  $\leftarrow \emptyset$ 
Initialise  $SC_{it}, \forall i \in N_c, t \in T$ 
while stopping criterion 3 is not met do
  while stopping criterion 2 is not met do
    while stopping criteria 1 are not met do
      • 1stPhase: Solve DLSPAVC and get  $\gamma_{it}, \forall i \in N_c, t \in T$ 
      • 2ndPhase: Solve Routing Problem
      • Update sol (if necessary) and  $SC_{it}$ 
    end
    Diversify  $SC_{it}$ 
  end
  Multi-start procedure:  $SC_{it} = \rho_{it} \cdot (c_{0i} + c_{i0}), \forall i \in N_c, t \in T$ 
end
```

Algorithm 3: Update of Approximate Visiting Costs SC_{it}

```
for  $\forall i \in N_c, \forall t \in T$  do
  if  $i \in route_t$  then
    |  $SC_{it} \leftarrow c_{i-i} + c_{ii+} - c_{i-i+};$ 
  else
    |  $SC_{it} \leftarrow \Delta_{it};$ 
  end
end
```

This method uses two types of diversification mechanism: the multi-start procedure and the update diversification. The multi-start procedure initialises the value of SC_{it} through multiplication with a random value, ρ_{it} . Its value is drawn between $[0.0, 1.5]$ as used in [Absi et al. \(2014\)](#). Thus, the approximate visiting cost SC_{it} is set to $\rho_{it} \cdot (c_{0i} + c_{i0})$. The update diversification aims to reject a periods having high number of retailers visited. It helps the method moving to the solution space that is not explored recently. For each t , the approximate visiting cost SC_{it} is multiplied by the number of retailers served plus one. One is kept to avoid zero multiplication.

After testing this method to few instances, we found that it converges rapidly but offers insufficient optimality gaps compared to CPLEX. Hence, some enhancements are required to improve the performance of this method.

Two-Phase Iterative Heuristic with Enhancements for CDP

Based on our experience, reducing the number of periods served in the first phase allows to end up with better optimality gaps particularly for the very first iterations. However, we found that the associated formulation requires longer CPU time. Henceforth, an adaptive steps is added to Algorithm 2 in order to accommodate these enhancements as depicted in Algorithm 4.

DLSPAVC II attempts to find a solution reducing the number of periods served in DLSPAVC. A period is "served" when a vehicle is dispatched in that period. In this formulation, a parameter Z is introduced to denote the number of periods served in the first phase. The value of Z is based on the value of γ_{it} obtained by solving DLSPAVC. Additionally, a binary variable z_t is utilised in DLSPAVC II denoting whether period t is served or not. DLSPAVC II is formalised as follows:

Disassembly Lot-Sizing Problem with Approximate Visiting Cost II (DLSPAVC II)

$$\text{Min (3.1)}$$

Subject to:

$$(2.7), (2.9), (2.10), (3.2)–(3.6)$$

$$\sum_{i \in N_c} \gamma_{it} \leq N_c \cdot z_t \quad \forall t \in T \quad (3.7)$$

$$\sum_{t \in T} z_t \leq Z - 1 \quad (3.8)$$

$$z_t \in \{0, 1\} \quad \forall t \in T \quad (3.9)$$

The values of SC_{it} in this formulation are identical to those used in DLSPAVC. Constraints (3.7) imposes that z_t is equal to one if at least one collection centre is visited at

3.2. APPROXIMATE METHODS

period t . The value of Z is fixed according to the value of γ_{it} obtained from DLSPAVC. Correspondingly, constraints (3.8) force the number of periods served is lower than the one obtained in DLSPAVC. Constraint (3.9) denotes the nature of z_t .

Algorithm 4: Two-Phase Iterative Heuristic with Enhancements for CDP

```

solution ← ∅
Initialise  $SC_{it}, \forall i \in N_c, t \in T$ 
while stopping criterion 3 is not met do
    while stopping criterion 2 is not met do
        while stopping criterion 1 is not met do
            FIRST STEP
            • 1st Phase: Solve DLSPAVC and get  $\gamma_{it}, \forall i \in N_c, \forall t \in T$ 
            • 2nd Phase: Solve Routing Problem
            • Update sol (if necessary) and  $SC_{it}$ 
            Generate a random value Rand from 0 to 1
            if  $Rand \leq Prob$  then
                SECOND STEP
                • 1st Phase: Solve DLSPAVC II and get  $\gamma_{it}, \forall i \in N_c, \forall t \in T$ 
                • 2nd Phase: Solve Routing problem
                • Update sol (if necessary) and  $SC_{it}$ 
            else
                |  $Prob = Prob/2$ 
            end
        end
        Diversify  $SC_{it}$ 
    end
    Multi-start procedure:  $SC_{it} = \rho_{it} \cdot (c_{0i} + c_{i0}), \forall i \in N_c, t \in T$ 
end

```

The first phase gives information regarding the set of collection centres to visit in each period. Thus, the routing problem becomes TSP and solved using LKH (Lin and Kernighan, 1973).

To prevent the CPU time issue caused by solving DLSPAVC II, a random value $Rand$ is generated and compared with a predetermined value $Prob$. Initially, the value of $Prob$ is set as 1.

The three stopping criteria in Algorithms 2 and 4 are as follows:

- Stopping criteria 1 : standard deviation of the last ten fitness values, maximum number of iterations and CPU time are less than 5 %, 100 and 7200 seconds, respectively,

- Stopping criteria 2 : maximum number of iterations of diversification mechanism is 5,
- Stopping criteria 3: maximum number of iterations of multi-start procedure is 5.

Following the second step showed in Algorithm 4, our contribution regarding the Two-Phase Iterative Heuristic (Absi et al., 2014) is three folds: (i) the addition of the second step into the method and (ii) the use of probability reduction corresponding to the additional step. The first contribution lies on the reduction of the number of periods in which the collection process occurs. The additional constraints are to prevent a certain number of periods of collection based on the first phase. The second contribution is to reduce the CPU times required by the added step.

3.2.2 Local Search-based Heuristic

This method is essentially to employ local search operators in order to improve the initial solution. The operators are applied to the routing solution. A random diversification mechanism is added to help the method to exit from local optima solutions.

Routing Representation

Since our problem differs from classical routing problem in the sense of visiting the set of nodes partially, a particular solution representation were utilised for the routing and the decision to visit a node. A variable $route_t$ is introduced as a complete route in which that all nodes are considered at period t . It is started and ended by the depot. Having the same size as $route_t$, a variable $route - bin_t$ contains binary numbers representing the choice of visiting corresponding node. Figure 14 depicts this solution representation.

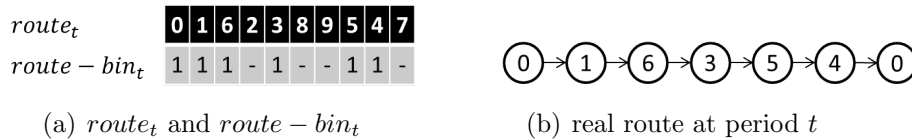


Figure 14 – Solution Representation at Period t

Local Search Mechanisms

The local search operators employed are 1-1 Exchange (Figure 15), 2-Opt (Figure 16) and 1-0 Exchange (Figure 17). Through all iterations, those are applied in both $route_t$ and $route - bin_t$ with following mechanism:

- **Change** $route_t$, **Keep** $route - bin_t$

As depicted in Figure 18, this fashion changes $route_t$ only by applying all the operators consecutively.

3.2. APPROXIMATE METHODS

- **Change route – bin_t, Keep route:**
It applies all operators into route – bin_t as depicted in Figure 19.
- **Keep route_t and route – bin_t, Change their orders:**
It attempts to change the order of route_t and route – bin_t simultaneously as shown in Figure 20.

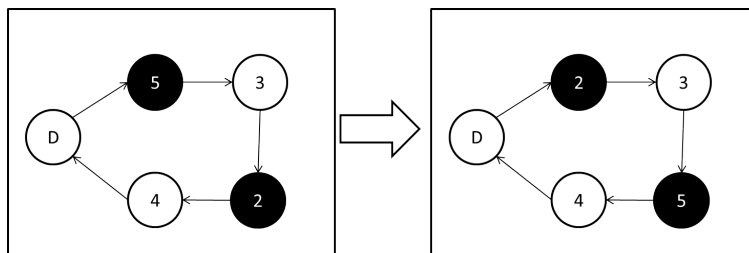


Figure 15 – 1-1 Exchange Operator

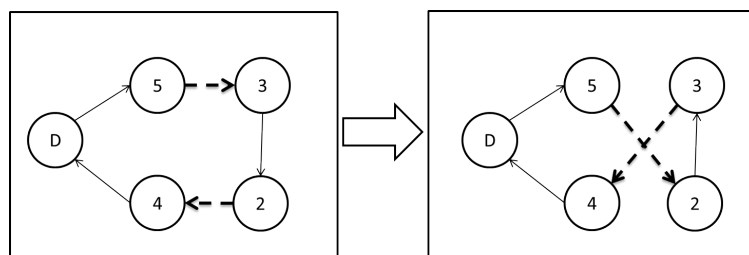


Figure 16 – 2-Opt Operator

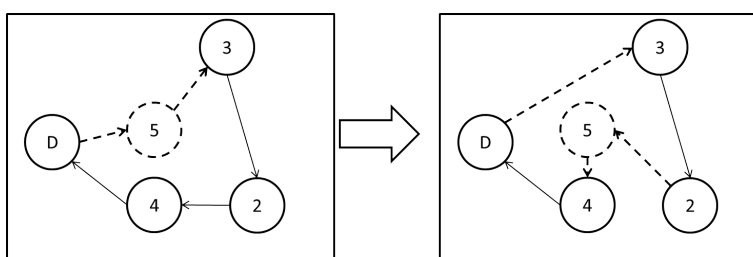


Figure 17 – 1-0 Exchange Operator

The color used in Figures 18 - 20 is to ease the reader observing the shift of each element in route_t and route – bin_t.

Figure 21 provides the random diversification mechanism which simply generates a random route_t and route – bin_t after using the local search operators.

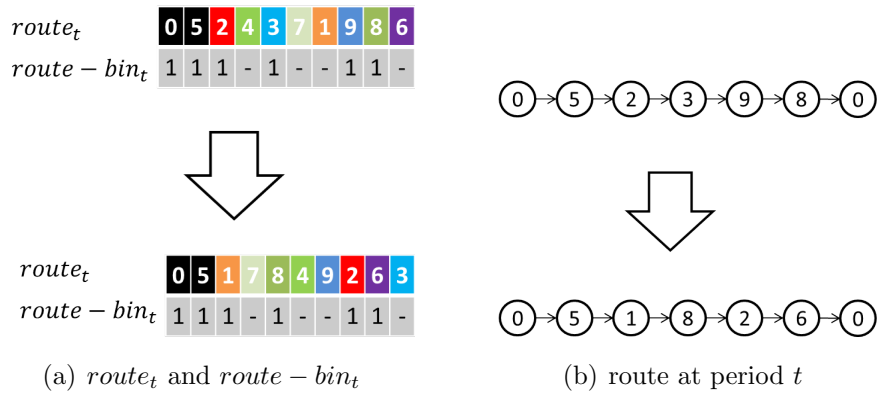


Figure 18 – Change $route_t$, Keep $route - bin_t$

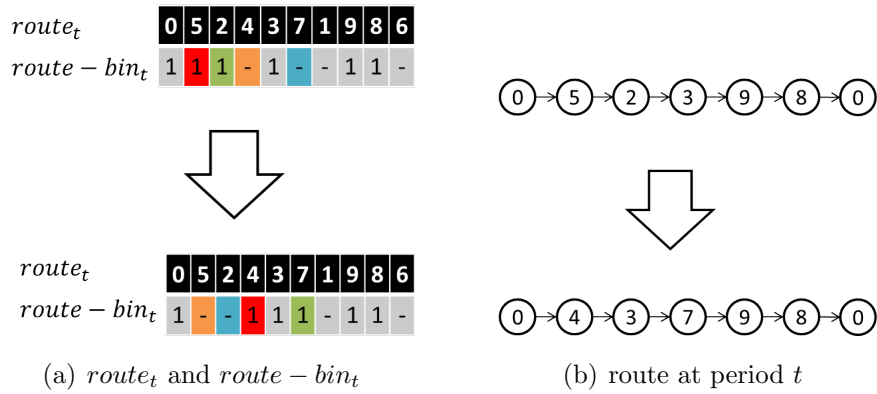


Figure 19 – Change $route-bin$, Keep $route$

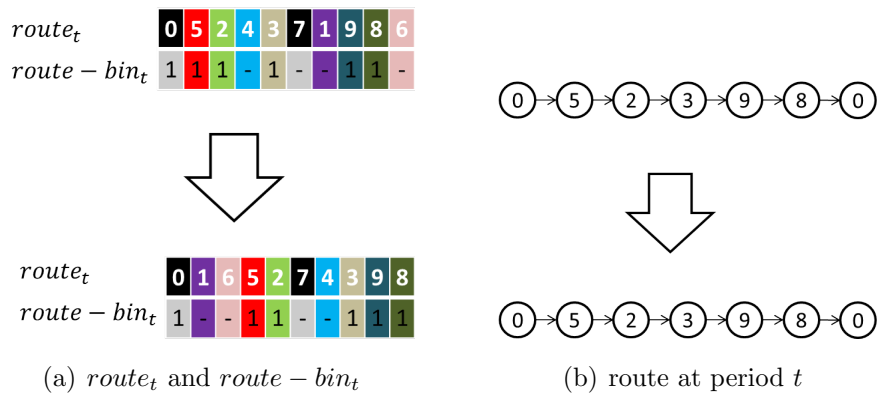


Figure 20 – Keep $route$ and $route-bin$, Change Orders

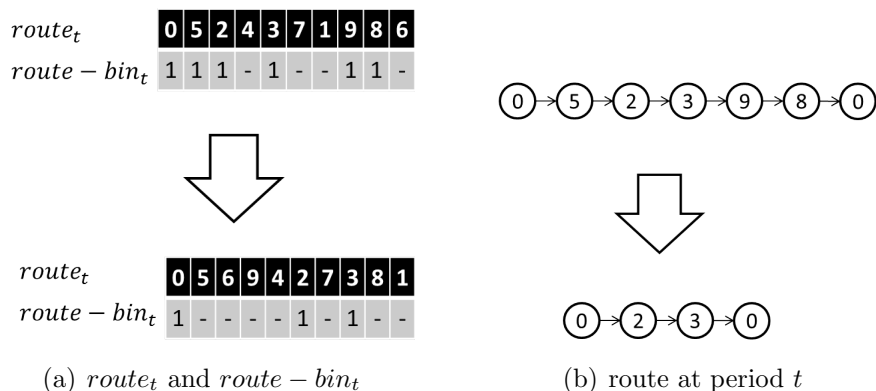


Figure 21 – Diversification

Description

As depicted in Algorithm 5, the method starts by initialising the decision variables (detail in 3.2.2). Based on this initial solution, all proposed local search operators are applied to improve the solution. We intensify the diversification mechanism after several iterations by creating large number of diversifications.

The stopping criteria are the maximum number of iterations and the standard deviation of the last 1000 iterations' fitness values. The fitness value is obtained by summing the objective value of CDP value as well the penalty incurred by constraint violations.

Algorithm 5: Local Search-based Heuristics

Initialisation

while a stopping criterion of standard deviation is not met **do**

 Change $route$ and keep $route - bin$

 Change $route - bin$ and keep $route$

 Keep both $route$ and $route - bin$ but change the order

 Diversification mechanism

if the number of diversification is performed less than a predetermined value **then**

 | Intensify the diversification mechanism

end

 Update decision variables if necessary

end

Initialisation

This initialisation is described in Algorithm 6. Since our problem concerns only a single type of product having several unique components, we can easily found $requirement_t$

denoting how many products needed at period t as follows:

$$requirement_t = Max_{a \in A} \frac{q_{at}}{n_a} \quad (3.10)$$

Subsequently, $route_t$ is constructed using nearest-neighbour heuristics. After, $route - bin_t$ is generated randomly to determine whether a node is visited or not. Correspondingly, x_{ijt} and y_{it} are updated.

$Collection_t$ denotes how many products collected at period t . It is calculated as follows:

$$Collection_t = \sum_{i,j \in N, i \neq j} S_{it} \cdot x_{ijt} \quad (3.11)$$

P_t , I_t and SO_{at} are calculated as follows:

$$P_t = rand \cdot Min\{requirement_t; Collection_t + I_{t-1}; Q\} \quad (3.12)$$

$$I_t = I_{t-1} + \sum_{i \in N} \sum_{j \in N, i \neq j} S_{it} \cdot x_{ijt} - P_t \quad (3.13)$$

$$SO_{at} = q_{at} - n_a \cdot P_t \quad \forall a \in A \quad (3.14)$$

$rand$ is a random value between 0 and 1.

All these steps are operated for all periods. Having the values of all decision variables, the fitness and objective values are consecutively calculated.

Algorithm 6: Initialisation of Local Search-based Heuristic

```

for  $\forall t \in T$  do
    Calculate  $requirement_t$ 
    Construct  $route_t$  using nearest neighbour heuristics
    Determine  $route - bin_i$  randomly
    Update  $x_{ijt}$  and  $y_{it}$ 
    Calculate  $Collection_t$ 
    Update  $P_t$ ,  $I_t$  and  $SO_{at}$ 
end
Calculate fitness and objective values

```

3.2.3 Particle Swarm Optimisation

First introduced in [Kennedy \(1995\)](#), Particle Swarm Optimisation mimics the behaviour of social organisms such as bird flock and fish school. It is found that a behaviour

3.2. APPROXIMATE METHODS

of an organism within the group is influenced by its own behaviour as well as the group the organism belongs to.

In PSO, a solution of problem is represented as a multi-dimensional position of particles. Gathering with the other particles, they forms a swarm and "collaborate" to find the best solution. Each particle $\kappa \in MaxParticle$, moves based on its previous position and its velocity denoted as $velocity_{\kappa}$. The velocity is influenced by the cognitive and social information of the swarm. The cognitive information $Pbest_{\kappa}$ represents the best solution of particle κ . The social information, $Gbest$, is the best solution among all $Pbest_{\kappa}$ for all particles.

In this part, we adapt this method into CDP. A particle κ contains the value of all decision variables: routing decision $x_{\kappa ijt}$, load of vehicle $y_{\kappa it}$, quantity of EOL products disassembled $P_{\kappa t}$, inventory level of EOL products $I_{\kappa t}$ and quantity of unmet demand $SO_{\kappa at}$. We use the routing solution representation as depicted in Figure 14 by adding the particle index. Thus, we have $route_{\kappa t}$ and $route - bin_{\kappa t}$.

Initialisation

The method in Algorithm 7 shows the initialisation step. For each particle κ and period t , we construct $route_{\kappa t}$ randomly in which it starts and ends by the depot. Having the same size, $route - bin_{\kappa t}$ is generated randomly using binary value indicating whether the corresponding node is visited or not. Accordingly, the values of $x_{\kappa ijt}$ and $y_{\kappa it}$ can be calculated. We use the same procedure as described in 3.2.2 where we remove random value in equation (3.12). This procedure is operated for all $t \in T$ and for all $\kappa \in MaxParticle$.

Algorithm 7: Initialisation of Particle Swarm Optimisation

```

for  $\forall \kappa \in MaxParticle$  do
  for  $\forall t \in T$  do
    Construct  $route_{\kappa t}$  randomly
    Determine  $route - bin_{\kappa t}$  randomly
    Update  $x_{\kappa ijt}$  and  $y_{\kappa it}$ 
    Calculate  $requirement_{\kappa t}$  and  $Collection_{\kappa t}$ 
    Update  $P_{\kappa t}$ ,  $I_{\kappa t}$  and  $SO_{\kappa at}$ 
  end
  Update  $Pbest_{\kappa}$ 
end
Update  $Gbest$ 

```

Description

The main procedure is provided in Algorithm 8. If the stopping criteria is unmet, the method continues to iterate. Similar to the prior methods, this method has to stop if the standard deviation of several last $Gbests$ is less than a pre-determined value.

For each iteration in particle κ at period t , the variable $random-number_{\kappa it}$ has to be initiated. It represents decimal values obtained from the division of each element of $route_{\kappa t}$ by its cardinality as illustrated in Figure 22.

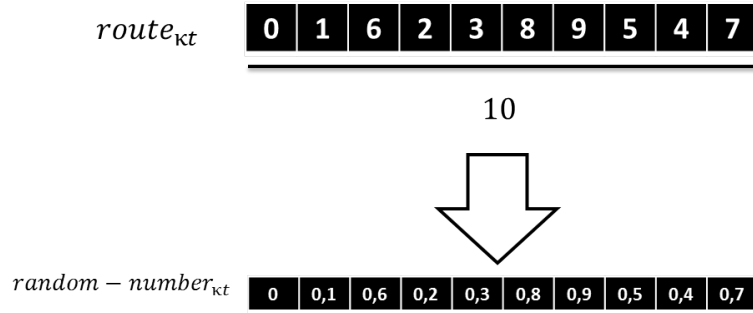


Figure 22 – Representation of Variable $random-number_{\kappa t}$

Subsequently, the variables $velocity_{\kappa it}$ and $random-number_{\kappa it}$ need to be updated using equations (3.15) and (3.16). Note that, the symbol " means the previous iteration and ' is the current iteration. For initialisation, $velocity_{\kappa it}$ is generated randomly between 0 and 1. The variables c_1 and c_2 are the acceleration of a particle to move in a single iteration. The random values $rand_1$ and $rand_2$ are between 0 and 1.

$$\begin{aligned}
 velocity''_{\kappa it} = & w \cdot velocity'_{\kappa it} + c_1 \cdot rand_1 \cdot (Pbest'_{\kappa} - random-number'_{\kappa it}) \\
 & + c_2 \cdot rand_2 \cdot (Gbest - random-number'_{\kappa it}) \quad (3.15)
 \end{aligned}$$

$$random-number''_{\kappa it} = random-number'_{\kappa it} + velocity''_{\kappa it} \quad (3.16)$$

The inertia PSO velocity update is used and denoted as w . It is calculated using equation (3.17). The maximum and minimum inertia are denoted as w_{max} and w_{min} .

$$w = w_{max} - \frac{w_{max} - w_{min}}{MaxIter} \cdot t \quad (3.17)$$

Consecutively, $random-number_{\kappa it}$ is sorted into ascending order to construct $route_{\kappa t}$. Correspondingly, $random-bin_{\kappa t}$ is randomly generated and follows the same procedure as in Algorithm 7.

Algorithm 8: Particle Swarm Optimisation

```

Initialisation;
while a stopping criterion is not met do
  for  $\forall \kappa \in MaxParticle$  do
    for  $\forall t \in T$  do
      Generate random - number $_{\kappa it}$ ,  $\forall i \in N$ 
      Update velocity $_{\kappa it}$  and random - number $_{\kappa it}$ ,  $\forall i \in N$ 
      Sort random - number $_{\kappa it}$ ,  $\forall i \in N$  into ascending order
      Construct route $_{\kappa t}$ 
      Determine route - bin $_{\kappa i}$  randomly
      Update  $x_{\kappa ijt}$ ,  $y_{\kappa it}$ ,  $P_{\kappa t}$ ,  $I_{\kappa t}$  and  $SO_{\kappa at}$ 
    end
    Update  $Pbest_{\kappa}$  and corresponding decision variables if necessary
  end
  Update  $Gbest$  and corresponding decision variables if necessary
end

```

If it is necessary, update $Pbest_{\kappa}$ as the best solution of particle κ until the current iteration. $Gbest$ is updated if the current iteration provides better solution than the previous one.

We also use three local search operators depicted in Figures 15, 17 and 16 in order to enhance the performance of this method. Henceforth, this enhancement is called Particle Swarm Optimisation with Local Search Enhancement.

3.2.4 Imperialist Competitive Algorithm

Imperialism is an ideology to extend the power and influence of a country towards the others. Initially, it aims to take control over another country in order to get access on its resources. Otherwise, it is to prevent the opponent imperialists from taking the possession.

Imperialist Competitive Algorithm is firstly introduced in [Atashpaz-Gargari and Lucas \(2007\)](#). It mimics some ideas of the imperialism in which each country represents a solution. Initially, a population having n countries is generated denoted as $nPop$. The power of a country is determined by its fitness value compared to the others. As mentioned, the fitness value is obtained by summing the objective value of CDP value as well the penalty incurred by constraint violations. In a minimisation problem, the imperialists are chosen as the countries having the lowest fitness values denoted as $f(imperialist)$. After, the other non-imperialist countries (called colonies) are assigned to the imperialists randomly and their fitness values are denoted as $f(colony)$.

The assimilation process is operated by implementing some improvements to some chosen colonies. The revolution process begins by comparing the power of each colony

of an empire towards its imperialist. If the power of the imperialist $f(imperialist)$ is less than a colony $f(colony)$, the corresponding colony becomes an imperialist. The total power of an empire denoted as TC_n is calculated by summing the countries' fitness values associated with the empire. The imperialistic competition is carried out by putting the weakest colony (those with biggest $f(colony)$ in the competing empire) into one of stronger empires. If there is an empire having no colony, this empire and its imperialist will be eliminated and assigned as a colony to another empire.

The global war begins by generating another population with n countries. The existing population is combined with the new one and stored into a temporary archive. After all countries in this archive sorted based on their power, a new population is created by conserving only the first n countries. This method is presented in Algorithm 9. The following parts are the detail implementation of this method into CDP.

Algorithm 9: Imperialist Competitive Algorithm

Initialisation

- Set Parameters
- Produce Initial Countries as pop_i with size $nPop$
- Evaluate fitness of each country
- Form the initial empires

while *MaxDecades is not met* **do**
 Assimilation
 Revolution
 Imperialistic Competition
 Global War
end

Initialisation

The number of countries $nPop$ and the initial number of empires $nEmpires$ are initialised. It is also necessary to determine the number of decades $NumDecades$ indicating the maximum iterations, the revolution rate $RevolutionRate$, the assimilation coefficient $AssimilationCoefficient$, the assimilation angle coefficient $AssimilationAngleCoefficient$, the reaction factor related to the total cost of empire ζ , the damp ratio $DampRatio$, the uniting threshold $UnitingThreshold$, the factor to prevent the weakest empire to have a probability equal to zero $zarib$ and a factor denoted as α which represents an importance of a mean minimum compare to the global minimum.

A country, as a representation of a solution, is forged by initialising the routes at period t . Accordingly, the quantity collected is calculated. The quantity disassembled P_t is the minimum value of inventory level at previous period I_{t-1} , the quantity collected

3.2. APPROXIMATE METHODS

at period t , the disassembly capacity and the demand of all components $\sum_{a \in A} q_{at}$. The values of inventory level at period t and unmet demands are determined consecutively.

After generating the initial countries pop , the fitness value is evaluated by summing the objective function and the penalties resulted from constraints violation, if any.

The initial empires are constructed by choosing $nEmpires$ countries having the lowest fitnesses as the initial imperialists. The others are assigned randomly to the imperialists. The initialisation step follows Algorithm 10.

Algorithm 10: Initialisation of Countries

```
for  $\forall Pop \in nPop$  do
  for  $\forall t \in T$  do
    Construct a feasible route randomly,  $x_{ijt}, \forall i, j$ 
    Calculate  $\sum_{i \in N} \sum_{j \in N, i \neq j} S_{it} \cdot x_{ijt}$ 
    Calculate  $P_t = \min(I_{t-1} + \sum_{i \in N} \sum_{j \in N, i \neq j} S_{it} \cdot x_{ijt}; DisCap; \sum_{a \in A} q_{at})$ 
    Calculate  $I_t = I_{t-1} + P_t + \sum_{i \in N} \sum_{j \in N, i \neq j} S_{it} \cdot x_{ijt}$ 
    Calculate  $SO_{at} = n \cdot P_t - q_{at}, \forall a \in A$ 
  end
end
```

Assimilation

Some colonies of an empire are chosen randomly to be assimilated. This step is carried out by re-generating the values of $x_{ijt}, \forall i, j, t$. It mimics the concept of neighbourhood search. Consecutively, the position of the imperialist is challenged by an assimilated colony or not. If the fitness of any colony is less than the imperialist, the colony and the imperialist exchange their position.

Revolution

A revolution is performed by generating some new countries in a selected empires. These new countries are sorted with the old ones based on their fitness. Finally, only the first $nPop$ countries are considered for the next step.

Imperialistic Competition

The total power TC_n of all empires, as sum of all fitness values of countries belong to the empire, is calculated and compared against the others. the weakest colony of the weakest empire (with highest TC_n) is assigned into another empire. If this empire has no colony, its imperialist is assigned as a new colony to another empire.

Global War

The global war is performed by generating countries with size $nPop$. These new countries are assigned randomly to each remaining empires. For each empires, the new generated countries are sorted with the old ones also based on their fitness. After, the empire contains only colonies with the same size.

3.3 Numerical Experiments

In this section, we assess the performance of CPLEX and our proposed methods. First, we describe about the experimental setting of CPLEX execution and the gap calculation. Second, the detail results are presented.

3.3.1 Experimental Setup

All formulations and algorithms (except Imperialist Competitive Algorithm) were implemented in Java using Concert Technology and were solved with IBM CPLEX 12.6 on a PC with processor Intel®Core™i7 CPU 2.9 GHz and 4 GB of RAM under Windows 7 Professional. The Imperialist Competitive Algorithm was implemented in MATLAB R2014a using the same PC.

In these experiments, we regenerate the three data sets (I, II and III) by following the procedure described in Section 2.4 which consist of 432, 18 and 9 instances, respectively. All instances were solved with CPLEX. In order to avoid memory issues with CPLEX and obtain its lower bounds, the following solution procedure was adopted. Initially, all instances were solved under 2 hours of CPLEX execution. The instances with memory issues were resolved under 30 minutes of CPLEX execution. Those with persistent memory issues were resolved under 10 minutes of CPLEX execution.

Table 3.1 – Global Average Gaps (in %)

Data Set	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
I	7.8 ⁽¹⁷²⁾	16.5	9.0 [‡]	33.5	40.6	40.5	33.4
II	0.0	3.4	1.5 [‡]	18.2	11.6	11.8	42.4
III	0.9 ⁽¹⁾	8.8	1.9 [‡]	18.6	17.0	20.5	79.9
Average	2.9	9.5	4.2	23.5	23.1	24.2	51.9

(-) indicates number of instances that were not solved optimally. Note that Data Set I, II and III consist of 432, 18 and 9 instances, respectively

‡ indicates the corresponding method provides the best optimality gap among the other approximate methods.

Some abbreviations are used to simplify the representation of the results:

- TPIH : Two-Phase Iterative Heuristic
- ETPIH : Two-Phase Iterative Heuristic with Enhancements

3.3. NUMERICAL EXPERIMENTS

Table 3.2 – Global Average CPU Times (in seconds)

Data Set	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
I	1712.8	5.5	14.4	101.1	182.1	186.8	165.8
II	427.6	5.0	9.8	33.6	54.9	53.5	77.8
III	1214.6	4.8	7.8	28.4	65.4	56.4	77.5
Average	1118.4	5.1	10.7	54.3	100.8	98.9	107.0

- LSH : Local Search-based Algorithm
- PSO : Particle Swarm Optimisation
- PSO-LS : Particle Swarm Optimisation with Local Search
- ICA : Imperialist Competitive Algorithm

The optimality gaps of CPLEX were obtained based on its best solutions and their corresponding lower bounds LBs . Similarly, the gaps for approximate methods were also obtained between their best found solutions and the LBs . One notes that the CPU times of TPIH, ETPIH and ATPIH correspond to all time required by the multi-start procedure described. Since the other heuristics were executed five times, their CPU times are also derived from the sum of all five executions.

3.3.2 Results

According to Table 3.1, ETPIH outperforms the other approximate methods for the data sets, as it provides the best average global gap and the second best average CPU time. As the fastest algorithm, TPIH is only second best in terms of average gap since its value is 5.3% worse than ETPIH.

Moreover, we argue that the formulation of CDP in (2.1) - (2.12) produces a weak lower bound notably due to the presence of constraints (2.5), which are known to be among the loosest ways to prevent subtours, and the big-M in constraints (2.8).

In order to support this assertion, we took the 287 instances that were solved to optimality by CPLEX and computed a weighted average of the gap that CPLEX still had to close when the optimal solution was first found, as depicted in Figure 23. The yielded value, 7.86%, seems to suggest that in general, when CPLEX cannot converge to optimality within a given time limit, most of the optimality gap is due to the poor quality of the best LB .

Since the performances of all the methods are evaluated with respect to this latter, and based on such a reasoning, it is not unrealistic to suppose that the solutions yielded by ETPIH are much closer to the optimum than suggested by its average gap of 5.1%, and ultimately of much higher quality.

Tables A.1 - A.6 provide the impact of each parameter used on average gaps and CPU times of each method. One notes that the instances with cluster location of collection cen-

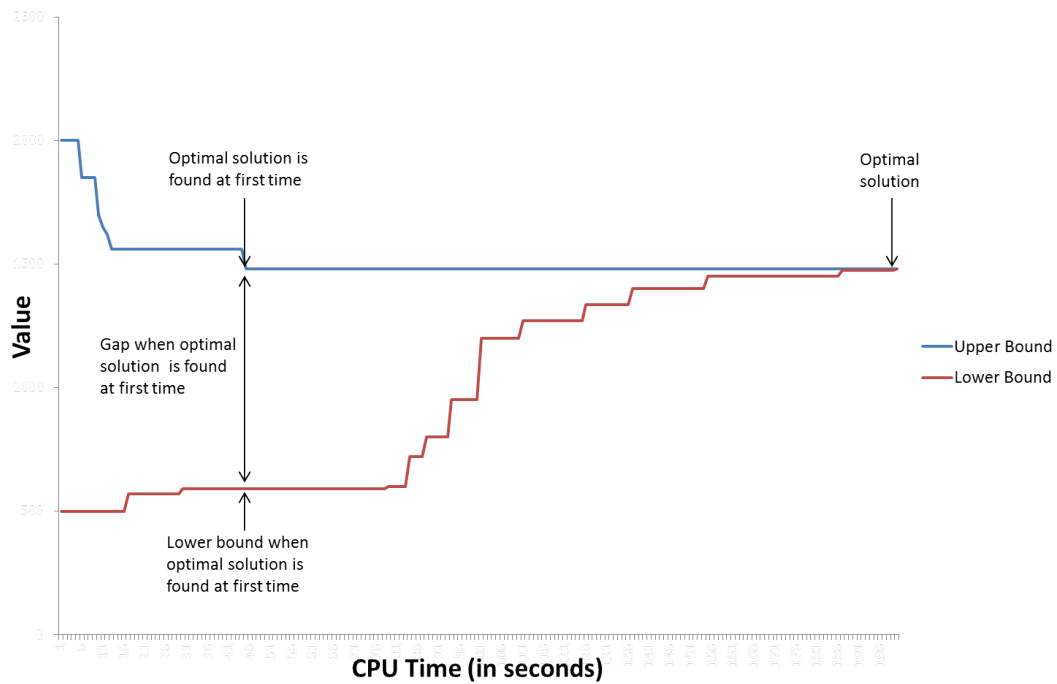


Figure 23 – Gap When Optimal Solution Found at First Time and Its Lower Bound

tres tend to induce larger heuristics' gaps. Parameter of Nodes, Periods and Components increment the optimality gaps and the CPU times of all proposed approximate methods. The instances with larger initial inventory level lead to decrease the gaps. Tables B.1 - B.8 provide more detail results of Data Set I.

Following the obtained results, we were suggested to re-execute the two best methods with longer CPU time in order to obtain better solutions. Accordingly, we had carried out the suggestions by running both methods in 25, 50, 75 and 100 seconds. The results are presented in Table 3.3.

Table 3.3 – Global Average Gaps (in %) within 100 seconds

Data Set	CPLEX CPU Time (s)	\mathcal{H}				\mathcal{H}^*			
		$\leq 25s$	$\leq 50s$	$\leq 75s$	$\leq 100s$	$\leq 25s$	$\leq 50s$	$\leq 75s$	$\leq 100s$
Random 1	1715.5 ⁽⁴⁶⁾	5.5	4.77	4.45	4.22	1.68	1.08	0.83	0.74
Random 2	1168.5 ⁽³³⁾	0.98	0.73	0.58	0.49	-0.17	-0.38	-0.48	-0.55
Cluster 1	1961.4 ⁽⁴⁶⁾	10.29	8.82	8.04	7.56	1.02	0.74	0.57	0.46
Cluster 2	2005.9 ⁽⁴⁷⁾	8.68	7.19	6.85	6.48	1.22	0.76	0.59	0.5

(-) indicates number of instances that were not solved optimally. Note that each data set consists of 108 instances.

3.3. NUMERICAL EXPERIMENTS

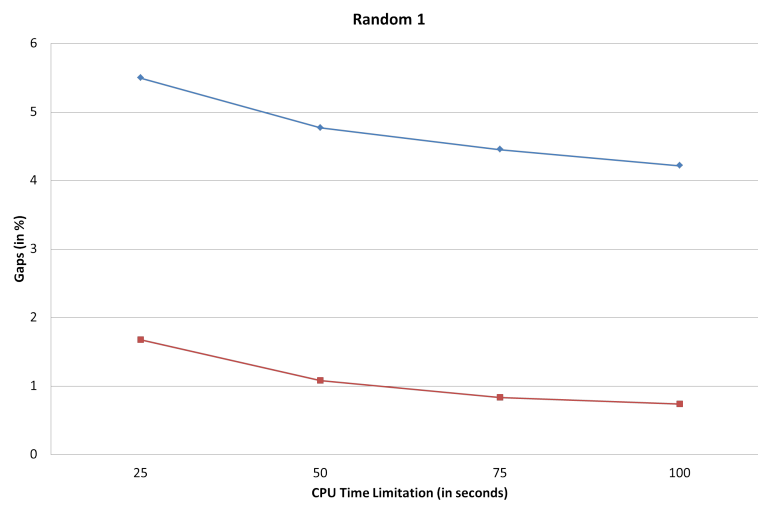


Figure 24 – Results of Data Set - Random 1 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)

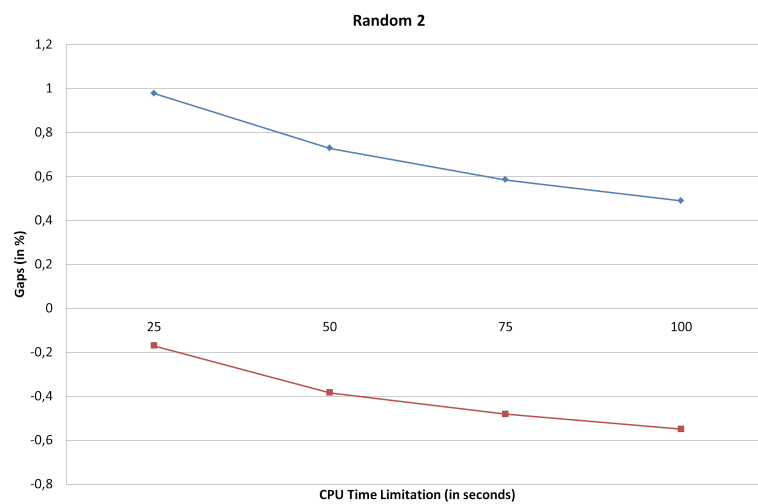


Figure 25 – Results of Data Set - Random 2 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)

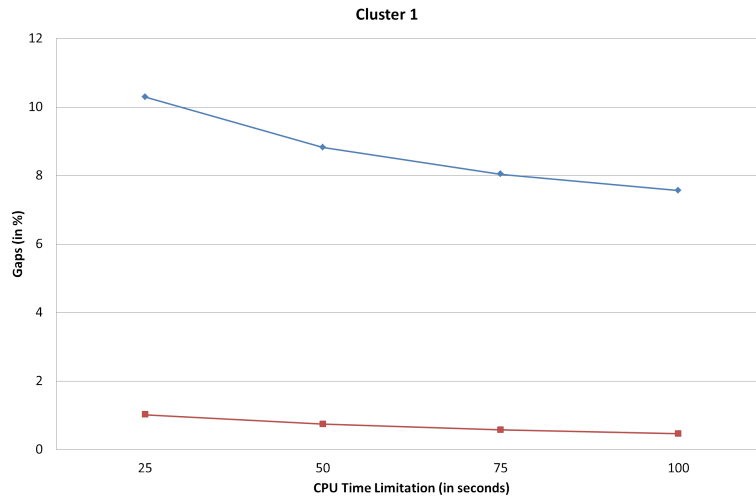


Figure 26 – Results of Data Set - Cluster 1 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)

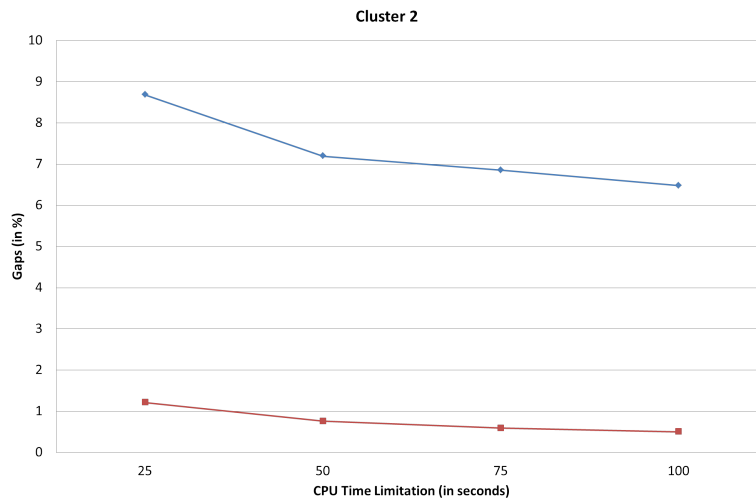


Figure 27 – Results of Data Set - Cluster 2 using \mathcal{H} (in blue) and \mathcal{H}^* (in red)

3.4 Conclusions

As shown in Chapter 2, the commercial solver CPLEX was unable to provide optimal solutions under acceptable CPU times on the tested problem instances. To provide (near) optimal solutions under faster CPU times, some approximate methods developed in the

current chapter: Two-Phase Iterative Heuristic, Two-Phase Iterative Heuristic with Enhancements, Local Search-based Heuristic, Particle Swarm Optimisation, Particle Swarm Optimisation with Local Search Enhancement and Imperialist Competitive Algorithm. All methods were evaluated using the generated data sets following the procedure described in Chapter 2.

Based on the numerical experiments, the obtained results show that the Two-Phase Iterative Heuristic with Enhancements outperforms the other approximate methods. Since the formulation of CDP is estimated as a weak formulation due to its quality of lower bounds, we argue that the gap between the optimal solutions of instances that were not solved optimally and the solutions proposed by the best approximate method might be smaller. One notes that the parameters of location, nodes, periods, components and initial stock level have significant impact on gaps and CPU times of the proposed approximate methods.

However, the uncertainty corresponding to the quantity and the quality of EOL products as well as the demands of their components is often present in reverse supply chains. Furthermore, companies processing EOL products frequently have more than one vehicle. Therefore, the next chapter deals with these issues.

3.5 Publication

Some parts of this chapter are based on the following article:

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. An Efficient Two-Phase Iterative Heuristic for Collection-Disassembly Problem, *Computers & Industrial Engineering*, vol. 110, 2017, p. 505-514.

Appendix A

Average Gaps and CPU Times

This appendix entails all tables containing the average gaps and CPU times based on parameters of each data sets.

Table A.1 – Average Gaps (in %) based on Parameters of Data Set I

	Parameter	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
Location	Random 1	5.7 ⁽⁴⁶⁾	12.6	7.4	31.2	35.3	35.5	28.9
	Random 2	4.3 ⁽³³⁾	6.3	4.3	32.5	36.1	36.5	29.0
	Cluster 1	10.7 ⁽⁴⁶⁾	24.4	12.2	35.1	46.1	45.7	38.0
	Cluster 2	10.6 ⁽⁴⁷⁾	22.7	12.3	35.3	44.8	44.3	37.5
Nodes	5	0.3 ⁽¹⁴⁾	7.5	2.5	14.6	21.1	21.0	27.9
	10	5.3 ⁽⁵³⁾	15.6	7.5	30.9	40.4	40.1	35.1
	25	17.9 ⁽¹⁰⁵⁾	26.4	17.2 [‡]	55.1	60.2	60.4	37.0
Periods	5	1.0 ⁽¹²⁾	8.3	2.5	10.9	4.4	4.4	22.4
	10	7.1 ⁽⁵⁴⁾	15.4	7.8	32.2	24.9	24.9	31.8
	25	15.3 ⁽¹⁰⁶⁾	25.8	16.8	57.4	92.4	92.2	45.9
Components	5	7.4 ⁽⁷⁹⁾	15.6	8.2	22.5	39.4	39.8	34.0
	10	8.2 ⁽⁹³⁾	17.4	9.9	44.6	41.7	41.3	32.7
Demands	$U(40\% : 60\%) \cdot \mathbf{S}$	7.1 ⁽⁶⁶⁾	14.6	8.5	31.2	51.7	51.1	47.1
	$U(90\% : 100\%) \cdot \mathbf{S}$	8.5 ⁽¹⁰⁶⁾	18.3	9.6	37.7	29.4	29.9	19.7
Disassembly	Under Constrained	7.7 ⁽⁶¹⁾	16.1	8.9	33.0	40.5	41.3	33.1
Capacity	Constrained	7.8 ⁽⁶¹⁾	16.5	8.9	32.7	40.6	39.5	33.7
	Infinite	7.9 ⁽⁵⁰⁾	16.9	9.3	34.1	40.6	40.7	33.3

$U(a : b)$ indicates that the corresponding parameter was generated with uniform distribution with parameter a and b

\mathbf{S} is the average of supply of EOL product for all collection centres and all periods

(–) indicates number of instances that were not solved optimally. Note that each line of parameter location, nodes, periods components, demands, disassembly capacity consist of 108, 144, 144, 216, 216 and 144 instances, respectively

[‡] indicates that the corresponding method provides better solutions than CPLEX.

Table A.2 – Average CPU Times (in seconds) based on Parameters of Data Set I

Parameter		CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
Location	Random 1	1715.5	6.3	20.0	114.1	188.4	184.8	165.2
	Random 2	1168.5	5.0	16.1	104.8	178.4	187.2	167.4
	Cluster 1	1961.4	8.8	17.6	90.9	178.6	184.2	165.0
	Cluster 2	2005.9	1.9	3.7	94.6	183.2	191.1	165.6
Nodes	5	863.1	4.0	9.9	17.1	26.6	30.8	59.0
	10	2317.7	5.5	12.9	42.8	74.3	80.0	102.2
	25	1957.7	7.2	20.2	243.3	445.5	449.6	336.0
Periods	5	607.3	2.5	4.1	39.5	64.5	67.6	65.2
	10	1342.7	4.3	8.5	81.1	130.9	135.9	126.0
	25	3188.4	9.8	30.4	182.6	351.0	357.0	306.1
Components	5	1580.4	4.7	12.2	88.6	181.1	182.3	165.2
	10	1845.2	6.3	16.5	113.5	183.2	191.3	166.3
Demands	$U(40\% : 60\%) \cdot \mathcal{S}$	1053.7	4.7	11.4	97.7	177.2	182.3	165.9
	$U(90\% : 100\%) \cdot \mathcal{S}$	2371.9	6.4	17.4	104.4	187.1	191.4	165.6
Disassembly Capacity	Under Constrained	1565,2	5,3	14,1	98,4	191,3	192,1	165,6
	Constrained	1795,8	5,4	14,3	108,4	174,4	190,6	165,8
	Infinite	1777,4	5,9	14,7	101,2	180,8	177,8	165,9

$U(a : b)$ indicates that the corresponding parameter was generated with uniform distribution with parameter a and b
 \mathcal{S} is the average of supply of EOL product for all collection centres and all periods.

Table A.3 – Average Gaps (in %) based on Parameters of Data Set II

Parameter		CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
Supply	$U(9 : 11)$	0.0	5.0	1.9	18.1	15.5	16.4	42.1
	$U(40 : 60)$	0.0	1.7	1.1	18.3	7.7	7.1	42.7
Vehicle Capacity	$2 \cdot \mathcal{S}$	0.0	2.7	1.6	18.4	12.0	12.3	43.0
	$3 \cdot \mathcal{S}$	0.0	3.3	1.7	18.2	11.3	11.5	41.9
	$4 \cdot \mathcal{S}$	0.0	4.1	1.2	18.1	11.4	11.5	42.2
Initial Stock	Zero	0.0	3.6	2.4	18.6	11.4	12.1	40.1
	Small	0.0	4.0	0.8	18.8	12.1	12.0	43.3
	Large	0.0	2.6	1.4	17.3	11.2	11.2	43.7

$U(a : b)$ indicates that the corresponding parameter was generated with uniform distribution with parameter a and b
 \mathcal{S} is the average of supply of EOL product for all collection centres and all periods.

Table A.4 – Average CPU Times (in seconds) based on Parameters of Data Set II

Parameter		CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
Supply	$U(9 : 11)$	721.5	6.3	10.5	33.8	56.3	52.8	77.6
	$U(40 : 60)$	133.8	3.8	9.2	33.3	53.5	54.2	78.0
Vehicle Capacity	$2 \cdot \mathcal{S}$	365.0	5.4	10.0	31.2	51.0	51.6	78.0
	$3 \cdot \mathcal{S}$	566.2	4.8	10.2	34.6	55.6	55.0	77.6
	$4 \cdot \mathcal{S}$	351.7	4.9	9.3	34.8	58.2	54.0	77.8
Initial Stock	Zero	1015.0	5.4	11.2	30.4	58.8	53.3	78.1
	Small	187.9	4.9	9.2	36.5	55.8	55.4	77.6
	Large	80.0	4.7	9.1	33.8	50.1	51.8	77.7

$U(a : b)$ indicates that the corresponding parameter was generated with uniform distribution with parameter a and b
 \mathcal{S} is the average of supply of EOL product for all collection centres and all periods.

Table A.5 – Average Gaps (in %) based on Parameters of Data Set III

Parameter		CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
Fixed Vehicle Cost (CF)	5	0.0	13.0	2.2	34.1	20.5	28.1	45.7
	10	0.0	8.7	1.1	15.6	14.4	17.8	51.3
	25	2.7 ⁽¹⁾	4.6	2.4	6.3	16.2	15.4	142.8
Disassembly Cost (CD)	$0.5 \cdot CF$	2.7 ⁽¹⁾	15.4	3.1	31.6	22.7	29.6	48.0
	$1 \cdot CF$	0.0	7.9	1.3	18.6	14.9	16.8	62.8
	$2 \cdot CF$	0.0	3.0	1.4	5.8	13.4	15.0	129.0

(-) indicates number of instances (out of 9) that were not solved optimally.

APPENDIX A. AVERAGE GAPS AND CPU TIMES

Table A.6 – Average CPU Times (in seconds) based on Parameters of Data Set III

Parameter		CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
Fixed Vehicle	5	621.7	6.9	10.7	30.7	66.9	55.7	77.5
Cost (CF)	10	2417.3	4.3	8.0	34.5	71.5	60.0	77.5
	25	605.0	3.1	4.7	19.9	58.0	53.6	77.6
	$0.5 \cdot CF$	3037.6	7.1	11.1	31.2	64.1	58.5	77.7
Disassembly Cost (CD)	$1 \cdot CF$	374.1	4.2	7.8	30.5	71.3	51.4	77.6
	$2 \cdot CF$	232.2	3.0	4.5	23.4	60.9	59.3	77.3

Appendix B

Detail Results of Data Set I

Table B.1 – Average Gaps (in %) of Data Set I - Random 1

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA	
5	5	5	0.0	2.7	0.0	1.3	1.7	1.7	2.3	
		10	0.0	3.1	1.0	6.5	1.9	1.7	5.2	
	10	5	0.0	2.9	0.7	9.6	8.0	9.1	30.0	
		10	0.0	7.1	1.9	20.3	10.8	11.6	24.2	
	10	25	5	1.4 ⁽²⁾	10.6	6.9	15.6	52.5	48.1	31.4
			10	0.9 ⁽⁴⁾	8.7	6.3	40.0	46.3	45.5	30.3
5		5	0.0	3.9	0.2	4.2	3.2	2.2	25.7	
		10	0.0	9.8	2.7	13.9	2.8	2.7	21.5	
10		5	0.0	9.3	1.3	15.5	15.0	12.8	31.5	
		10	2.6 ⁽³⁾	9.0	3.7	33.1	17.3	19.8	31.0	
25	25	5	11.8 ⁽⁶⁾	19.5	15.6	39.3	90.1	100.3	46.6	
		10	13.4 ⁽⁶⁾	20.7	15.6	72.7	91.8	85.6	47.7	
	5	5	0.5 ⁽¹⁾	7.9	1.4	13.2	4.7	6.0	29.4	
		10	0.0	6.8	1.8	20.2	5.5	5.4	26.0	
	10	5	10.3 ⁽⁶⁾	21.2	12.3	38.2	32.1	33.3	28.7	
		10	16.7 ⁽⁶⁾	24.2	17.5	62.4	39.5	39.7	30.4	
25	5	20.9 ⁽⁶⁾	29.0	20.9	55.5	95.2	105.3	38.4		
	10	23.2 ⁽⁶⁾	30.6	23.0 [‡]	101.1	116.5	108.2	40.2		
Average			5.7 ⁽⁴⁶⁾	12.6	7.4	31.2	35.3	35.5	28.9	

(-) indicates number of instances (out of 6) that were not solved optimally

‡ indicates that the corresponding method provides better solutions than CPLEX.

Table B.2 – Average CPU Times (in seconds) of Data Set I - Random 1

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
5	5	5	0.1	2.2	4.6	6.8	8.6	12.1	20.3
		10	0.1	2.0	5.4	7.5	8.8	11.8	20.9
	10	5	1.2	2.6	9.0	13.4	22.3	25.3	45.7
		10	3.5	3.9	9.2	14.5	22.3	25.6	44.9
	25	5	3806.0	6.8	23.5	27.1	49.0	54.9	106.0
		10	3856.1	10.8	23.0	38.2	55.6	58.4	107.9
10	5	5	3.3	3.2	4.9	17.7	26.6	29.9	40.6
		10	3.1	3.2	5.6	18.3	30.0	34.4	41.1
	10	5	1471.5	4.0	10.2	37.3	49.8	63.1	77.7
		10	2752.8	6.0	12.6	41.1	62.0	66.1	79.0
	25	5	3408.1	9.0	25.4	73.0	125.0	142.9	184.4
		10	4506.8	12.8	42.5	99.1	149.5	149.0	185.2
25	5	5	391.4	3.2	5.5	95.8	153.7	153.3	132.0
		10	851.4	3.3	7.2	99.7	149.5	153.6	131.6
	10	5	4513.2	4.5	13.3	200.2	321.0	327.8	254.2
		10	1825.0	7.6	18.2	223.6	308.7	321.6	254.7
	25	5	1619.4	12.7	64.1	482.9	955.2	837.9	621.7
		10	1865.4	16.4	76.1	557.4	893.0	857.9	624.9
Average			1715.5	6.3	20.0	114.1	188.4	184.8	165.2

Table B.3 – Average Gaps (in %) of Data Set I - Random 2

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
5	5	5	0.0	3.1	1.3	1.6	1.9	2.1	19.5
		10	0.0	1.6	0.4	7.9	2.6	2.9	21.4
	10	5	0.0	3.9	0.6	9.0	10.5	9.8	30.1
		10	0.0	2.0	1.1	23.2	14.2	14.7	25.2
	25	5	0.4 ⁽¹⁾	7.5	2.3	13.9	49.1	51.6	36.0
		10	0.2 ⁽²⁾	3.9	2.1	42.5	50.7	48.2	31.7
10	5	5	0.0	3.4	0.2	6.1	3.2	3.4	24.8
		10	0.0	4.4	1.8	14.7	1.8	2.6	21.9
	10	5	0.0	1.6	0.2	21.1	16.0	15.9	24.7
		10	0.2 ⁽¹⁾	4.9	2.1	34.9	20.5	18.3	25.7
	25	5	5.5 ⁽³⁾	9.6	6.4	36.3	81.4	90.9	46.2
		10	6.9 ⁽⁵⁾	13.6	10.6	70.3	96.7	86.2	47.8
25	5	5	0.0	0.9	0.1	13.7	3.9	3.2	7.8
		10	0.0	1.1	0.0	25.2	3.2	4.7	12.4
	10	5	6.8 ⁽³⁾	7.2	6.1 [‡]	37.1	30.5	29.8	24.2
		10	11.8 ⁽⁶⁾	10.6	10.4 [‡]	65.2	37.1	38.6	31.1
	25	5	20.8 ⁽⁶⁾	16.4	16.1 [‡]	56.7	112.4	122.2	45.2
		10	24.8 ⁽⁶⁾	16.8	16.1 [‡]	105.4	113.3	112.3	46.6
Average			4.3 ⁽³³⁾	6.3	4.3	32.5	36.1	36.5	29.0

(-) indicates number of instances (out of 6) that were not solved optimally

‡ indicates that the corresponding method provides better solutions than CPLEX.

Table B.4 – Average CPU Times (in seconds) of Data Set I - Random 2

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA
5	5	5	0.1	2.0	3.7	7.2	7.9	9.4	24.2
		10	0.1	2.3	4.2	7.7	8.7	11.8	24.6
	10	5	0.7	3.0	7.0	12.6	21.1	22.4	46.0
		10	1.3	3.5	8.5	15.1	22.2	23.3	46.4
	25	5	2413.4	5.1	19.9	24.5	44.8	55.7	112.5
		10	1555.3	8.5	32.8	37.3	50.7	61.4	114.6
10	5	5	1.2	2.1	4.5	19.1	25.5	29.4	41.2
		10	2.0	2.7	5.5	20.0	29.9	27.0	41.2
	10	5	11.0	3.2	8.2	39.4	60.8	61.7	78.8
		10	1497.2	4.7	11.6	34.8	58.3	60.9	78.6
	25	5	3038.5	7.2	24.6	61.6	115.2	150.1	191.5
		10	5194.0	12.4	37.8	82.8	131.8	172.4	195.3
25	5	5	174.6	1.8	5.5	96.3	152.8	167.2	134.8
		10	302.3	2.2	6.2	88.1	165.4	163.3	134.1
	10	5	1350.1	3.0	8.7	196.4	323.2	339.5	255.1
		10	1817.7	5.9	14.8	194.4	305.7	341.5	256.5
	25	5	1833.1	9.2	37.1	364.0	899.8	816.2	614.4
		10	1840.4	11.9	49.8	585.0	788.2	857.1	623.1
Average			1168.5	5.0	16.1	104.8	178.4	187.2	167.4

APPENDIX B. DETAIL RESULTS OF DATA SET I

Table B.5 – Average Gaps (in %) of Data Set I - Cluster 1

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA	
5	5	5	0.0	10.2	2.3	1.3	2.8	2.6	34.6	
		10	0.0	10.6	2.0	7.7	2.7	1.8	29.8	
	10	5	0.0	10.4	2.7	4.7	7.6	7.3	33.1	
10	10	10	0.0	8.9	2.1	16.7	11.8	11.1	27.7	
		25	5	0.5 ⁽¹⁾	10.9	4.1	11.1	57.4	55.4	42.9
	5	10	2.0 ⁽¹⁾	18.6	6.6	41.2	55.8	48.9	33.1	
		10	5	0.0	10.7	0.1	3.3	3.8	3.1	31.3
	10	10	10	0.0	15.9	4.9	12.7	2.2	3.7	24.9
		25	5	1.3 ⁽¹⁾	13.2	2.3	14.6	17.7	17.9	35.0
25	5	10	2.4 ⁽²⁾	18.3	4.6	36.6	25.0	25.7	28.8	
		10	5	21.1 ⁽⁶⁾	40.9	23.1	42.7	109.6	115.6	58.0
	10	10	21.6 ⁽⁶⁾	45.6	26.5	82.0	109.5	112.7	55.3	
		25	5	3.0 ⁽²⁾	10.9	3.2	12.6	8.0	8.8	18.7
	10	10	10	4.1 ⁽³⁾	14.3	6.5	24.9	9.5	9.1	19.1
		25	5	30.5 ⁽⁶⁾	39.6	21.2 [‡]	43.3	43.0	41.1	40.4
25	10	10	30.9 ⁽⁶⁾	49.4	32.3	82.6	64.2	63.7	49.4	
		25	5	37.3 ⁽⁶⁾	57.5	36.3 [‡]	68.0	154.9	141.4	61.0
	10	10	38.3 ⁽⁶⁾	52.7	38.0 [‡]	125.1	145.1	152.6	61.5	
Average			10.7 ⁽⁴⁶⁾	24.4	12.2	35.1	46.1	45.7	38.0	

(-) indicates number of instances (out of 6) that were not solved optimally

‡ indicates that the corresponding method provides better solutions than CPLEX.

Table B.6 – Average CPU Times (in seconds) of Data Set I - Cluster 1

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA	
5	5	5	0.1	2.7	3.5	6.4	7.2	8.6	23.5	
		10	0.2	3.4	4.9	7.8	8.8	10.5	23.6	
	10	5	8.0	4.2	6.5	11.3	20.3	24.4	44.5	
10	10	10	11.9	5.8	8.6	14.7	21.3	26.9	45.0	
		25	5	158.6	7.8	20.3	24.9	50.9	54.8	106.0
	5	10	1906.0	9.9	27.5	34.6	51.2	57.2	106.2	
		10	5	4.3	4.0	4.3	17.9	25.9	28.6	40.6
	10	10	10	4.5	3.9	5.3	20.0	32.1	32.8	40.6
		25	5	1853.7	6.2	8.4	31.9	57.6	55.6	77.6
25	10	10	2818.9	7.4	10.7	32.6	53.2	56.0	78.2	
		25	5	6306.9	12.2	28.3	50.0	147.0	149.3	184.0
	5	10	7203.8	16.4	38.9	70.1	145.5	153.8	186.4	
		10	5	3615.9	4.7	5.0	87.0	149.2	165.7	130.5
	10	10	10	3640.7	6.8	6.0	93.9	170.9	153.1	130.3
		25	5	2313.5	7.9	12.0	170.9	322.0	309.3	253.0
25	10	10	1814.5	10.7	13.9	209.5	325.6	337.3	253.4	
		25	5	1822.0	18.8	43.5	297.0	781.3	790.1	622.1
	10	10	1821.0	26.5	69.2	455.6	844.5	902.0	623.6	
Average			1961.4	8.8	17.6	90.9	178.6	184.2	165.0	

Table B.7 – Average Gaps (in %) of Data Set I - Cluster 2

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA	
5	5	5	0.0	8.4	1.4	0.7	3.3	3.3	30.2	
		10	0.0	6.7	0.8	5.8	1.9	1.9	27.0	
	10	5	0.0	7.0	1.9	6.4	10.1	9.4	30.8	
		10	0.0	7.9	1.4	16.3	10.8	12.2	27.7	
	10	25	5	0.0 ⁽¹⁾	11.1	5.7	9.7	41.3	51.7	32.5
			10	1.0 ⁽²⁾	13.0	4.5	36.4	51.6	52.3	33.5
5		5	0.0	6.3	2.0	3.5	3.4	3.2	26.1	
		10	0.0	14.4	5.2	11.5	3.1	3.3	25.5	
10		5	0.0 ⁽¹⁾	14.2	2.2	13.9	16.0	15.4	27.2	
		10	0.7 ⁽¹⁾	9.5	2.1	34.6	22.7	24.3	24.0	
25	25	5	19.4 ⁽⁶⁾	34.0	22.2	44.4	114.0	98.4	56.8	
		10	20.2 ⁽⁶⁾	40.3	23.8	79.8	101.9	99.5	55.5	
	5	5	9.9 ⁽³⁾	21.2	10.0	21.2	15.8	15.8	28.2	
		10	6.1 ⁽³⁾	19.8	10.2	29.0	13.5	10.7	23.2	
	10	5	27.8 ⁽⁶⁾	41.8	27.2 [‡]	52.8	52.9	54.6	51.6	
		10	29.2 ⁽⁶⁾	45.0	29.6	79.5	65.0	61.5	49.9	
25	5	5	37.3 ⁽⁶⁾	52.8	35.5 [‡]	66.7	146.7	139.1	62.4	
		10	38.4 ⁽⁶⁾	54.7	36.0 [‡]	122.1	132.8	141.5	62.3	
Average			10.6 ⁽⁴⁷⁾	22.7	12.3	35.3	44.8	44.3	37.5	

(-) indicates number of instances (out of 6) that were not solved optimally

‡ indicates that the corresponding method provides better solutions than CPLEX.

Table B.8 – Average CPU Times (in seconds) of Data Set I - Cluster 2

Nodes	Periods	Components	CPLEX	TPIH	ETPIH	LSH	PSO	PSO-LS	ICA	
5	5	5	0.1	0.8	0.9	6.8	7.0	8.6	23.3	
		10	0.2	0.8	1.0	7.5	8.2	11.3	23.9	
	10	5	1.3	1.2	1.6	13.5	21.9	23.1	44.5	
		10	2.3	1.2	1.8	16.1	20.8	24.6	45.2	
	10	25	5	3086.1	2.0	4.8	22.2	49.0	60.0	107.4
			10	3900.8	2.5	6.1	33.5	51.0	57.0	109.9
5		5	3.4	0.9	1.1	18.4	23.6	27.1	39.9	
		10	4.7	0.8	1.3	21.7	26.7	28.2	40.7	
10		5	2015.2	1.2	2.1	33.6	52.2	58.0	77.8	
		10	1802.4	1.5	2.4	31.6	64.6	61.1	78.7	
25	25	5	5411.3	3.4	5.9	49.7	141.1	141.7	185.9	
		10	6305.5	3.1	8.6	105.5	148.2	141.5	188.5	
	5	5	1912.2	1.1	1.0	91.2	153.9	167.7	130.4	
		10	3660.1	1.1	1.4	85.8	167.3	176.7	130.6	
	10	5	2514.7	1.7	2.4	173.2	314.4	288.9	253.7	
		10	1822.2	1.9	2.7	184.6	289.6	317.1	253.9	
25	5	5	1829.3	4.9	9.1	310.1	831.3	902.7	622.3	
		10	1833.8	4.4	12.1	497.4	926.2	943.6	623.5	
Average			2005.9	1.9	3.7	94.6	183.2	191.1	165.6	

Appendix C

Details Results of Data Set I within 100 seconds

Table C.1 – Average Gaps in Data Set - Random 1 (in %) within 100 seconds

Nodes	Periods	Components	CPLEX CPU Time (s)	TPIH				ETPIH			
				≤ 25s	≤ 50s	≤ 75s	≤ 100s	≤ 25s	≤ 50s	≤ 75s	≤ 100s
5	5	5	0.1	2.14	2.14	1.52	1.52	0	0	0	0
		10	0.1	1.65	1.6	1.6	0.97	0.97	0.97	0.94	
		5	1.2	2.12	1.49	1.46	1.41	0.84	0.57	0.19	0.07
	10	10	3.5	6.25	4.89	4.89	4.69	0.71	0.4	0.38	0.38
		5	3806 ⁽²⁾	8.23	7.85	6.97	6.97	3.99	3.74	2.9	2.85
		10	3856.1 ⁽⁴⁾	7.24	6.85	6.39	6.25	5.28	3.03	2.75	2.51
10	5	5	3.3	2.75	1.54	1.07	1.07	0.06	0.06	0.02	0.02
		10	3.1	3.19	3.03	3.03	2.05	2.18	1.21	1.21	1.21
		5	1471.5	6.66	4.95	4.51	4.27	0.25	0.21	0.08	0.08
	25	10	2752.8 ⁽³⁾	4.86	4.05	3.88	3.84	1.29	0.88	0.68	0.55
		5	3408.1 ⁽⁶⁾	6.99	6.36	6.14	6.06	3.9	3.33	3.03	2.36
		10	4506.8 ⁽⁶⁾	5.98	5.6	5.6	5.31	2.43	2.02	1.8	1.18
25	5	5	391.4	5.94	4.85	4.23	4.23	0.39	0.2	0.2	0.2
		10	851.4	5.97	5.08	5.05	4.59	1.39	1.39	1.11	1.11
		5	4513.2 ⁽⁶⁾	8.11	7.08	6.23	5.98	1.57	0.93	0.93	0.91
	10	10	1825 ⁽⁶⁾	5.77	5.09	4.71	4.31	0.73	-0.16	-0.41	-0.41
		5	1619.4 ⁽⁶⁾	7.98	7.22	7.03	6.7	3.86	1.65	0.14	0.14
		10	1865.4 ⁽⁶⁾	7.08	6.17	5.8	5.05	0.3	-1.02	-1.02	-0.88
			Max	8.23	7.85	7.03	6.97	5.28	3.74	3.03	2.85
			Min	1.65	1.49	1.07	1.07	0	-1.02	-1.02	-0.88
			Average	5.5	4.77	4.45	4.22	1.67	1.08	0.83	0.73

(-) indicates number of instances that were not solved optimally. Note that each line consists of 6 instances.

Table C.2 – Average Gaps in Data Set - Random 2 (in %) within 100 seconds

Nodes	Periods	Components	CPLEX CPU Time (s)	TPIH				ETPIH			
				≤ 25s	≤ 50s	≤ 75s	≤ 100s	≤ 25s	≤ 50s	≤ 75s	≤ 100s
5	5	5	0.1	0.33	0.12	0.12	0.12	0	0	0	0
		10	0.1	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
	10	5	0.7	3.1	3.03	2.3	2.07	0.63	0.31	0.24	0.24
		10	1.3	1.45	1.32	1.32	1.32	0.09	0.09	0.09	0.09
	25	5	2413.4 ⁽¹⁾	6.38	5.88	5.25	5.25	1.73	1.34	1.34	1.3
		10	1555.3 ⁽²⁾	3.42	3.13	3.03	2.59	2.35	1.97	1.95	1.41
10	5	5	1.2	0.48	0.35	0.35	0.3	0	0	0	0
		10	2	2.02	1.99	1.99	1.99	1.12	1.12	1.12	1.12
	10	5	11	1.2	0.64	0.64	0.64	0.01	0.01	0.01	0.01
		10	1497.2 ⁽¹⁾	2.5	2.48	2.36	2.01	0.87	0.28	0.28	0.28
	25	5	3038.5 ⁽³⁾	2.91	2.64	2.53	2.5	0.41	0.32	0.26	0.26
		10	5194 ⁽⁵⁾	5.76	5.3	4.78	4.75	4.28	3.21	2.48	2.04
25	5	5	174.6	0.87	0.23	0.23	0.23	0.01	0.01	0.01	0.01
		10	302.3	0.86	0.64	0.59	0.46	0	0	0	0
	10	5	1350.1 ⁽³⁾	-0.19	-0.41	-0.41	-0.67	-0.78	-0.94	-0.96	-0.96
		10	1817.7 ⁽⁶⁾	-1.05	-1.42	-1.53	-1.63	-1.59	-1.66	-1.81	-1.83
	25	5	1833.1 ⁽⁶⁾	-4.44	-4.68	-4.8	-4.9	-4.44	-4.95	-5.27	-5.34
		10	1840.4 ⁽⁶⁾	-8.27	-8.41	-8.5	-8.51	-8.01	-8.27	-8.65	-8.76
			Max	6.38	5.88	5.25	5.25	4.28	3.21	2.48	2.04
			Min	-8.27	-8.41	-8.5	-8.51	-8.01	-8.27	-8.65	-8.76
			Average	0.98	0.73	0.58	0.49	-0.17	-0.38	-0.48	-0.55

(-) indicates number of instances that were not solved optimally. Note that each line consists of 6 instances.

Table C.3 – Average Gaps in Data Set - Cluster 1 (in %) within 100 seconds

Nodes	Periods	Components	CPLEX CPU Time (s)	TPIH				ETPIH			
				≤ 25s	≤ 50s	≤ 75s	≤ 100s	≤ 25s	≤ 50s	≤ 75s	≤ 100s
5	5	5	0.1	2.95	2.08	1.54	1.47	0.83	0.83	0.71	0.71
		10	0.2	4.08	2.96	1.83	1.65	1.54	1.54	1.43	1.43
	10	5	8	8.45	6.92	6.68	6.44	2.01	1.96	1.95	1.69
		10	11.9	7.96	5.44	4.4	4.06	1.29	0.52	0.27	0.27
	25	5	158.6 ⁽¹⁾	8.69	8.69	8.53	8.53	3.9	3.5	3.33	2.66
		10	1906 ⁽¹⁾	15.19	14.13	13.45	13.4	4.76	4.47	3.76	3.75
10	5	5	4.3	5.32	4.84	3.67	3.67	0.12	0.12	0.12	0.12
		10	4.5	7.39	7.32	7.32	7.2	3.38	3.38	3.38	3.38
	10	5	1853.7 ⁽¹⁾	5.75	4.2	4.2	4.2	0.59	0.52	0.39	0.39
		10	2818.9 ⁽²⁾	10.61	9.71	8.53	8.17	1.55	0.99	0.64	0.64
	25	5	6306.9 ⁽⁶⁾	19.1	18.46	16.56	15.31	2.56	1.62	1.06	1.06
		10	7203.8 ⁽⁶⁾	20.3	19.86	19.16	19.16	3.66	2.55	2.26	2.19
25	5	5	3615.9 ⁽²⁾	4.96	1.91	1.68	1.68	0.06	0.04	0.03	0.03
		10	3640.7 ⁽³⁾	10.57	8.42	7.51	5.74	2.43	2.43	2.43	2.43
	10	5	2313.5 ⁽⁶⁾	1.59	-0.07	-1.85	-2.72	-9.6	-9.72	-9.72	-9.73
		10	1814.5 ⁽⁶⁾	15.78	12.16	10.64	9.56	-0.3	-0.43	-0.59	-0.69
	25	5	1822 ⁽⁶⁾	16.85	15.94	15.18	13.44	-0.84	-0.84	-0.9	-1.51
		10	1821 ⁽⁶⁾	19.64	15.78	15.7	15.12	0.46	-0.1	-0.23	-0.53
			Max	20.3	19.86	19.16	19.16	4.76	4.47	3.76	3.75
			Min	1.59	-0.07	-1.85	-2.72	-9.6	-9.72	-9.72	-9.73
			Average	10.29	8.82	8.04	7.56	1.02	0.74	0.57	0.46

(-) indicates number of instances that were not solved optimally. Note that each line consists of 6 instances.

APPENDIX C. DETAILS RESULTS OF DATA SET I WITHIN 100 SECONDS

Table C.4 – Average Gaps in Data Set - Cluster 2 (in %) within 100 seconds

Nodes	Periods	Components	CPLEX CPU Time (s)	TPIH				ETPIH			
				< 25s	< 50s	< 75s	< 100s	< 25s	< 50s	< 75s	< 100s
5	5	5	0.1	2.55	2.02	2.02	1.68	0.25	0	0	0
		10	0.2	4.68	2.19	2.19	1.23	0.82	0.82	0.82	0.82
	10	5	1.3	4.59	4.44	3.33	3.33	1.05	0.82	0.82	0.82
		10	2.3	5.14	4.67	4.39	4.35	0.78	0.6	0.59	0.59
	25	5	3086.1	8.76	8.04	7.97	7.64	4.93	4.13	3.71	3.51
		10	3900.8 ⁽²⁾	10.16	9.49	9.4	8.95	3.56	2.46	2.15	2.15
10	5	5	3.4	4.02	2.99	2.78	2.78	1.49	0.38	0.38	0.32
		10	4.7	9.16	7.44	7.27	6.48	3.59	3.14	3.14	3.14
	10	5	2015.2 ⁽¹⁾	9.93	7.68	7.36	6.8	1.5	1.02	0.95	0.59
		10	1802.4 ⁽¹⁾	8.07	6.95	6.24	6.11	0.85	0.33	0.33	0.33
	25	5	5411.3 ⁽⁶⁾	13.11	12.18	12.18	12.12	1.95	1.3	0.64	0.6
		10	6305.5 ⁽⁶⁾	19.13	16.98	16.18	15.2	3.51	2.83	2.67	2.29
25	5	5	1912.2 ⁽³⁾	3.69	2.82	2.69	2.63	0	0	0	0
		10	3660.1 ⁽³⁾	8.15	4.78	4.78	4.78	2.02	2.02	2.02	2.02
	10	5	2514.7 ⁽⁶⁾	8.21	7.36	6.82	6.82	-1.03	-1.21	-1.51	-1.51
		10	1822.2 ⁽⁶⁾	10.24	8.02	7.13	5.9	0.33	-0.67	-1.32	-1.32
	25	5	1829.3 ⁽⁶⁾	14.74	11.74	11.07	10.83	-1.79	-2.17	-2.52	-2.72
		10	1833.8 ⁽⁶⁾	11.92	9.68	9.55	8.99	-1.92	-2.07	-2.2	-2.55
			Max	19.13	16.98	16.18	15.2	4.93	4.13	3.71	3.51
			Min	2.55	2.02	2.02	1.23	-1.92	-2.17	-2.52	-2.72
			Average	8.68	7.19	6.85	6.48	1.22	0.76	0.59	0.5

(-) indicates number of instances that were not solved optimally. Note that each line consists of 6 instances.

Chapter 4

Stochastic Multi-Vehicle Collection-Disassembly Problem

Contents

4.1	Introduction	73
4.2	Problem Description	74
4.3	Formulations	75
4.4	Solution Methods	78
4.4.1	Two-Phase Iterative Heuristic	78
4.4.2	Sample Average Approximation	83
4.5	Numerical Experiments	84
4.5.1	Experimental Setup	84
4.5.2	Results	85
4.6	Conclusions	86
4.7	Publication	87

4.1 Introduction

The quantity and the quality of EOL products as well as the demands of its components are often uncertain. To deal with these issues a formulation of CDP under uncertainty is provided in this chapter and called *Stochastic CDP* (SCDP). An extension dealing with the multi-vehicle case is also provided and called *Stochastic Multi-Vehicle CDP* (SMCDP).

Two solution methods were developed to deal with SMCDP. First, Two-Phase Iterative Heuristic is addressed. Second, Two-Phase Iterative Heuristic with Enhancements is provided. The methods are selected due to their good performances on CDP as shown in

Chapter 3. An algorithmic framework of Sample Average Approximation is implemented combined with the two methods to provide high quality solutions for SMCDP.

4.2 Problem Description

The formulation presented is a CDP dealing with the uncertainty of the quantity and the quality of EOL products returned as well as of the demands of component. In particular, the quality of EOL products is translated as the quantity of their components. Referring to CDP formulation in 2.1 - 2.8, the uncertainty corresponds to S_{it} (quantity of EOL products available at collection centre i at period t), n_a (quantity of component a in a product) and q_{at} (demand of component a at period t).

Those parameters are assumed following some known probability distributions. We put all corresponding distributions into a finite scenario Ω indexed by ω . The probability associated with scenario ω is ρ^ω . For each scenario ω , we have:

- S_{it}^ω quantity of EOL products available at collection centre i at period t
- n_{at}^ω quantity of component a in a product at period t
- q_{at}^ω demand of component a at period t

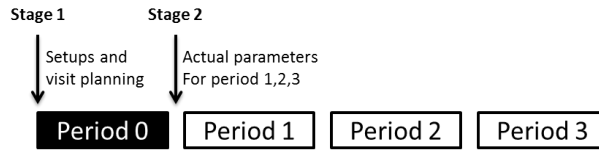


Figure 28 – Two-Stage Stochastic Problem

The problem is formalised as a two-stage stochastic problem as depicted in Figure 28 since we assume that the realisation of S_{it}^ω , n_{at}^ω and q_{at}^ω come right after the planning stage. Its implementation is illustrated in Figure 29.

In two-stage stochastic problem, there are two subsets of decision variables, namely, the first-stage and the second-stage decisions. The first-stage decision variables have to be determined before the actual realisation of the uncertain parameters. Once the uncertain parameters have presented themselves, the second-stage (also known as *recourse*) variables have to be taken subsequently. As aforementioned in Chapter 1, PDP is also formalised as a two-stage stochastic problem in Adulyasak et al. (2015a).

For SCDP with a single vehicle, the first-stage decision variables correspond to the planning of the routing for each period. Thus, the corresponding decision variables are as follows:

- z_t 1 iff period t is served, 0 otherwise
- x_{ijt} 1 iff node j visited after i directly at period t , 0 otherwise.

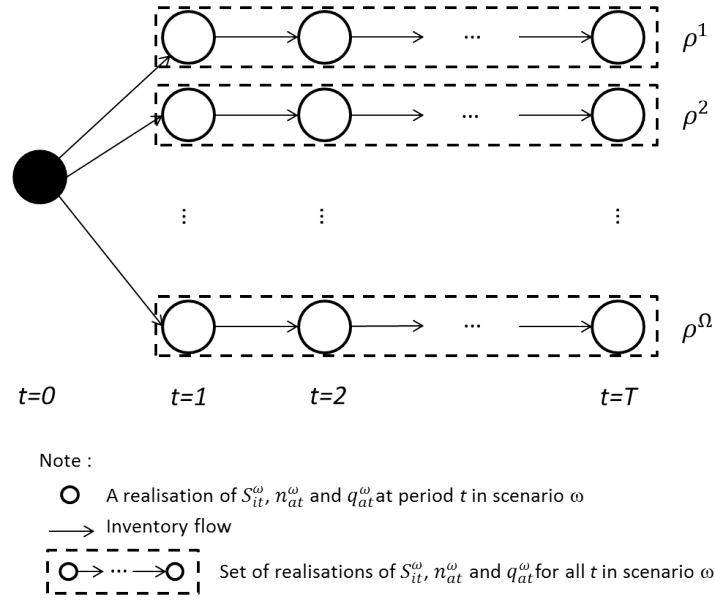


Figure 29 – Two-Stage SMCDP

The variable z_t , denotes the decision whether the period t is served. It also implies that a period served incurs the dispatch cost of using the vehicle. A period is served when there is at least one vehicle dispatched into one collection centre. Referring to Figure 28 and Figure 29, the variables z_t and x_{ijt} have to be decided at period zero.

The second-stage decision variables correspond to the vehicle load and the disassembly decisions. Those decisions will be taken after knowing the realisation of the uncertain parameters. These decision variables are:

- y_{it}^ω vehicle load after visiting node i at period t in scenario ω
- I_t^ω inventory level of EOL products at period t in scenario ω
- P_t^ω quantity of EOL products disassembled at period t in scenario ω
- SO_{at}^ω unmet demand of component a at period t in scenario ω .

These second-stage decision variables are decided by also respecting the first-stage variables.

In the next part, two formulations are presented corresponding to single-vehicle and multi-vehicle.

4.3 Formulations

Stochastic Collection-Disassembly Problem

The following formulation employs the scenario $\omega \in \Omega$ and introduces the variable z_t described in the CDP formulation (2.1) - (2.8). It is then so called SCDP.

4.3. FORMULATIONS

$$\text{Min} \sum_{t \in T} (CF \cdot z_t + \sum_{i,j \in N} c_{ij} \cdot x_{ijt} + \sum_{\omega \in \Omega} \rho^\omega (CH \cdot I_t^\omega + CD \cdot P_t^\omega + \sum_{a \in A} CP_a \cdot SO_{at}^\omega)) \quad (4.1)$$

Subject to:

$$\sum_{j \in N, i \neq j} x_{ijt} \leq 1 \quad \forall i \in N_c, \forall t \in T \quad (4.2)$$

$$x_{1it} \leq z_t \quad \forall i \in N_c, \forall t \in T \quad (4.3)$$

$$z_t \leq \sum_{i \in N_c} x_{1it} \quad \forall t \in T \quad (4.4)$$

$$\sum_{i \in N, i \neq v} x_{ivt} = \sum_{j \in N, j \neq v} x_{vjt} \quad \forall v \in N, \forall t \in T \quad (4.5)$$

$$y_{it}^\omega + (Q - S_{it}^\omega) \cdot x_{1it} \leq Q \quad \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (4.6)$$

$$y_{it}^\omega - y_{jt}^\omega + Q \cdot x_{ijt} + (Q - S_{jt}^\omega - S_{it}^\omega) \cdot x_{jit} \leq Q - S_{jt}^\omega \quad i \neq j, \forall i, j \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (4.7)$$

$$I_t^\omega = I_{t-1}^\omega + \sum_{i,j \in N, i \neq j} S_{it}^\omega \cdot x_{ijt} - P_t^\omega \quad \forall t \in T, \forall \omega \in \Omega \quad (4.8)$$

$$n_{at}^\omega \cdot P_t^\omega + SO_{at}^\omega \geq q_{at}^\omega \quad \forall a \in A, \forall t \in T, \forall \omega \in \Omega \quad (4.9)$$

$$\sum_{j \in N, i \neq j} S_{it}^\omega \cdot x_{ijt} \leq y_{it}^\omega \leq \sum_{j \in N, i \neq j} Q \cdot x_{ijt} \quad \forall i \in N, \forall t \in T, \forall \omega \in \Omega \quad (4.10)$$

$$I_t^\omega \leq \text{InvCap} \quad \forall t \in T, \forall \omega \in \Omega \quad (4.11)$$

$$P_t^\omega \leq \text{DisCap} \quad \forall t \in T, \forall \omega \in \Omega \quad (4.12)$$

$$z_t, x_{ijt} \in \{0, 1\} \quad \forall i, j \in N, \forall t \in T \quad (4.13)$$

$$y_{it}^\omega, SO_{at}^\omega, I_t^\omega, P_t^\omega \in \mathbb{Z}_{\geq 0} \quad \forall a \in A, \forall i \in N, \forall t \in T, \forall \omega \in \Omega. \quad (4.14)$$

The objective function (4.1) minimises the cost of the first stage decision and the expected cost of the second stage decisions. The first and second terms correspond to the dispatch and mileage vehicle costs. The last terms consist of the expected costs of inventory, disassembly and penalty.

Constraints (4.2) impose that each collection centre i is visited at most once during period t . Constraints (4.3) assure that a vehicle is used when at least one collection i served at period t . Constraints (4.4) state that there is no use to dispatch any vehicle if no collection centre served at period t . Constraints (4.5) balance the flows entering and leaving a node. The subtour elimination constraints (4.6) and (4.7) are based on lifting

method proposed in [Desrochers and Laporte \(1991\)](#) as CDP in Chapter 2. Constraints (4.8) are the inventory balance of the disassembly site. Constraints (4.9) impose the demand fulfilment. Constraints (4.10), (4.11) and (4.12) are the limitation of load of vehicle, inventory level and disassembly, respectively.

Stochastic Multi-Vehicle Collection-Disassembly Problem

The following formulation is called SMCDP as a version of SCDP with multi-vehicle assumption. It is assumed that there are more than one vehicle available with homogeneous capacity. A vehicle is indexed as $k, \in K$. x_{ijt}^k is 1 iff vehicle k visits node j right after i at period t . This vehicle index is also used in [Aduyasak et al. \(2015a\)](#); [Armentano et al. \(2011\)](#); [Boudia et al. \(2007\)](#) for PDP. The load of vehicle k after visiting node i at period t in scenario ω is denoted as $y_{it}^{k\omega}$. The number of vehicles dispatched at period t is denoted as z_t .

$$\text{Min} \sum_{t \in T} (CF \cdot z_t + \sum_{k \in K} \sum_{i, j \in N} c_{ij} \cdot x_{ijt}^k + \sum_{\omega \in \Omega} \rho^\omega (CH \cdot I_t^\omega + CD \cdot P_t^\omega + \sum_{a \in A} CP_a \cdot SO_{at}^\omega)) \quad (4.15)$$

subject to:

$$(4.9), (4.11), (4.12), (4.14)$$

$$\sum_{k \in K} \sum_{j \in N, i \neq j} x_{ijt}^k \leq 1 \quad \forall i \in N_c, \forall t \in T \quad (4.16)$$

$$\sum_{i \in N_c} x_{1it}^k \leq 1 \quad \forall k \in K, \forall t \in T \quad (4.17)$$

$$\sum_{k \in K} \sum_{i \in N_c} x_{1it}^k \leq z_t \quad \forall t \in T \quad (4.18)$$

$$z_t \leq K \quad \forall t \in T \quad (4.19)$$

$$\sum_{i \in N, i \neq v} x_{ivt}^k = \sum_{j \in N, j \neq v} x_{vjt}^k \quad \forall v \in N, \forall k \in K, \forall t \in T \quad (4.20)$$

$$y_{it}^{k\omega} + (Q - S_{it}^\omega) \cdot x_{1it}^k \leq Q \quad \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (4.21)$$

$$y_{it}^{k\omega} - y_{jt}^{k\omega} + Q \cdot x_{ijt}^k + (Q - S_{jt}^\omega - S_{it}^\omega) \cdot x_{jit}^k \leq Q - S_{jt}^\omega \quad i \neq j, \forall i, j \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (4.22)$$

$$I_t^\omega = I_{t-1}^\omega + \sum_{k \in K} \sum_{i, j \in N, i \neq j} S_{it}^\omega \cdot x_{ijt}^k - P_t^\omega \quad \forall t \in T, \forall \omega \in \Omega \quad (4.23)$$

$$\sum_{j \in N, i \neq j} S_{it}^\omega \cdot x_{ijt}^k \leq y_{it}^{k\omega} \leq \sum_{j \in N, i \neq j} Q \cdot x_{ijt}^k \quad \forall i \in N, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (4.24)$$

$$x_{ijt}^k \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K, \forall t \in T \quad (4.25)$$

$$z_t, y_{it}^{k\omega} \in \mathbb{Z}^+ \quad \forall i \in N, \forall k \in K, \forall t \in T, \forall \omega \in \Omega. \quad (4.26)$$

The objective function (4.15) is equivalent to (4.1) by adding the sum of vehicles at the second term.

Constraints (4.16) ensure that each collection centre is visited at most once by at most one vehicle in each period. Constraints (4.17) guarantee that each vehicle leaves the depot at most once in each period. The number of vehicles leaving the depot is limited by the number of available vehicles in constraints (4.18) and (4.19). Constraints (4.20), (4.21), (4.22) and (4.24) are equivalent to (4.5), (4.6), (4.7) and (4.10), respectively, by adding index k at \mathbf{x} and \mathbf{y} , $\forall k \in K$.

4.4 Solution Methods

Due to the second-stage variables, some constraints of SMCDP are associated with the scenario index ω . Once the number of scenarios Ω becomes very large, the problem is difficult to solve. Sample Average Approximation (SAA), a Monte Carlo simulation-based sampling method, is often used to deal with such problem. However, CPLEX is unable to solve large size instances of CDP optimally within acceptable CPU times based on Chapter 2. According to Chapter 3, Two-Phase Iterative Heuristic and Two-Phase Iterative Heuristic with Enhancements outperform the other proposed methods. To deal with the difficulties imposed by the number of scenarios and the size of instances, each of the two methods combined with SAA were implemented.

4.4.1 Two-Phase Iterative Heuristic

Two-Phase Iterative Heuristic for SMCDP

As described in Chapter 3, this method decomposes the problem into *Disassembly Lot-Sizing Problem with Approximate Visiting Costs* and routing problem. In this stochastic and multi-vehicle case, we need to add stochastic and multi-vehicle factors into the sub-problems. Henceforth, a problem called *Stochastic Multi-Vehicle Disassembly Lot-Sizing Problem with Approximate Visiting Costs* (SMDLSPAVC) is introduced.

SMDLSPAVC determines how many vehicles to dispatche z_t , which collection centre to visit by each vehicle, how many EOL products to put in the inventory I_t^ω , how many

EOL products to disassemble P_t^ω and how much penalty occurred SO_{at}^ω . Instead of using c_{ij} , it uses approximate visiting costs SC_{it}^k . These costs are initialised using $c_{0i} + c_{i0}$ and updated throughout the algorithm.

The decision variables of SMDLSPAVC are described as follows:

$$\gamma_{it}^k \begin{cases} 1 & \text{if node } i \text{ is visited by vehicle } k \text{ at period } t, i \in N_c, k \in K, t \in T \\ 0 & \text{otherwise.} \end{cases}$$

$$\beta_t^k \begin{cases} 1 & \text{if vehicle } k \text{ visits any node } N_c \text{ at period } t, k \in K, t \in T \\ 0 & \text{otherwise.} \end{cases}$$

$r_{it}^{k\omega}$ number of product collected from node i by vehicle k at period t under scenario ω
SMDLSPAVC is formalised as follows:

SMDLSPAVC

$$\text{Min} \sum_{t \in T} (CF \cdot z_t + \sum_{k \in K} \sum_{i \in N_c} SC_{it}^k \cdot \gamma_{it}^k + \sum_{\omega \in \Omega} \rho^\omega (CH \cdot I_t^\omega + CD \cdot P_t^\omega + \sum_{a \in A} CP_a \cdot SO_{at}^\omega)) \quad (4.27)$$

Subject to:

(4.9), (4.11), (4.12)

$$I_t^\omega = I_{t-1}^\omega + \sum_{k \in K} \sum_{i \in N_c} r_{it}^{k\omega} - P_t^\omega \quad \forall t \in T, \forall \omega \in \Omega \quad (4.28)$$

$$r_{it}^{k\omega} = S_{it}^\omega \cdot \gamma_{it}^k \quad \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (4.29)$$

$$\sum_{i \in N_c} r_{it}^{k\omega} \leq \min \left\{ Q; \max_a \left\{ \frac{\sum_{t'=t}^T q_{at'}^\omega}{n_a} \right\} \right\} \quad \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (4.30)$$

$$\sum_{k \in K} \gamma_{it}^k \leq 1 \quad \forall i \in N_c, \forall t \in T \quad (4.31)$$

$$\sum_{i \in N_c} \gamma_{it}^k \leq N_c \cdot \beta_t^k \quad \forall k \in K, \forall t \in T \quad (4.32)$$

$$\beta_t^k \leq \sum_{i \in N_c} \gamma_{it}^k \quad \forall k \in K, \forall t \in T \quad (4.33)$$

$$\sum_{k \in K} \beta_t^k = z_t \quad \forall t \in T \quad (4.34)$$

$$z_t \leq K \quad \forall t \in T \quad (4.35)$$

$$\gamma_{it}^k, \beta_t^k \in \{0, 1\} \quad \forall i \in N_c, \forall k \in K, \forall t \in T \quad (4.36)$$

$$SO_{at}^\omega, I_t^\omega, P_t^\omega, r_{it}^{k\omega}, z_t \in \mathbb{Z}^+ \quad \forall a \in A, \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega. \quad (4.37)$$

4.4. SOLUTION METHODS

The objective function (4.27) aims to minimise the total cost consisting of fixed cost of vehicles dispatched, approximate visiting cost and the expected cost of second-stage decisions of inventory, product disassembled and unmet demand.

Constraints (4.28) impose the inventory balance of EOL products. Constraints (4.29) state all EOL products belonging to collection centre i have to be picked up once it is visited by any vehicle. Constraints (4.30) is the maximum limit of $r_{it}^{k\omega}$. Constraints (4.31) guarantee that a collection centre is visited at most once by any vehicle for each period. Constraints (4.32) state that β_t^k is equal to 1 if vehicle k visited at least one collection centre in period t . Otherwise, β_t^k is equal to 0 as imposed by constraints (4.33). Constraints (4.34) and (4.35) state that number of vehicles used in each period is limited to the number of available vehicles. The nature of the decision variables on both stages are imposed in constraints (4.36) and (4.37).

Based on the values of γ_{it}^k obtained in SMDLSPAVC, the route of each vehicle is constructed using LKH (Lin and Kernighan, 1973) as described in Algorithm 2 in Chapter 3. If necessary, the decision values and SC_{it}^k are updated based on the objective value obtained. Apart from its initial values, the diversification mechanism of SC_{it}^k is employed in order to move to the other solution space. The implementation of the method to the problem is provided in algorithm 11.

Algorithm 11: Two-Phase Iterative Heuristic for SMCDP

```

solution  $\leftarrow \emptyset$ 
Initialise  $SC_{it}^k, \forall i \in N_c, k \in K, t \in T$ 
while stopping criterion 3 is not met do
    while stopping criterion 2 is not met do
        while stopping criteria 1 are not met do
            Solve SMDLSPAVC and get  $\gamma_{it}^k, \forall i \in N_c, \forall k \in K, \forall t \in T$ 
            Solve Routing Problem
            Update solution (if necessary) and  $SC_{it}^k$ 
        end
        Diversify  $SC_{it}^k$ 
    end
    Multi-start procedure:  $SC_{it}^k = \rho_{it} \cdot (c_{0i} + c_{i0}), \forall i \in N_c, k \in K, t \in T$ 
end

```

As mentioned in Section 3.2.1, the approximate visiting costs SC_{it}^k has a pivotal role to connect between SMDLSPAVC and the routing decisions. In each iteration, it is updated following Algorithm 12.

The diversification of SC_{it}^k is implemented by multiplying SC_{it}^k by the number of retailers served at period t plus one for all retailers. One is added to avoid zero multiplication.

Algorithm 12: Update of Approximate Visiting Costs SC_{it}^k

```

for  $\forall t \in T, \forall k \in K, \forall i \in N_c$  do
  if  $i \in route_t^k$  then
     $SC_{it}^k \leftarrow c_{i-i} + c_{ii+} - c_{i-i+}$ 
  else
     $SC_{it}^k \leftarrow \Delta_{it}$ 
  end
end

```

Two-Phase Iterative Heuristic with Enhancements for SMCDP

In this part, the enhancement of Two-Phase Iterative Heuristic for SMCDP is proposed. As described in Section 3.2.1, the enhancements are used to improve the method to obtain better solutions.

In Algorithm 11, the problem is decomposed into SMDLSPAVC and the routing problem. The solutions and SC_{it}^k are updated if the corresponding iteration provides better solution. In this method, this step is denoted as the first step.

The enhancements used in this adaptive method expands the method presented in Algorithm 11 by putting an additional step in order to propose a better solution. The step introduces SMDLSPAVC II in order to provide the solution of SMDLSPAVC serving less periods by introducing Z as a parameter indicating number of periods served in SMDLSPAVC. The variable α_t is equal to 1 if period t is served. Otherwise, it is 0. One notes that the approximate visiting costs SC_{it}^k of the second step are identical as at the first step. SMDLSPAVC II is formalised as follows:

SMDLSPAVC II

Min (4.27)

Subject to:

$$(4.9), (4.11), (4.12), (4.28) - (4.37) \quad \forall t \in T \quad (4.38)$$

$$z_t \leq K \cdot \alpha_t \quad \forall t \in T \quad (4.38)$$

$$\sum_{t \in T} \alpha_t \leq Z - 1 \quad (4.39)$$

$$\alpha_t \in \mathbb{Z}^+ \quad \forall t \in T. \quad (4.40)$$

Based on our numerical experiments in Chapter 3, the additional step indeed provides better optimality gaps but longer CPU times than Two-Phase Iterative Heuristic. Therefore, an adaptive procedure is used in this method to deal with this issue as follows. The probability of using the step is denoted as *prob*. These values are halved once the

4.4. SOLUTION METHODS

step has no contribution to improve the solution by comparing these values with random values *rand*. This method is depicted in Algorithm 13. This procedure is an extended adaptation of this method for CDP described in Section 3.2.1.

The three stopping criteria entailed in Algorithm 11 and Algorithm 13 are as follows:

- stopping criteria 1 : standard deviation of the last ten fitness values, maximum iterations and CPU times are less than 5 %, 100 and 7200 seconds, respectively
- stopping criterion 2 : maximum number of iterations of diversification mechanisms is 5
- stopping criterion 3 : maximum number of iterations of multi-start procedure is 5.

Algorithm 13: Two-Phase Iterative Heuristic with Enhancements for SMCDP

```

solution ← ∅
Initialise  $SC_{it}^k, \forall i \in N_c, k \in K, t \in T$ 
while stopping criterion 3 is not met do
    prob = 1
    while stopping criterion 2 is not met do
        while stopping criteria 1 is not met do
            FIRST STEP
            • Solve SMDLSPAVC and get  $\gamma_{it}^k, \forall i \in N_c, \forall k \in K, \forall t \in T$ 
            • Solve Routing Problem
            • Update solution (if necessary) and  $SC_{it}^k$ 

            SECOND STEP
            Generate rand
            if rand ≤ prob then
                • Solve SMDLSPAVC II and get  $\gamma_{it}^k, \forall i \in N_c, \forall k \in K, \forall t \in T$ 
                • Solve Routing Problem
                • Update solution (if necessary) and  $SC_{it}^k$ 

                if solution is not updated then
                    | prob = prob/2
                end
            end
        end
        Diversify  $SC_{it}^k$ 
    end
    Multi-start procedure:  $SC_{it}^k = \rho_{it} \cdot (c_{0i} + c_{i0}), \forall i \in N_c, k \in K, t \in T$ 
end

```

4.4.2 Sample Average Approximation

Since SMCDP is stochastic discrete optimization problem, an algorithmic framework of Sample Average Approximation (SAA) is adapted to provide statistical lower and upper bounds. It is Monte Carlo-based sampling in order to solve a problem having a very large number of scenarios denoted as Ω' , which is intractable, by solving the problem with smaller and tractable scenario Ω where $|\Omega| \ll |\Omega'|$ (Adulyasak et al., 2015a; Ghilas et al., 2016; Kleywegt et al., 2002).

The following procedure of SAA is applied to SMCDP:

1. Set replication \mathcal{M} and generate a set of scenarios Ω as well as very large scenario Ω' independently (a replication \mathcal{M} is a set of Ω). The probability of each scenario ω associated with $|\Omega|$ is $\rho^\omega = \frac{1}{|\Omega|}$.
2. For $s = 1 \rightarrow \mathcal{M}$, do :
 - 2.1. Solve SMCDP. Store the objective value Z_Ω^s , the vectors of first-stage solutions $(\mathbf{z}_\Omega^s, \mathbf{x}_\Omega^s)$ and the vectors of second-stage solutions $(\mathbf{I}_\Omega^s, \mathbf{P}_\Omega^s, \mathbf{SO}_\Omega^s)$. The average and the variance of the objective value (lower bound) after s -th replication denoted as v_Ω^s and σ_Ω^{s2} are obtained as follows:

$$v_\Omega^s = \frac{1}{s} \sum_{i=1}^s Z_\Omega^i$$

$$\sigma_\Omega^{s2} = \frac{1}{s \cdot (s-1)} \sum_{i=1}^s (Z_\Omega^i - v_\Omega^s)^2$$

- 2.2. To obtain the second stage solutions $\mathbf{I}_{\Omega'}^s, \mathbf{P}_{\Omega'}^s$ and $\mathbf{SO}_{\Omega'}^s$ of a very large scenario Ω' , use the best first stage solution after replication s denoted as $\hat{\mathbf{z}}_\Omega^s$ and $\hat{\mathbf{x}}_\Omega^s$. The corresponding objective value (upper bound) is denoted as $v_{\Omega'}(\hat{Z}^s)$ and its variance is obtained as follows:

$$\sigma_{\Omega'}^2 = \frac{1}{\Omega' \cdot (\Omega' - 1)} \sum_{\omega=1}^{\Omega'} (G_\omega - v_{\Omega'}(\hat{Z}^s))^2$$

where,

$$G_\omega = \sum_{t \in T} \left\{ CF \cdot \hat{z}_t + \sum_{k \in K} \sum_{i,j \in N} c_{ij} \cdot \hat{x}_{ijt}^k + CH \cdot \tilde{I}_t^\omega + CD \cdot \tilde{P}_t^\omega + \sum_{a \in A} CP_a \cdot \tilde{SO}_{at}^\omega \right\}$$

Note that $\tilde{I}_t^\omega, \tilde{P}_t^\omega$ and \tilde{SO}_{at}^ω correspond to the second stage solution for scenario $\omega \in \Omega'$.

4.5. NUMERICAL EXPERIMENTS

2.3. Calculate the SAA gap ε and its variance σ_ε^2 as follows:

$$\begin{aligned}\varepsilon &= v_{\Omega'}(\hat{Z}^s) - v_\Omega^s \\ \sigma_\varepsilon^2 &= \sigma_{\Omega'}^2 + \sigma_\Omega^{s^2}\end{aligned}$$

3. Return $\hat{\mathbf{z}}_\Omega^s$ and $\hat{\mathbf{x}}_\Omega^s$ as the best solution.

4.5 Numerical Experiments

In this part, the results of numerical experiments obtained are evaluated in order to assess the performance of the methods.

4.5.1 Experimental Setup

All formulations and algorithms were implemented in Java using Concert Technology and were solved by IBM CPLEX 12.6 on a PC with processor Intel®Core™i7 CPU 2.9 GHz and 4 GB of RAM under Windows 7 Professional.

The Monte Carlo simulation was used for the scenario generation of the parameters associated with uncertainty (S_{it}^ω , n_{at}^ω and q_{at}^ω). Those parameters were generated independently by multiplying the corresponding values of deterministic CDP with random value following uniform distribution from 0 to 1.5. The number of vehicles were set as 1, 3 and 5 while the large scenario Ω' were set as 1000.

To avoid memory issues, the maximum number of branch nodes of CPLEX for both two SMRLSDCs and large scenario problem of SAA is limited to 75000. The methods were tested using instances 49, 61, 73, 85 and 97 of Random 1, Random 2, Cluster 1 and Cluster 2 of Data Set I (see 2.4). The characteristic of the instances is provided in Table 4.1.

Table 4.1 – Instances

Instance	Characteristics				
	Nodes	Periods	Components	Demand	Disassembly Capacity
49	10	10	10	$U(40\% : 60\%) \cdot \mathbf{S}$	∞
61	10	5	10	$U(40\% : 60\%) \cdot \mathbf{S}$	∞
73	5	25	10	$U(40\% : 60\%) \cdot \mathbf{S}$	∞
85	5	10	10	$U(40\% : 60\%) \cdot \mathbf{S}$	∞
97	5	10	5	$U(40\% : 60\%) \cdot \mathbf{S}$	∞

$U(a : b)$ indicates that the corresponding parameter was generated with uniform distribution with parameter a and b
 \mathbf{S} is the average of supply of EOL product for all collection centres and all periods.

Table 4.2 – Results of All Data Sets

Data Sets	\mathcal{H}					\mathcal{H}^*				
	SAA			EVPI	VSS	SAA			EVPI	VSS
	μ	σ	CPU Time			μ	σ	CPU Time		
Random 1	539.5	26.9	1782.0	2349.5	-158.7	541.0	27.8	2030.6	2352.2	-160.6
Random 2	539.7	27.5	2162.1	2372.1	-129.2	540.8	27.9	2369.3	2345.4	-135.0
Cluster 1	539.3	27.6	1485.3	2208.8	-327.8	539.7	26.9	1423.3	2279.1	-326.0
Cluster 2	539.2	27.7	1438.4	2259.8	-290.7	540.1	26.7	1465.4	2282.0	-284.2

μ is average
 σ is standard deviation
 CPU time is in seconds

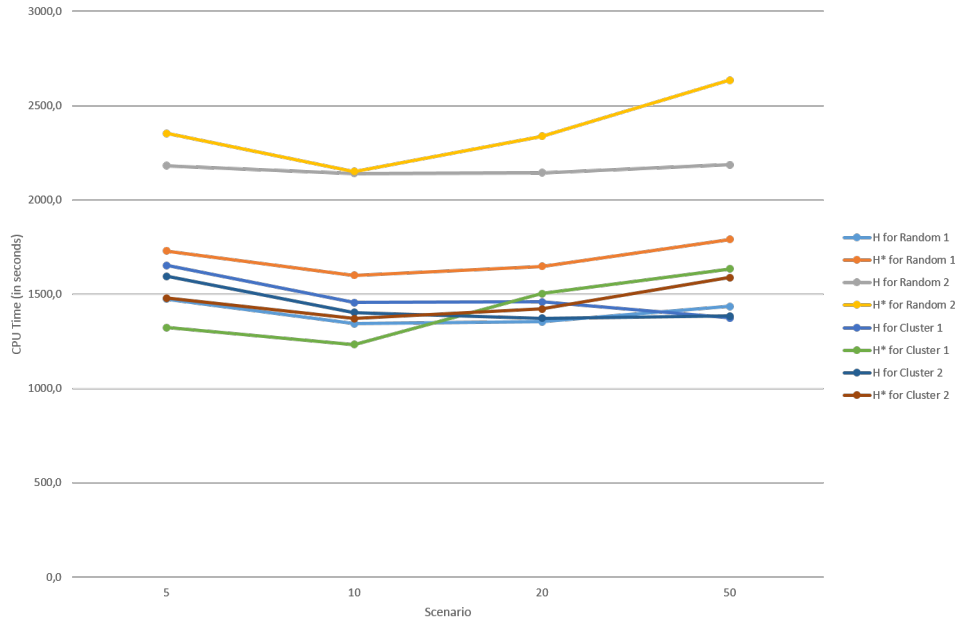


Figure 30 – Average CPU Times (in seconds) for All \mathcal{M}

4.5.2 Results

In order to simplify the representation of the results, \mathcal{H} and \mathcal{H}^* refer to Two-Phase Iterative Heuristic and Two-Phase Iterative Heuristic with Enhancements, respectively.

According to Table 4.2, both methods provide solutions with no significant difference in terms of average, standard deviation, EVPI and VSS. In terms of CPU times, \mathcal{H}^* requires shorter time in solving the data set of Random 1 and Random 2 rather than \mathcal{H}^* as shown in Table 4.2.

Based on Figures 30 and 31, one notes that both methods are stable to solve the instances although there is a variation of scenario \mathcal{M} . However, the increase of the number of available vehicle \mathcal{K} causes longer CPU times for both methods

4.6. CONCLUSIONS

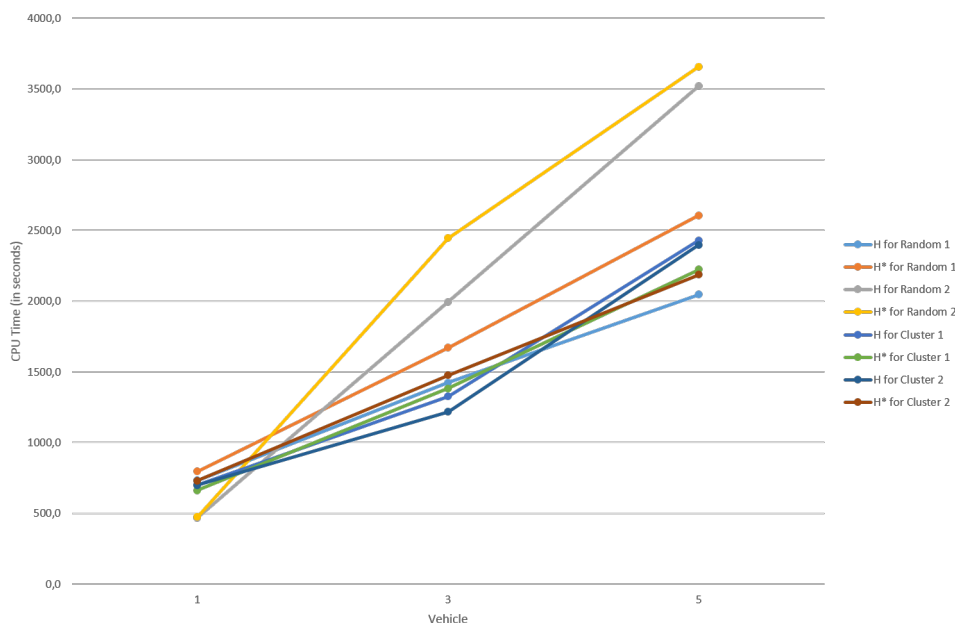


Figure 31 – Average CPU Times (in seconds) for All \mathcal{K}

4.6 Conclusions

In this chapter, the formulation of the integer linear programming described in Chapter 2 is extended. The new formulation deals with the uncertainty of the quantity and the quality of the single type EOL product as well as the demands of the associated components. The uncertain parameters are the availability of EOL product at collection centres, the number of components in each EOL product collected and their demands.

The problem is formalised as a two-stage stochastic problem in which the first-stage decision variables have to be taken during the planning stage before any realisation of the uncertain parameters. The second-stage decision variables are determined consecutively.

The first stage decisions correspond to the number of vehicles dispatched and their routing decisions. Whilst, the second stage decisions correspond to the inventory level, the number of disassembled and the unmet demands.

Accordingly, two methods are proposed to deal with: (i) Two-Phase Iterative Heuristic (\mathcal{H}) and (ii) Two-Phase Iterative Heuristic with Enhancements (\mathcal{H}^*). \mathcal{H} decomposes the problem into two subproblems: the lot-sizing problem with approximate visiting costs and the routing problem. Then, the two subproblems are solved iteratively. (\mathcal{H}^*) is an enhanced version of (\mathcal{H}) through an additional step that improves the solution provided by both subproblems. The algorithms are combined with the algorithmic framework of Sample Average Approximation. This framework is based on the Monte Carlo simulation

that allows to solve a problem having very large scenario, which is intractable, by solving the problem with smaller and tractable scenario.

Based on the obtained results, both methods have no significant difference in terms of solution and CPU time. However, \mathcal{H}^* requires longer average CPU time rather than \mathcal{H} in data sets of Random 1 and Random 2 and shorter CPU time in data sets of Cluster 1 and Cluster 2.

Chapter 2 - Chapter 4 deal with the decision integration between collection and disassembly process in which the latter process is formalised as lot-sizing problem of EOL product. The following chapter is to deal with similar integration but the disassembly process is formalised as disassembly line balancing problem.

4.7 Publication

The following paper based on this chapter is currently under preparation:

- Habibi, M. K. K., Battaïa, O., Cung, V.-D., Dolgui, A., Tiwari, M. K. Sample Average Approximation for Multi-Vehicle Collection-Disassembly Problem under Uncertainty. Submitted to and currently under revision in *International Journal of Production Research*.

4.7. PUBLICATION

Table 4.3 – Results of Random 1

Instance	K	Ω	M	\mathcal{H}						\mathcal{H}^*					
				SAA			EVPI	VSS	SAA			EVPI	VSS		
				μ	σ	CPU Time			μ	σ	CPU Time				
49	1	5	200	497.6	20.8	1094.5	2158.8	-85.9	493.5	20.3	1150.9	1954.9	-88.1		
				487.4	20.2	847.7	2020.3	-88.8	500.8	19.6	1011.0	2126.6	-85.4		
				480.1	20.3	804.8	2196.4	-101.5	493.4	21.9	1087.9	2196.6	-90.7		
				486.3	24.2	753.3	2156.4	-91.3	500.3	20.6	818.4	2182.5	-82.2		
				494.9	21.3	2266.1	2369.0	-75.5	495.0	22.0	2189.7	2260.3	-80.8		
	3	20	50	20	496.2	28.2	1631.6	1966.3	-85.4	499.2	21.6	1915.6	2034.0	-87.2	
					494.4	20.7	2044.9	2360.4	-85.6	501.0	21.1	2033.4	2176.8	-82.4	
					496.8	20.8	1846.7	2183.5	-80.1	491.1	19.3	1863.5	2193.0	-89.2	
					496.1	21.2	3165.3	2349.5	-84.3	501.6	22.3	3020.1	2185.7	-83.9	
					491.4	21.0	2620.7	2250.8	-79.6	492.4	21.9	2740.7	2101.2	-87.8	
	5	100	50	20	495.0	21.8	2787.3	2080.0	-80.8	500.3	23.3	2762.9	2145.7	-92.4	
					501.0	19.9	3898.2	2417.7	-83.3	506.1	21.3	2741.1	2280.9	-85.7	
					239.4	10.3	505.1	1117.0	-81.9	246.1	10.0	435.2	873.1	-78.7	
					239.5	10.3	380.7	1071.3	-79.4	247.3	9.0	378.8	1033.3	-78.6	
					247.2	9.6	362.5	1128.7	-74.1	239.7	9.2	351.3	1095.8	-82.3	
61	3	50	20	241.0	11.5	333.8	1019.8	-84.6	242.1	8.9	283.8	874.0	-78.9		
				242.1	9.8	722.7	870.5	-83.1	245.3	9.6	681.3	1018.1	-74.8		
				240.1	9.3	686.1	1091.1	-84.9	241.9	10.2	589.0	1080.8	-83.4		
				245.2	9.8	403.9	875.2	-74.0	235.8	9.8	568.0	867.8	-80.8		
				236.5	9.1	500.7	865.8	-81.2	245.5	13.0	485.2	1145.9	-76.8		
	5	200	5	20	241.8	10.5	869.3	872.8	-72.5	244.7	9.5	779.5	1132.3	-77.3	
					238.1	10.3	733.1	1058.5	-77.1	242.0	11.0	651.4	1144.4	-72.6	
					242.0	9.6	636.4	1058.2	-79.4	241.3	9.3	596.2	1101.2	-75.1	
					242.0	10.9	751.4	869.7	-80.9	239.4	9.6	565.1	868.5	-83.4	
					1223.9	50.6	1882.6	5439.7	-509.1	1222.8	51.1	1767.2	5133.7	-492.5	
	73	1	100	10	1233.4	48.1	1509.5	5261.3	-480.8	1210.2	55.0	1530.6	5401.5	-503.7	
					1220.6	47.2	1568.4	5177.1	-492.5	1232.0	48.0	1503.8	5275.7	-496.2	
					1217.4	49.1	1493.9	5496.0	-482.1	1234.2	40.7	1550.9	5060.4	-498.4	
					1227.0	51.4	3304.9	5144.6	-473.7	1225.2	49.3	3690.6	5562.9	-489.0	
					1224.4	50.5	3470.4	5319.6	-493.8	1216.3	55.0	3962.8	5706.9	-498.9	
3		100	20	50	1219.4	42.3	3150.9	5377.4	-492.1	1218.8	57.5	4160.3	5222.9	-493.9	
					1216.9	54.2	3578.6	5000.5	-488.7	1236.1	47.0	4497.5	5199.7	-474.1	
					1225.7	47.2	4285.0	5047.9	-489.5	1243.6	54.4	6365.8	5546.4	-487.0	
					1225.3	50.2	4272.0	4938.6	-494.8	1221.9	48.9	6288.2	5391.1	-497.1	
					1221.5	46.2	5372.5	5376.4	-491.5	1230.5	48.4	7675.5	5296.1	-491.8	
85		1	200	50	1226.3	41.6	5667.6	5264.1	-498.4	1224.5	67.5	9996.3	5533.2	-492.7	
					498.8	20.3	566.2	2289.2	-153.4	501.4	21.7	784.4	2225.0	-173.4	
					504.9	22.6	501.3	2020.0	-211.1	491.7	20.2	643.7	2240.7	-203.7	
					499.4	20.1	489.5	2282.9	-242.1	488.8	20.9	519.2	1970.9	-224.9	
					494.2	19.5	394.4	2268.2	-202.6	491.8	18.9	516.4	2230.1	-199.4	
	3	100	5	20	495.2	21.1	919.5	2263.2	-225.5	501.7	22.3	1468.9	2468.6	-200.5	
					498.8	20.7	898.2	2034.9	-246.6	497.0	23.5	1237.7	1987.0	-248.8	
					496.7	21.2	770.6	1959.9	-217.6	494.7	20.9	1038.0	2219.7	-224.4	
					494.2	18.3	728.1	2278.0	-249.9	500.1	27.9	931.1	2054.3	-175.3	
					490.6	22.9	1184.6	2263.1	-205.9	498.6	20.8	1678.3	1970.2	-191.0	
	97	1	100	10	492.9	21.0	1056.6	2049.2	-221.2	494.0	18.0	1413.2	2295.2	-197.8	
					499.2	19.9	1027.3	2194.7	-205.9	495.5	21.2	1284.2	2299.5	-168.0	
					497.0	21.3	886.6	2051.4	-213.9	495.2	20.1	1195.1	1974.8	-255.8	
					243.7	9.5	340.3	1179.1	-70.1	242.5	10.1	432.4	1073.0	-68.5	
					244.5	9.3	329.8	1119.7	-65.1	243.1	10.2	421.0	1120.0	-68.1	
3		200	5	20	239.3	11.9	248.2	1017.9	-72.9	241.1	10.6	338.7	1083.9	-64.3	
					244.6	8.5	260.3	1070.8	-70.1	245.7	9.8	387.9	1074.3	-68.9	
					242.2	10.0	417.7	1019.5	-71.1	240.9	10.2	728.2	1088.2	-68.1	
					242.1	10.0	391.4	1109.8	-69.4	245.1	10.9	536.9	1181.7	-65.8	
					236.8	9.8	377.5	1051.5	-75.3	247.8	11.0	423.9	1071.9	-68.1	
5		100	50	20	240.8	10.7	370.2	872.1	-65.3	243.4	10.4	440.4	1259.0	-62.6	
					242.2	9.6	598.1	1130.9	-68.8	241.6	9.6	763.5	1083.8	-70.4	
					238.0	10.3	409.2	870.6	-72.8	245.4	9.8	571.9	1043.8	-72.2	
					241.9	9.2	455.5	1084.3	-72.2	246.4	9.4	542.2	1034.6	-65.9	
					248.2	9.8	292.5	876.2	-65.2	243.9	9.0	524.1	1090.4	-70.1	

μ is average
 σ is standard deviation
CPU time is in seconds

CHAPTER 4. STOCHASTIC MULTI-VEHICLE COLLECTION-DISASSEMBLY
PROBLEM

Table 4.4 – Results of Random 2

Instance	K	Ω	M	\mathcal{H}					\mathcal{H}^*					
				SAA			EVPI	VSS	SAA			EVPI	VSS	
				μ	σ	CPU Time			μ	σ	CPU Time			
49	1	5	200	493.5	21.0	1083.2	2314.9	-44.3	495.2	21.0	986.7	2275.8	-67.4	
			100	488.5	21.1	1146.2	2134.3	-58.7	496.2	21.8	976.8	2192.9	-63.1	
			50	491.2	23.8	1045.2	2143.8	-49.6	496.9	21.6	872.6	2224.9	-52.6	
			20	495.4	20.4	835.8	2207.2	-48.6	498.0	22.0	781.6	2315.9	-47.6	
			5	490.6	20.8	2187.7	2123.9	-50.8	498.9	21.2	2161.6	2099.4	-45.5	
	3	10	100	200	498.2	21.3	1806.3	2213.5	-42.2	492.7	19.5	1678.9	2168.7	-75.2
				100	491.1	22.3	1647.2	2437.5	-43.3	505.4	19.9	1478.9	2108.6	-66.5
				50	492.0	28.9	1929.7	2406.4	-49.3	496.3	23.4	1637.6	2199.1	-48.6
				20	493.8	21.9	6001.0	2250.4	-46.7	493.1	21.2	2983.3	2021.6	-41.8
				5	493.2	22.5	5976.1	2161.0	-48.7	502.4	21.5	2500.0	2125.0	-57.3
	5	10	50	200	492.0	24.8	6545.2	2229.1	-58.4	505.4	20.0	2482.6	2007.7	-45.6
				100	497.6	21.2	4535.5	2255.1	-53.0	499.2	19.5	2642.6	2309.6	-47.7
				50	247.3	9.5	341.1	1060.4	-58.2	241.0	10.1	309.5	1082.9	-60.6
				20	243.0	9.1	278.5	1160.8	-52.6	242.9	10.4	307.2	1071.3	-58.3
				5	244.1	9.3	243.6	872.8	-58.8	243.4	10.6	288.8	1051.4	-58.5
61	3	50	200	240.2	8.9	245.6	1059.2	-53.9	246.6	9.0	246.6	1126.2	-53.4	
			100	243.7	10.5	535.5	874.0	-55.9	246.1	9.6	525.8	876.6	-50.4	
			100	243.6	10.7	474.8	1094.7	-55.7	239.5	10.7	591.8	1079.8	-59.7	
			50	237.8	10.8	484.2	1080.5	-58.5	241.8	9.6	532.2	1043.5	-59.3	
			20	237.8	9.5	376.4	867.7	-62.4	242.4	10.8	398.9	1079.6	-56.5	
	5	5	200	200	249.3	9.7	708.7	875.6	-52.5	244.2	9.2	779.5	1140.5	-59.4
				100	242.9	10.3	644.0	1124.7	-53.5	237.5	9.3	675.3	1055.2	-65.5
				50	244.9	9.2	502.1	874.3	-58.1	240.0	10.0	631.3	1035.2	-60.7
				20	240.2	9.8	561.4	1087.3	-61.1	237.6	8.4	595.1	1118.5	-65.9
				5	1209.4	53.1	2806.0	5303.1	-201.6	1231.8	54.8	3201.2	5247.3	-201.0
	73	1	100	200	1225.5	47.7	2844.3	5475.2	-216.9	1223.4	50.2	2742.8	5605.5	-241.3
				100	1231.0	58.2	2422.0	5415.1	-196.3	1237.0	52.3	2533.2	5293.4	-260.2
				50	1218.7	48.6	2403.6	5394.4	-241.5	1221.4	55.2	3031.0	5531.3	-300.1
				5	1224.3	47.1	5757.0	5298.7	-224.3	1214.0	50.6	7632.9	5509.1	-250.1
				10	1232.8	51.0	5773.8	5366.5	-200.2	1233.6	47.0	7329.7	5647.0	-238.4
3		10	100	200	1220.2	47.7	6601.9	5529.3	-193.7	1228.6	57.1	8286.1	5204.4	-229.2
				100	1218.7	46.0	6751.6	5235.9	-162.1	1221.4	54.5	10630.4	5531.6	-199.6
				50	1226.1	50.3	8530.0	5464.0	-185.9	1221.7	53.2	11650.8	5107.9	-231.0
				10	1227.7	49.9	8989.0	5470.0	-140.6	1238.6	50.2	11097.4	5608.8	-169.2
				5	1223.1	57.4	8883.3	5289.0	-240.4	1217.1	53.7	14068.7	5466.4	-200.2
85		5	20	200	1219.4	56.8	11522.3	5310.2	-188.8	1217.1	53.8	15947.8	5339.4	-201.2
				100	501.9	19.7	770.4	2343.3	-183.9	492.6	19.3	819.7	2102.0	-175.4
				100	497.7	23.6	619.7	2315.7	-152.6	499.3	21.1	658.0	2189.9	-185.1
				50	501.6	17.6	610.7	2345.8	-174.8	495.1	21.6	555.6	1932.8	-173.2
				20	499.3	24.9	528.7	2215.9	-181.2	501.1	27.6	492.7	1943.5	-175.9
	3	5	200	200	496.7	22.2	1199.6	2282.1	-185.9	497.7	20.9	1227.0	2226.7	-140.4
				100	497.5	19.3	985.1	1957.7	-154.5	495.8	22.2	1271.1	1996.6	-181.7
				50	497.5	21.5	1021.5	2248.9	-179.2	494.1	20.3	943.7	2269.2	-154.5
				20	491.7	24.3	846.6	2058.4	-183.7	492.7	24.1	827.5	2147.9	-171.4
				5	502.1	20.3	1378.0	2068.5	-151.9	495.0	20.2	1708.2	2041.3	-186.3
	97	1	100	200	490.4	21.5	1313.0	2179.8	-187.6	501.4	20.2	1296.2	2199.3	-162.5
				100	500.0	22.3	1173.1	2070.1	-184.5	496.9	18.2	1235.6	1980.2	-185.4
				50	500.2	16.8	1159.8	2192.4	-184.8	489.5	20.4	1318.6	1987.8	-165.6
				5	239.2	10.0	363.9	1117.8	-164.2	243.4	10.0	343.2	871.8	-165.8
				10	236.6	10.4	292.9	1087.3	-155.8	236.6	9.6	349.6	867.2	-168.3
3		10	100	200	240.5	9.1	272.8	1118.3	-167.8	241.5	9.3	297.6	1144.5	-114.2
				100	242.2	8.7	276.6	1082.2	-162.0	243.6	10.1	276.6	873.7	-146.5
				5	236.3	9.3	444.5	1052.3	-165.6	242.5	10.0	510.7	1061.5	-148.0
				10	248.0	9.9	399.3	1090.3	-157.4	239.8	10.2	448.1	1043.3	-186.1
				20	240.9	9.5	270.8	871.4	-162.8	242.5	9.9	438.3	1074.5	-169.6
5		5	200	200	237.9	9.6	418.0	867.0	-169.8	243.6	9.3	364.8	874.4	-180.4
				100	242.4	10.6	618.8	873.4	-181.7	253.5	9.5	452.4	1128.9	-154.9
				10	241.6	9.8	561.9	1051.4	-166.7	237.9	9.4	335.4	869.3	-166.6
				20	246.2	11.3	424.4	1070.8	-158.6	239.1	10.2	425.3	1036.6	-180.5
				50	240.9	11.0	416.3	1125.7	-180.0	242.8	10.3	336.4	873.0	-166.9

μ is average
 σ is standard deviation
CPU time is in seconds

4.7. PUBLICATION

Table 4.5 – Results of Cluster 1

Instance	K	Ω	M	\mathcal{H}						\mathcal{H}^*					
				SAA			EVPI	VSS	SAA			EVPI	VSS		
				μ	σ	CPU Time			μ	σ	CPU Time				
49	1	5	200	489.7	19.8	1008.6	2315.9	-282.4	497.4	20.6	668.3	2096.4	-280.5		
			100	493.2	22.1	790.5	2040.8	-292.8	496.1	19.0	614.3	2053.7	-263.3		
			50	489.9	22.3	782.6	2082.5	-232.7	500.0	20.4	692.0	2155.1	-273.9		
			20	503.9	17.7	725.3	2057.8	-239.9	497.2	20.6	706.1	2271.9	-283.5		
			5	497.7	22.0	1528.8	2026.5	-279.7	499.5	20.2	1450.4	1791.6	-259.1		
	3	10	100	200	493.9	21.1	1288.5	2048.1	-187.1	487.7	20.4	1227.0	2064.5	-179.5	
				50	500.7	18.4	1314.7	2058.0	-268.3	492.9	23.3	1207.5	2101.5	-179.5	
				20	489.1	16.9	1383.0	2095.0	-276.9	505.3	25.9	1214.8	2327.2	-291.8	
				5	480.7	21.4	5917.0	2097.4	-286.7	484.7	21.3	2155.5	2078.7	-262.9	
				10	498.0	21.0	5431.5	2105.4	-194.4	496.7	19.7	1888.1	2034.1	-260.2	
	5	20	50	200	501.4	19.5	4459.2	1795.4	-268.1	480.5	20.7	2099.5	2121.1	-237.1	
				100	489.9	22.0	2969.9	2080.8	-272.3	495.5	20.1	2245.9	2326.8	-229.3	
				50	243.4	9.7	267.5	874.4	-126.5	246.6	10.5	333.6	873.8	-128.4	
				20	239.1	10.4	315.2	1113.5	-136.9	243.0	9.9	308.7	1133.9	-130.3	
				5	240.5	9.5	329.9	871.2	-131.4	239.4	10.7	256.3	870.7	-132.3	
61	3	50	200	244.3	10.0	400.2	874.2	-123.9	247.0	10.2	309.3	1142.1	-124.4		
			100	242.8	10.4	495.4	874.9	-129.4	241.4	9.3	566.7	1141.2	-128.8		
			50	239.7	11.5	494.1	1122.6	-135.8	234.1	9.6	439.9	868.9	-129.8		
			20	241.2	11.0	463.4	1176.2	-135.1	243.6	9.2	410.7	875.5	-124.3		
			5	237.6	9.6	364.2	869.4	-137.9	241.3	11.5	440.7	1098.7	-133.2		
	5	20	100	200	241.7	10.1	895.1	868.5	-133.2	241.7	9.9	680.5	870.0	-130.9	
				100	241.0	10.0	803.8	1060.2	-132.9	236.4	10.7	701.1	1111.4	-136.6	
				50	237.5	9.0	701.2	868.1	-134.9	243.0	9.0	566.6	1092.8	-129.6	
				20	246.6	10.7	660.8	876.0	-127.4	236.6	10.5	507.1	872.8	-131.5	
				5	1220.1	51.6	1740.5	4947.6	-855.6	1230.2	52.3	1602.8	4974.8	-835.0	
	73	1	100	200	1214.1	57.6	1681.3	5338.2	-854.2	1224.6	55.9	1432.6	5103.3	-834.8	
				50	1207.3	47.7	1344.7	4409.1	-843.2	1229.2	50.4	1501.9	5565.9	-839.3	
				20	1236.6	61.0	1326.0	4993.6	-845.9	1220.0	43.8	1515.1	4940.6	-843.3	
				5	1223.1	50.8	3612.4	5054.1	-848.1	1222.0	49.0	3155.6	5092.7	-846.4	
				10	1229.2	55.3	3065.3	4945.5	-826.6	1224.2	47.7	3009.2	5180.0	-846.8	
3		20	50	200	1224.8	48.6	3516.6	4966.4	-817.9	1217.7	45.0	4726.1	5044.6	-857.7	
				100	1225.6	55.3	3870.4	4697.2	-845.8	1211.9	51.8	4366.4	4857.9	-854.1	
				50	1217.3	51.3	5139.9	5343.3	-847.0	1239.1	53.0	5021.3	5063.7	-822.0	
				5	1225.0	49.1	4431.5	4987.7	-842.3	1228.4	48.9	5279.9	4934.7	-829.8	
				20	1224.1	53.0	5325.7	4939.7	-858.8	1222.2	51.5	7561.0	5093.6	-836.9	
85		5	100	200	1231.1	49.5	5421.2	5073.2	-838.9	1217.4	52.4	9495.7	5238.9	-848.8	
				100	493.9	23.2	614.6	2033.6	-275.1	495.8	20.6	547.2	1794.5	-272.5	
				10	494.6	20.2	513.2	2063.4	-278.2	496.4	22.5	538.3	2283.7	-272.6	
				50	497.8	22.9	478.4	2099.7	-271.4	498.0	20.5	478.3	2308.7	-283.4	
				20	499.4	19.7	428.1	2108.0	-276.8	492.4	18.7	488.2	2143.8	-278.4	
	3	10	100	200	496.5	19.5	941.6	2052.7	-270.9	500.0	20.6	952.1	2039.1	-269.7	
				100	499.2	18.2	868.1	2157.1	-274.2	496.1	20.2	956.3	2161.4	-274.8	
				20	493.0	18.3	891.9	2066.7	-277.0	500.2	23.3	987.1	2341.5	-271.5	
				50	497.3	19.0	809.7	2321.1	-273.3	488.1	22.3	949.2	2206.7	-280.3	
				5	492.4	20.4	1305.7	2103.5	-284.8	497.9	22.8	1288.5	2284.4	-272.9	
	97	5	100	200	498.7	19.3	992.0	2055.9	-264.5	496.9	21.3	984.5	2103.7	-284.3	
				50	502.5	21.8	1086.4	1799.1	-266.0	491.9	18.9	1047.5	2107.6	-279.9	
				20	488.0	20.5	1074.4	2104.9	-274.8	500.6	23.1	1132.1	2062.6	-266.8	
				5	241.5	10.2	350.2	870.5	-131.2	243.6	9.2	355.2	1056.1	-131.0	
				10	237.7	10.5	310.3	1121.9	-135.8	242.4	10.4	334.0	1089.2	-135.0	
1		100	50	200	243.3	11.5	295.6	1082.6	-130.1	244.1	10.8	272.3	1185.6	-134.8	
				20	239.8	8.6	312.3	1069.6	-132.4	242.8	11.4	296.4	1113.5	-134.2	
				5	240.1	10.5	441.3	1125.9	-134.6	240.5	10.1	458.6	1049.1	-135.4	
				10	242.2	9.4	379.7	873.6	-127.5	240.8	10.2	370.0	1081.4	-128.1	
				20	245.2	9.2	385.0	874.1	-134.9	243.1	10.7	413.7	1075.1	-128.4	
5		20	50	200	243.6	8.9	399.2	875.2	-132.3	246.0	10.2	357.7	874.9	-134.2	
				100	244.9	10.1	525.7	873.4	-131.1	243.9	9.6	610.9	1089.7	-133.0	
				5	239.3	9.2	479.1	1101.2	-136.8	241.5	10.1	414.1	871.5	-130.0	
				20	243.6	9.4	499.8	871.1	-129.7	242.0	10.6	326.3	869.6	-134.5	
				50	239.8	12.1	468.1	872.8	-133.8	242.0	8.5	479.3	1086.6	-137.4	

μ is average
 σ is standard deviation
CPU time is in seconds

CHAPTER 4. STOCHASTIC MULTI-VEHICLE COLLECTION-DISASSEMBLY
PROBLEM

Table 4.6 – Results of Cluster 2

Instance	K	Ω	M	\mathcal{H}					\mathcal{H}^*					
				SAA			EVPI	VSS	SAA			EVPI	VSS	
				μ	σ	CPU Time			μ	σ	CPU Time			
49	1	5	200	503.2	21.0	903.6	2087.0	-262.0	497.2	21.2	723.8	2080.8	-286.7	
			100	486.2	19.9	704.4	2058.3	-250.9	503.6	19.8	595.0	2088.5	-223.8	
			50	497.8	20.3	747.3	2261.4	-282.4	490.5	22.6	613.4	2290.8	-288.6	
			20	502.0	23.5	705.0	2403.0	-284.5	492.4	20.1	637.3	2058.4	-238.1	
			5	488.3	20.8	1730.4	2039.5	-285.3	496.2	21.0	1829.5	1792.2	-286.2	
	3	10	100	500.2	20.4	1532.2	2131.4	-280.8	498.1	21.5	1512.9	2055.2	-284.2	
				20	491.4	22.9	1273.8	2091.8	-228.7	497.6	18.9	1428.7	2019.5	-234.0
				50	490.4	21.0	1360.1	2070.9	-286.9	491.7	20.3	1615.0	2383.7	-181.7
				5	501.0	22.8	6611.2	2032.7	-280.9	497.5	21.4	2387.4	2127.0	-284.1
				10	499.0	19.5	6153.6	2041.8	-201.1	497.9	23.1	2115.3	2143.0	-220.6
	5	20	50	497.8	22.1	4860.0	2073.2	-279.5	496.5	18.7	2017.9	2056.4	-286.8	
				50	502.0	26.9	3893.3	2088.4	-282.6	496.4	17.3	2377.4	2061.3	-283.6
				1	241.8	10.8	485.6	1135.4	-100.4	238.2	10.1	414.6	1107.3	-101.1
				10	239.8	10.1	362.1	872.2	-103.0	247.2	9.5	375.1	1061.0	-98.6
				20	240.5	9.4	365.2	872.5	-105.3	239.6	12.1	381.7	1047.8	-104.1
61	3	20	50	243.5	8.8	338.1	873.4	-105.7	242.3	9.4	336.6	1113.6	-103.0	
			5	239.2	10.5	619.9	867.9	-106.4	243.0	9.4	771.4	1147.5	-97.2	
			10	239.8	9.9	593.7	1117.1	-108.2	242.7	9.2	645.1	1113.0	-102.0	
			20	233.7	9.5	576.5	1105.4	-109.1	241.9	9.3	482.1	873.8	-101.4	
			50	240.4	8.5	531.9	1086.8	-104.8	245.8	10.2	621.5	1131.1	-101.0	
	5	200	5	241.1	11.0	915.1	1098.3	-97.0	242.5	9.9	924.2	873.1	-101.3	
			10	244.5	10.0	741.8	873.9	-99.9	242.6	10.3	738.5	873.6	-96.2	
			20	232.3	10.3	717.4	1166.9	-108.9	239.6	8.7	691.3	1052.6	-100.8	
			50	239.1	9.5	598.3	870.6	-102.6	239.2	8.5	762.1	1154.8	-102.5	
			1	1218.5	50.7	1916.9	5325.8	-750.5	1222.5	50.1	2118.9	5597.6	-657.6	
	73	1	100	100	1215.6	53.2	1675.4	5474.6	-674.6	1223.9	53.2	1807.7	5323.1	-766.9
				20	1217.4	47.6	1422.6	5005.3	-764.1	1229.1	48.0	1590.1	5264.0	-603.5
				50	1216.8	48.5	1665.1	5241.3	-692.1	1225.1	51.4	1652.7	5306.2	-611.0
				5	1224.7	50.3	3494.5	4928.7	-601.3	1222.2	51.1	3673.4	4646.0	-761.1
				10	1217.9	54.0	2626.0	4697.1	-593.5	1232.7	47.7	3798.4	5025.8	-748.2
3		20	50	20	1223.7	52.2	3017.7	4937.5	-753.6	1224.2	51.8	3873.3	5311.4	-753.6
				50	1230.7	57.8	3009.5	4421.4	-743.2	1226.1	41.6	4485.9	5418.4	-580.2
				5	1227.7	53.1	3927.8	5529.3	-701.0	1219.2	50.1	5221.7	5079.4	-673.3
				10	1230.6	48.0	3550.0	5260.1	-689.0	1214.7	50.9	5099.8	5228.4	-690.6
				20	1230.0	54.0	4700.4	5227.5	-750.2	1225.2	54.4	7049.2	4893.5	-757.3
85		1	200	50	1217.3	52.1	5893.3	5285.3	-749.6	1213.6	56.9	8144.8	4908.1	-681.3
				5	495.4	21.3	494.8	2104.7	-234.9	494.0	19.8	628.3	2061.8	-241.4
				10	490.0	23.0	531.8	2379.4	-211.4	494.7	21.8	550.9	2035.6	-240.9
				20	490.0	21.4	493.4	2289.1	-242.5	500.6	18.2	478.8	2030.0	-230.7
				50	494.6	20.4	398.1	1791.3	-246.2	495.5	21.2	468.9	2118.6	-240.2
	3	5	200	498.4	19.5	766.4	1794.4	-267.3	501.6	19.7	1035.1	2112.9	-235.9	
				10	501.4	22.2	760.4	2065.5	-240.3	494.1	20.0	996.7	1789.6	-223.7
				20	493.1	22.8	665.2	2289.1	-245.0	496.1	17.6	782.9	2039.4	-240.8
				50	492.1	20.3	679.9	2311.4	-225.5	498.3	19.0	751.7	2109.4	-225.5
				5	489.9	21.3	1186.9	2301.0	-240.3	497.1	20.3	1337.8	2129.0	-240.3
	97	1	100	493.6	21.7	1047.8	2258.6	-234.1	492.4	18.8	1291.7	2326.4	-231.2	
				20	491.1	20.8	948.3	2346.6	-240.5	493.9	20.9	1023.7	2044.0	-240.0
				50	495.1	21.8	965.2	2047.6	-232.4	493.3	18.8	1044.3	2112.2	-242.2
				5	245.8	11.7	234.8	1125.0	-140.5	241.4	9.7	360.1	872.7	-138.5
				10	246.4	9.9	177.5	874.0	-138.9	239.0	10.7	319.7	869.7	-143.9
3		20	50	243.1	11.7	206.8	869.9	-141.9	247.2	9.5	291.9	876.2	-133.2	
				50	240.0	9.0	164.4	871.2	-138.7	241.0	9.5	289.4	1064.2	-103.6
				5	244.4	9.1	311.5	873.8	-124.6	241.7	9.3	347.6	1140.1	-135.9
				10	242.5	10.6	268.2	871.3	-139.2	244.8	9.7	325.3	1165.4	-138.3
				20	243.6	9.0	261.5	874.1	-138.6	244.0	9.8	273.4	872.6	-138.3
5		20	50	236.0	12.3	271.5	1097.1	-160.7	239.2	9.6	286.9	1161.7	-143.1	
				5	242.8	10.5	324.1	873.1	-135.3	246.7	10.3	418.5	1116.5	-139.0
				10	244.5	9.6	322.8	874.7	-137.3	242.2	9.5	392.5	1119.0	-135.4
				20	242.6	11.1	318.5	875.9	-130.4	241.5	9.1	352.7	1170.4	-138.2
				50	246.9	9.4	286.7	875.5	-137.3	248.1	8.4	361.4	872.6	-142.5

μ is average
 σ is standard deviation
CPU time is in seconds

4.7. PUBLICATION

Chapter 5

Integrated Procurement-Disassembly Problem

Contents

5.1	Introduction	93
5.2	Problem Definition	94
5.3	Formulation	95
5.4	Instance Generation	97
5.5	Numerical Experiments	97
5.6	Conclusions	98
5.7	Publication	98

5.1 Introduction

This chapter provides another problem integrating collection and disassembly processes. Particularly, the collection decisions concern about the routing problem while the disassembly decisions focus on the disassembly line balancing. This problem is so called Integrated Procurement-Disassembly Problem.

The assumption is that the disassembly line is flexible enough to adapt the routing decisions. It is supported by the fact the disassembly process is often performed manually.

This chapter is intended to deal with the collection and the disassembly process of EOL products. It proposes an integrated model for Vehicle Routing Problem based EOL product collection (VRP) and Disassembly Line Balancing Problem (DLBP) called integrated procurement-disassembly problem. The model minimises the total cost corresponding to the collection of EOL products and their disassembly process related to VRP and DLBP, respectively. During products collection process, we assume only one vehicle available and several suppliers who have EOL product to be collected. The model

admits partial disassembly since it aims to release the demanded parts only. To the best of our knowledge, it is the first work proposes the integration between collection process and disassembly balancing problem.

5.2 Problem Definition

Before being disassembled, the EOL products are collected from suppliers to disassembly plant. Subsequently, the products are stored at the inventory and will be disassembled in order to release the demanded parts. A single vehicle and a product type are considered. The data concerning product collection and disassembly process are known and deterministic.

In this section, an linear programming model of integrated procurement-disassembly problem is presented. The problem is defined on weighted and undirected graph network $G = (N, E, D)$. N is the set of node denoting the considered suppliers and the inventory of facility plant. This inventory is denoted as node 1. E is the set of edges and D_{ab} is the distance between node a and node b where $D_{ab} = D_{ba}$, $a \in N$, $b \in N$. The plant disassembles a single product type during planning horizon T . A single vehicle with capacity C and unit running cost RC is used for collecting EOL products from suppliers. A supplier a has certain amount of EOL products at period t denoted as S_{at} . A single vehicle visits each supplier at most once for each period. The collected products are stored at the inventory with unlimited capacity where the inventory level at the end of period t is denoted as I_t , $t \in T$.

At the beginning of period t , the disassembly process begins based on the required demand of part l denoted by d_{lt} , $l \in L$, $t \in T$. Our model requires binary variable d_{lt}^b , $l \in L$, $t \in T$ which is equal to 1 if d_{lt} is greater than 0. Since it is assumed that each product consists of exactly one part of type l , the minimum amount of products stored at inventory d_t^{max} , $t \in T$, before performing disassembly process at the beginning of period is the biggest amount for all required part at period t .

Our model adopts the AND/OR graph (AOG) in [Koc et al. \(2009\)](#) where auxiliary node A_k , $k \in K$ represents subassembly and basic node B_i , $i \in I$ denotes disassembly task. The relations between subassemblies and disassembly tasks are presented in AOG. The dummy task \mathbf{s} is introduced to indicate that disassembly process at period t is finished due to the consideration of partial disassembly. P_k is the set of tasks which precedes subassembly A_k , $k \in K$. S_k , $k \in K$, denotes the set of tasks which succeeds A_k , $k \in K$. Our model requires the set of tasks which permits to release part l , $l \in L$, denoted as P_l .

Disassembly task times $time_i$ are known where the time of dummy task \mathbf{s} , $time_s$ is 0. At period t , each required disassembly is assigned to a workstation. Workstation time is less than the given cycle time CT . FC denotes the fixed cost of opening a workstation. In our model, the decision variables are:

- I_t inventory level at the end of period t

- Y_{at} cumulative load of vehicle after visiting node a at period t
- $X_{abt} \begin{cases} 1 & \text{if the vehicle visits node } a \text{ just before node } b \text{ at period } t \\ 0 & \text{otherwise.} \end{cases}$
- $x_{ijt} \begin{cases} 1 & \text{if disassembly task } i \text{ is assigned to workstation } j \text{ at period } t \\ 0 & \text{otherwise.} \end{cases}$
- $z_{jt} \begin{cases} \text{CT if } x_{s_jt} = 1 \\ 0 & \text{otherwise.} \end{cases}$

5.3 Formulation

$$\text{Min } Z = RC \sum_{t \in T} \sum_{b \in N} \sum_{a \in N} D_{ab} \cdot X_{abt} + FC \sum_{t \in T} \sum_{j \in J} j \cdot z_{jt} \quad (5.1)$$

Subject to:

$$I_t = I_{t-1} + \sum_{b \in N} \sum_{a \in N \setminus \{1\}} X_{abt} \cdot S_{at} - d_t^{max} \quad \forall t \in T, a \neq b \quad (5.2)$$

$$\sum_{a \in N} X_{abt} \leq 1 \quad \forall b \in N \setminus \{1\}, \forall t \in T, a \neq b \quad (5.3)$$

$$\sum_{a \in N} X_{act} = \sum_{b \in N} X_{cbt} \quad \forall c \in N, \forall t \in T, c \neq a, c \neq b \quad (5.4)$$

$$\sum_{a \in S} \sum_{b \in S} X_{abt} \leq |S| - 1 \quad \forall t \in T, \forall S \subseteq N \setminus \{1\} : |S| \geq 2 \quad (5.5)$$

$$Y_{bt} - S_{bt} - Y_{at} \geq - \left(\sum_{l \in L} d_{lt} \right) \cdot (1 - X_{abt}) \quad \forall a \in N, \forall b \in N \setminus \{1\}, \forall t \in T, a \neq b \quad (5.6)$$

$$Y_{bt} - S_{bt} - Y_{at} \leq \left(\sum_{l \in L} d_{lt} \right) \cdot (1 - X_{abt}) \quad \forall a \in N, \forall b \in N \setminus \{1\}, \forall t \in T, a \neq b \quad (5.7)$$

$$Y_{at} \leq C \quad \forall a \in N, \forall t \in T \quad (5.8)$$

$$Y_{1t} = 0 \quad \forall t \in T \quad (5.9)$$

$$\sum_{j \in J} \sum_{i \in P_l} x_{ijt} \geq d_{lt}^b \quad \forall l \in L, \forall t \in T \quad (5.10)$$

$$\sum_{j \in J} \sum_{i \in S_0} x_{ijt} = 1 \quad \forall t \in T \quad (5.11)$$

$$\sum_{j \in J} x_{ijt} \leq 1 \quad \forall i \in I, \forall t \in T \quad (5.12)$$

$$\sum_{i \in S_k} x_{ivt} \leq \sum_{i \in P_k} \sum_{j=1}^v x_{ijt} \quad \forall k \in K \setminus \{0\}, \forall v \in J, \forall t \in T \quad (5.13)$$

$$\sum_{i \in S_k} \sum_{j \in J} x_{ijt} \leq \sum_{i \in P_k} \sum_{j \in J} x_{ijt} \quad \forall k \in K \setminus \{0\}, \forall t \in T \quad (5.14)$$

$$\sum_{j \in J} x_{s_j t} = 1 \quad \forall t \in T \quad (5.15)$$

$$\sum_{j \in J} j \cdot x_{ijt} \leq \sum_{j \in J} j \cdot x_{s_j t} \quad \forall i \in I, \forall t \in T \quad (5.16)$$

$$z_{jt} = CT \cdot x_{s_j t} \quad \forall j \in J, \forall t \in T \quad (5.17)$$

$$\sum_{i \in I} x_{ijt} \cdot time_i \leq CT \quad \forall j \in J, \forall t \in T \quad (5.18)$$

$$I_t \geq 0 \quad \forall t \in T \quad (5.19)$$

$$X_{abt} \in \{0, 1\} \quad \forall a \in N, \forall b \in N, \forall t \in T, a \neq b \quad (5.20)$$

$$Y_{at} \geq 0 \quad \forall a \in N, \forall t \in T \quad (5.21)$$

$$x_{s_j t}, x_{ijt}, \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (5.22)$$

$$z_{jt} \in \{0, CT\} \quad \forall j \in J, \forall t \in T \quad (5.23)$$

The objective function (5.1) aims to minimize the total cost consisting total procurement cost and total cost of opening workstations for entire planning horizon. Constraint set (5.2) balances the plant inventory at each period. At each period, constraint set (5.3) imposes that each supplier is visited at most once. Constraint set (5.4) guarantees that the vehicle leaves a node after visiting it. Constraint set (5.5) eliminates the subtour occurrence. Constraint sets (5.6 - 5.7) update the vehicle load after visiting a node at each period. Constraint set (5.8) limits the vehicle load during its trips. After leaving the plant as depot of the trip, constraint set (5.9) resets the vehicle load as zero. Constraints sets (5.10) describe the relation between part demand and its predecessors. Constraint sets (5.11 - 5.18) are simplification of the model in [Bentaha et al. \(2013b\)](#). Constraint set (5.11) selects the first tasks succeeding EOL product. Constraint set (5.12) assigns the disassembly task into at most a workstation. The precedence relations between disassembly tasks and subassemblies are described by constraint set (5.13). Constraint set (5.14) selects only one OR successor. Constraint set (5.15) assigns sink node into a workstation. Constraint set (5.16) guarantees that each disassembly task is assigned into a workstation with lower or equal index of sink node's workstation. The value of z_{jt} is determined by constraints set (5.17). Constraint set (5.18) denotes the limitation of workstation time. Constraint sets (5.19 - 5.23) describe the nature of decision variables.

5.4 Instance Generation

Since no benchmark instance exists for this problem, we considered the following example based on [Bentaha et al. \(2013a\)](#). A compass consisting seven parts is studied. Ten tasks permits to release one or some parts. At first period, the plant has 20 products available in the inventory. The cycle time is 0.61 second. The setup cost of workstation is €7 / second. A vehicle with 5000 capacity is used with running cost as €5 / km. [Table 5.1](#) and [Table 5.2](#) present the data of demand, part and supplier.

Table 5.1 – Part and Demand Data

Part	Predecessor	Demand(*1000)			
		t = 1	t = 2	t = 3	t = 4
1	3,5,7,9	2	6	2	2
2	7,9	0	3	6	0
3	3,9,10	4	0	5	2
4	2,4,8	1	4	0	1
5	2,4,8	2	7	3	7
6	1,6,10	0	1	0	0
7	1,6,10	4	2	6	2

Table 5.2 – Supplier Data

Node	Coordinate		Supply(*1000)			
	X	Y	t = 1	t = 2	t = 3	t = 4
Depot	30	40	-	-	-	-
Supplier 1	37	52	1	1	4	4
Supplier 2	49	49	5	3	1	1
Supplier 3	52	64	4	1	2	2
Supplier 4	20	26	1	1	3	3

5.5 Numerical Experiments

The model of Integrated Procurement-Disassembly Problem was implemented in Java 7 using GNU Linear Programming Kit (GLPK) 4.9 on a PC with processor Intel® Core™ i7 CPU 2.9 GHz and 4 GB of RAM under Windows 7 Professional.

The optimal solution is obtained in 85.25 seconds with the total cost of €1232.83. Only 2 workstations are opened during four periods considered. The vehicle's trips and disassembly tasks assignment are presented in [Table 5.3](#) and [Table 5.4](#), respectively.

Table 5.3 – Vehicle Trip

Vehicle Trip	Period			
	1	2	3	4
First trip	1, 4, 2, 1	1, 3, 1	1, 2, 1	1, 2, 1
Second trip	1, 3, 1	-	1, 4, 1	-

Table 5.4 – Task Assignment

Workstation	Period			
	1	2	3	4
1	2,6	1	1	2,6
2	9	4,9	4,9	9

5.6 Conclusions

This work addresses integrated procurement-disassembly problem. It combines VRP and DLBP for collecting and disassembling the EOL product. A capacitated vehicle collects EOL product from suppliers. The vehicle begins its trip with zero load. Its capacity forces the vehicle to return back into the inventory for disposing its load. If the inventory level of EOL products is sufficient, the disassembly process begins releasing the demanded parts. The proposed model considers partial DLBP under deterministic condition with single product type.

The objective function minimises the total cost of product collection and disassembly process through vehicle routing determination and disassembly task assignment. The model takes into account the constraints of DLBP, VRP and the balancing constraints coordinating these problems.

It is the first attempt to integrate the collection of EOL products and its disassembly line balancing. The results show the feasibility of such integration. For future work, more complex products and disassembly tasks may be considered.

5.7 Publication

This chapter is based on the following article:

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Integrated procurement-disassembly problem, In: *Advances in Production Management Systems: Innovative and Knowledge-Based Production Management in a Global-Local World*, Part II, B. Grabot, B. Vallespir, S. Gomes, A. Bouras, D. Kiritsis (Eds.), Series: IFIP Advances in Information and Communication Technology, Springer, vol. 439, 2014, ISSN: 1868-4238, p. 382–390. (IFIP WG 5.7 International Conference, APMS)

CHAPTER 5. INTEGRATED PROCUREMENT-DISASSEMBLY PROBLEM

2014, Ajaccio, Corsica, France, September 20-24, 2014).

5.7. PUBLICATION

Chapter 6

General Conclusions

This dissertation aims to propose decision support tools to better manage the implementation of circular economy due to the drawbacks of the linear economy. In detail, this dissertation focuses on the reverse supply chains (RSC) particularly on the collection of End-of-Life (EOL) products. We notably interest on the integration of both the collection of EOL products and their disassembly. Our hypothesis is that integrating decisions of two or more functions in RSC simultaneously leads to better decisions.

The integration of decisions associated with the two processes (collection and disassembly) in RSC is studied in this dissertation. It addresses new problems called Collection-Disassembly Problem (CDP) and Integrated Collection-Procurement Problem. The first problem incorporates the collection routing of EOL products and its disassembly lot-sizing decisions. A stochastic version CDP is also provided to deal with the uncertainty of some parameters as a two-stage stochastic problem. The second problem is an integration between decisions related to the collection routing problem and disassembly line balancing problem.

For CDP, we initially address two different formulations in which the first one uses integrated approach for both problems and the second one is to optimise them separately. Some instances were generated to facilitate this comparison by conforming some setted rules. Using the instances, both problems (integrated and non-integrated) were solved with the commercial solver CPLEX. Based on the obtained results, we found that the integrated approach proposes efficiency in terms of total cost.

However, the solver CPLEX is unable to provide optimal solutions for CDP under acceptable CPU times for large size instances. To deal with the issue, six approximate methods are proposed: Two-Phase Iterative Heuristic, Two-Phase Iterative Heuristic with Enhancements, Local Search-based Algorithm, Particle Swarm Optimisation, Particle Swarm Optimisation with Local Search Enhancements and Imperialist Competitive Algorithm. Based on the obtained results, Two-Phase Iterative Heuristic with Enhancements provides the best performance in terms of global average gaps and CPU times. Parameters of location, nodes, periods, components and initial stock level have significant impact on gaps and CPU times of the proposed methods.

An extension of the deterministic formulation of CDP is proposed consecutively in order to enhance its applicability. It deals with the uncertainty of the quantity and the quality of EOL products, the demands of their components in the case of multi-vehicle. The quality of EOL products is translated as the quantity of their components. The corresponding problem is called Stochastic Multi-Vehicle Collection-Disassembly Problem (SMCDP). This problem is formalised as a two-stage stochastic problem in which the realisation of the uncertain parameters of supply, number of components and demands of components comes after the planning stage. It also means that the first-stage decision variables are taken before we know the value of these parameters.

To solve the SMCDP, two solution algorithms are proposed: Two-Phase Iterative Heuristic and Two-Phase Iterative Heuristic with Enhancements, due to their flexibilities and performances in solving CDP. The methods are combined with an algorithmic framework of Sample Average Approximation (SAA) to provide high quality solutions. According to the results obtained, the Two-Phase Iterative Heuristic with Enhancements has no significant performance than the other one. However, it requires longer average CPU time than Two-Phase Iterative Heuristic for some data sets and shorter CPU time for the remaining data sets.

Apart from Collection-Disassembly Problem, we also studied another integrated problem of collection and disassembly process. For disassembly process, we solve as disassembly line balancing problem. The feasibility of this integration is shown.

Some prospects are addressed to further extend this work. For deterministic CDP, the other type of subtour elimination constrains apart from constraints (2.5) and some valid inequalities may be integrated to propose better lower bounds. It is highly likely that such a problem deals with multi-vehicle and multi-EOL product. Therefore, additional indexes regarding those factors are required to be adapted in the model described in Chapter 2. Apart from IM-MultiTSP adapted in Chapter 3, the version of IM-VRP can be implemented and compared to the Two-Phase Iterative Heuristic and the one with enhancements. Unlike IM-MultiTSP, the method of IM-VRP would propose that the assignment of collection centres visited by available vehicles has to be taken into account during routing construction and excluded from the lot-sizing problem with approximate visiting costs.

As mentioned, the realisation of parameters of the quantity of EOL products available at collection centres, the quantity of components on each EOL product and the demands of component in SMCDP comes after the planning stage. We strongly believe that such realisation of second-stage decisions may also occur in each period as depicted in Figure 32. Its implementation is illustrated in Figure 33. To formalise the problem, the formulation of SMCDP can be extended with additional constraints ensuring the consistency of the inventory level, products disassembled and unmet demands between scenarios.

Furthermore, companies commonly deals with more than one product. It indicates that extending SMCDP by dealing with multi-products is highly possible. Consequently,

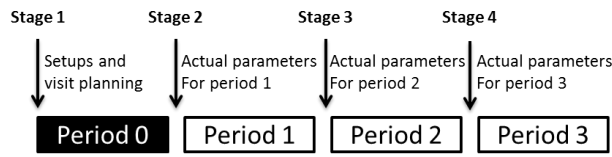


Figure 32 – Multi-Stage Stochastic Problem

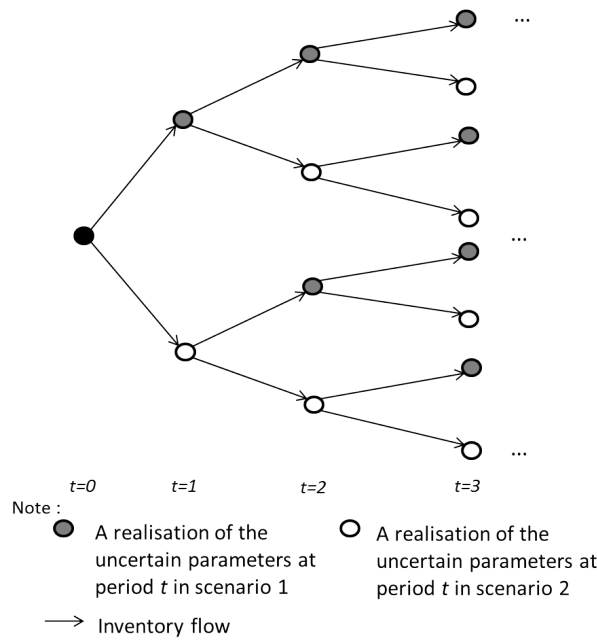


Figure 33 – Multi-Stage SMCDP with Two Scenarios

additional index of products needs to be incorporated in terms of formulation. Due to its flexibility and performance, Two-Phase Iterative Heuristic and its enhanced version combined with the rollout algorithm (Bertsekas et al., 1997) may also be implemented to tackle such problem.

Some extended instances are required to investigate the formulation of the Integrated Procurement-Disassembly Problem in Chapter 5. If the commercial solver is not able to provide optimal solutions under acceptable CPU times, approximate methods may also be considered.

All these prospects are feasible but require more time to adapt the mathematical formulations and the solving methods. However, this dissertation provides some foundations to start.

Appendix A

Publications

Our scientific productions based on this dissertation have been published and submitted into international journals, conferences and seminars as follows:

Peer-Reviewed Journals

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Collection-Disassembly Problem in Reverse Supply Chain, *International Journal of Production Economics*, vol. 183, 2017, p. 334-344
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. An Efficient Two-Phase Iterative Heuristic for Collection-Disassembly Problem, *Computers & Industrial Engineering*, vol. 110, 2017, p. 505-514
- Habibi, M. K. K., Battaïa, O., Cung, V.-D., Dolgui, A., Tiwari, M. K. Sample Average Approximation for Multi-Vehicle Collection-Disassembly Problem under Uncertainty. Submitted to and currently under revision in *International Journal of Production Research*.

Post-Conference Publication

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Coordination of Collection and Disassembly Planning for End-of-Life Product. *IFAC-PapersOnLine*, vol. 48, no. 3, 2015, p. 76–80, doi:10.1016/j.ifacol.2015.06.061.

Book Chapter

- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Integrated procurement–disassembly problem, In: *Advances in Production Management Systems: Innovative and Knowledge-Based Production Management in a Global-Local World*, Part II, B. Grabot, B. Vallespir, S. Gomes, A. Bouras, D. Kiritsis (Eds.), Series: IFIP Advances in Information and Communication Technology, Springer, vol. 439, 2014, ISSN: 1868-4238, p. 382–390. (IFIP WG 5.7 International Conference, APMS 2014, Ajaccio, Corsica, France, September 20-24, 2014).

Conference, Seminar and Workshop without Publication

-
- Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Dealing with the collection and the disassembly planning for simple end-of-life product, Actes des Journées Nationales/ Doctorales MACS, June 18th-19th, 2015, Bourges, France, 5 pages (USB)
 - Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. La coordination de la collecte des produits en fin de vie et de leur désassemblage dans les chaînes logistiques inversées. La 20^{ème} journée du pôle Sciences et Techniques de la Production (STP) du GdR MaCS, February 5th, 2015, Troyes, France
 - Habibi, M.K.K., Battaïa, O., Cung, V.-D., Dolgui, A. Combining Procurement and Disassembly Decisions for End-of-Life Product. Le 16^{ème} congrès de la société française de recherche opérationnelle et aide à la décision (ROADEF). February 25th – 26th, 2015, Marseille, France.

International Doctoral Exchange

We also had performed an international exchange at the Department of Industrial and Systems engineering of Indian Institute of Technology, Kharagpur, India under the supervision of Professor Manoj Kumar TIWARI. During this period, we had participated in the following workshop as an invited speaker:

- International Workshop on European Union-India Research & Innovation Partnership for Efficient and Sustainable Freight Transportation (REINVEST). January 8th – 9th, 2016. Indian Institute of Technology, Mumbai, India.

Bibliography

- Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2014). A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4):784–795.
- Achillas, C., Vlachokostas, C., Moussiopoulos, N., and Baniass, G. (2010). Decision support system for the optimal location of electrical and electronic waste treatment plants: a case study in greece. *Waste management (New York, N.Y.)*, 30(5):870–9.
- Adenso-Díaz, B., García-Carbajal, S., and Gupta, S. M. (2008). A path-relinking approach for a bi-criteria disassembly sequencing problem. *Computers & Operations Research*, 35(12):3989–3997.
- Adulyasak, Y., Cordeau, J., and Jans, R. (2014a). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45.
- Adulyasak, Y., Cordeau, J., and Jans, R. (2015a). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867.
- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1):103–120.
- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015b). The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, 55:141–152.
- Alshamrani, A., Mathur, K., and Ballou, R. H. (2007). Reverse logistics: simultaneous design of delivery routes and returns strategies. *Computers & Operations Research*, 34(2):595–619.
- Altekin, F. T., Kandiller, L., and Ozdemirel, N. E. (2008). Profit-oriented disassembly-line balancing. *International Journal of Production Research*, 46(10):2675–2693.

BIBLIOGRAPHY

- Alumur, S. a., Nickel, S., Saldanha-da Gama, F., and Verter, V. (2012). Multi-period reverse logistics network design. *European Journal of Operational Research*, 220(1):67–78.
- Amorim, P., Belo-Filho, M., Toledo, F., Almeder, C., and Almada-Lobo, B. (2013). Lot sizing versus batching in the production and distribution planning of perishable goods. *International Journal of Production Economics*, 146(1):208–218.
- Aras, N. and Aksen, D. (2008). Locating collection centers for distance- and incentive-dependent returns. *International Journal of Production Economics*, 111(2):316–333.
- Aras, N., Aksen, D., and Gönül Tanuğur, A. (2008). Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles. *European Journal of Operational Research*, 191(3):1223–1240.
- Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12):1731–1746.
- Armentano, V., a.L. Shiguemoto, and Lø kketangen, a. (2011). Tabu search with path relinking for an integrated production–distribution problem. *Computers & Operations Research*, 38(8):1199–1209.
- Atashpaz-Gargari, E. and Lucas, C. (2007). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *IEEE Congress on Evolutionary computation*, pages 4661–4667.
- Barba-Gutiérrez, Y., Adenso-Díaz, B., and Gupta, S. (2008). Lot sizing in reverse MRP for scheduling disassembly. *International Journal of Production Economics*, 111(2):741–751.
- Bard, J. F. and Nananukul, N. (2009). Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, 57(3):713–723.
- Bard, J. F. and Nananukul, N. (2010). Computers & Operations Research A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers and Operation Research*, 37(12):2202–2217.
- Belfiore, P. and Yoshizaki, H. T. (2013). Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries. *Computers & Industrial Engineering*, 64(2):589–601.

- Bentaha, M., Battaïa, O., and Dolgui, A. (2013a). A stochastic formulation of the disassembly line balancing problem. In *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services*, volume 397, pages 397–404.
- Bentaha, M., Battaïa, O., and Dolgui, A. (2013b). L-Shaped Algorithm for Stochastic Disassembly Line Balancing Problem. In *7th IFAC Conference on Modelling, Management, and Control*, pages 407–411, Saint Petersburg. International Federation of Automatic Control.
- Bentaha, M., Battaïa, O., and Dolgui, a. (2014a). A Sample Average Approximation Method for Disassembly Line Balancing Problem under Uncertainty. *Computers & Operations Research*, 51:111–122.
- Bentaha, M. L., Battaïa, O., and Dolgui, A. (2014b). Disassembly Line Balancing and Sequencing under Uncertainty. *Procedia CIRP*, 15:239–244.
- Bentaha, M. L., Battaïa, O., and Dolgui, A. (2014c). Lagrangian Relaxation for Stochastic Disassembly Line Balancing Problem. *Procedia CIRP*, 17:56–60.
- Bentaha, M. L., Battaïa, O., Dolgui, A., and Hu, S. J. (2014d). Dealing with uncertainty in disassembly line design. *CIRP Annals - Manufacturing Technology*, 63(1):21–24.
- Bentaha, M. L., Battaïa, O., Dolgui, A., and Hu, S. J. (2015). Second order conic approximation for disassembly line design with joint probabilistic constraints. *European Journal of Operational Research*, 247(3):957–967.
- Bertazzi, L., Paletta, G., and Speranza, M. (2005). Minimizing the total cost in an integrated vendor—managed inventory system. *Journal of heuristics*, 11(5):393–419.
- Bertsekas, D., Tsitsiklis, J., and Wu, C. (1997). Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3(3):245–262.
- Boudia, M., Dauzere-Peres, S., Prins, C., and Louly, M. A. O. (2006). Integrated optimization of production and distribution for several products. In *International Conference on Service Systems and Service Management*, pages 272 – 277, Troyes.
- Boudia, M., Louly, M., and Prins, C. (2007). A reactive GRASP and path relinking for a combined production–distribution problem. *Computers & Operations Research*, 34(11):3402–3419.
- Boudia, M. and Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production–distribution problem. *European Journal of Operational Research*, 195(3):703–715.

BIBLIOGRAPHY

- Brandão, J. (2009). A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 195(3):716–728.
- Brown, G., Keegan, J., Vigus, B., and Wood, K. (2001). The Kellogg company optimizes production, inventory, and distribution. *Interfaces*, 31(6):1–15.
- Buer, M. V., Woodruff, D., and Olson, R. (1999). Solving the medium newspaper production/distribution problem. *European Journal of Operational Research*, 115(2):237–253.
- Calvete, H. I., Galé, C., and Oliveros, M.-J. (2011). Bilevel model for production–distribution planning solved by using ant colony optimization. *Computers & Operations Research*, 38(1):320–327.
- Çetinkaya, S., Üster, H., Easwaran, G., and Keskin, B. B. (2009). An Integrated Outbound Logistics Model for Frito-Lay: Coordinating Aggregate-Level Production and Distribution Decisions. *Interfaces*, 39(5):460–475.
- Chandra, P. (1993). A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society*, pages 1–11.
- Chandra, P. and Fisher, M. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517.
- Chen, H.-K., Hsueh, C.-F., and Chang, M.-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research*, 36(7):2311–2319.
- Chouinard, M., D’Amours, S., and Aït-Kadi, D. (2008). A stochastic programming approach for designing supply loops. *International Journal of Production Economics*, 113(2):657–677.
- Dat, L. Q., Truc Linh, D. T., Chou, S.-Y., and Yu, V. F. (2012). Optimizing reverse logistic costs for recycling end-of-life electrical and electronic products. *Expert Systems with Applications*, 39(7):6380–6387.
- Desrochers, M. and Laporte, G. (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36.
- Díaz-Madroño, M., Peidro, D., and Mula, J. (2015). A review of tactical optimization models for integrated production and transport routing planning decisions. *Computers & Industrial Engineering*.
- Dolgui, A. and Proth, J.-M. (2006). Les systèmes de production modernes. Two Volumes London: Hermes Science.

- Dolgui, A. and Proth, J.-M. (2010). *Supply chain engineering: useful methods and techniques*. Springer Science & Business Media.
- Duta, L., Filip, F., and Caciula, I. (2008). Real time balancing of complex disassembly lines. In *Proceeding of the 17th World Congress of The International Federation of Automatic Control*, pages 913–918, Seoul.
- El korchy, A. and Millet, D. (2011). Designing a sustainable reverse logistics channel: the 18 generic structures framework. *Journal of Cleaner Production*, 19(6-7):588–597.
- ElSayed, A., Kongar, E., and Gupta, S. M. (2012). An Evolutionary Algorithm for Selective Disassembly of End-of-Life Products. *International Journal of Swarm Intelligence and Evolutionary Computation*, 1:1–7.
- El-Sayed, M., Afia, N., and El-Kharbotly, a. (2010). A stochastic model for forward—reverse logistics network design under risk. *Computers & Industrial Engineering*, 58(3):423–431.
- Forum, W. E. (2014). Towards the circular economy: Accelerating the scale-up across global supply chains. Technical Report January, World Economic Forum, Geneva.
- Fumero, F. and Vercellis, C. (1999). Synchronized development of production, inventory, and distribution schedules. *Transportation science*, 33(3):330–340.
- Ghilas, V., Demir, E., and Woensel, T. V. (2016). A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. *Transportation Research Part B: Methodological*, 91:34–51.
- Grunow, M. and Gobbi, C. (2009). Designing the reverse network for WEEE in Denmark. *CIRP Annals - Manufacturing Technology*, 58(1):391–394.
- Gungor, A. and Gupta, S. (1999). Disassembly line balancing. In Gupta, S. M., editor, *Proceedings of the 1999 Annual Meeting of the Northeast Decision Sciences Institute*, pages 193–195, Newport, Rhode Island.
- Gupta, S. (2013). *Reverse Supply Chains: Issues and Analysis*. CRC Press, Boca Raton, 1 edition.
- Gupta, S. and Imtanavanich, P. (2009). Evolutionary computational approach for disassembly sequencing in a multiproduct environment. *International Journal of Biomedical Soft Computing and Human Sciences*, 15(1):73–78.
- Hanafi, J., Kara, S., and Kaebernick, H. (2008). Reverse logistics strategies for end-of-life products. *International Journal of Logistics Management*, 19(3):367–388.

BIBLIOGRAPHY

- Jiang, J., Ng, K. M., Poh, K. L., and Teo, K. M. (2014). Vehicle routing problem with a heterogeneous fleet and time windows. *Expert Systems with Applications*, 41(8):3748–3760.
- Kalayci, C. and Gupta, S. (2011). Tabu search for disassembly line balancing with multiple objectives. *Proceedings of the 41st International Conference on Computers and Industrial Engineering (CIE41)*, (617):477–482.
- Kalayci, C. and Gupta, S. (2013a). Simulated annealing algorithm for solving sequence-dependent disassembly line balancing problem. In *7th IFAC Conference on Modelling, Management, and Control*, pages 93–98, Saint Petersburg. International Federation of Automatic Control.
- Kalayci, C. B. and Gupta, S. M. (2013b). Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. *Expert Systems with Applications*, 40(18):7231–7241.
- Kennedy, J. (1995). Particle swarm optimization. In *IEEE International of first Conference on Neural Networks*, pages 167–171.
- Kleywegt, A., Shapiro, A., and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.
- Koc, A., Sabuncuoglu, I., and Erel, E. (2009). Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph. *IIE Transactions*, 41(10):866–881.
- Kumar, R. S., Kondapaneni, K., Dixit, V., Goswami, a., Thakur, L., and Tiwari, M. (2015). Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. *Computers & Industrial Engineering*.
- Kumar, S. and Putnam, V. (2008). Cradle to cradle: Reverse logistics strategies and opportunities across three industry sectors. *International Journal of Production Economics*, 115(2):305–315.
- Lambert, S., Riopel, D., and Abdul-Kader, W. (2011). A reverse logistics decisions conceptual framework. *Computers & Industrial Engineering*, 61(3):561–581.
- Lee, C., Realff, M., and Ammons, J. (2011). Integration of channel decisions in a decentralized reverse production system with retailer collection under deterministic non-stationary demands. *Advanced Engineering Informatics*, 25(1):88–102.

- Lee, D.-H. and Dong, M. (2009). Dynamic network design for reverse logistics operations under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):61–71.
- Lee, J.-E., Gen, M., and Rhee, K.-G. (2009). Network model and optimization of reverse logistics by hybrid genetic algorithm. *Computers & Industrial Engineering*, 56(3):951–964.
- Lin, S. and Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research*, 21(2):498–516.
- McGovern, S. and Gupta, S. (2011). *The disassembly line: balancing and modeling*. McGraw-Hill, New York, 1 edition.
- Mutha, A. and Pokharel, S. (2009). Strategic network design for reverse logistics and remanufacturing using new and old product modules. *Computers & Industrial Engineering*, 56(1):334–346.
- Özceylan, E. and Paksoy, T. (2013). Reverse supply chain optimisation with disassembly line balancing. *International Journal of Production Research*, 51(20):5985–6001.
- Özceylan, E. and Paksoy, T. (2014). Interactive fuzzy programming approaches to the strategic and tactical planning of a closed-loop supply chain under uncertainty. *International Journal of Production Research*, 52(8):2363–2387.
- Özceylan, E., Paksoy, T., and Bektaş, T. (2014). Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing. *Transportation Research Part E: Logistics and Transportation Review*, 61:142–164.
- Pishvaei, M. and Torabi, S. (2010). A possibilistic programming approach for closed-loop supply chain network design under uncertainty. *Fuzzy Sets and Systems*, 161(20):2668–2683.
- Pishvaei, M. S., Farahani, R. Z., and Dullaert, W. (2010). A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Computers & Operations Research*, 37(6):1100–1112.
- Pishvaei, M. S., Jolai, F., and Razmi, J. (2009a). A stochastic optimization model for integrated forward/reverse logistics network design. *Journal of Manufacturing Systems*, 28(4):107–114.
- Pishvaei, M. S., Kianfar, K., and Karimi, B. (2009b). Reverse logistics network design using simulated annealing. *The International Journal of Advanced Manufacturing Technology*, 47(1-4):269–281.

BIBLIOGRAPHY

- Pishvaei, M. S., Rabbani, M., and Torabi, S. A. (2011). A robust optimization approach to closed-loop supply chain network design under uncertainty. *Applied Mathematical Modelling*, 35(2):637–649.
- Pochampally, K. and Gupta, S. (2009). Reverse supply chain design: a neural network approach. In Wang, H.-F., editor, *Web-Based Green Products Life Cycle Management Systems: Reverse Supply Chain Utilization*, chapter XIII, pages 283–300. Information Science Reference, Hershey, 1 edition.
- Qin, Z. and Ji, X. (2010). Logistics network design for product recovery in fuzzy environment. *European Journal of Operational Research*, 202(2):479–490.
- Rafiei, H., Rabbani, M., Vafa-Arani, H., and Khoshnudi, N. (2013). A Heterogeneous Fleet Vehicle Waste Collection Problem with Various Zones and Intermediate Facilities. *International Journal of Engineering Sciences and Research Technology*, 2(5):1251–1259.
- Salhi, S., Imran, A., and Wassan, N. a. (2013). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, pages 1–11.
- Shiguemoto, A. L. and Armentano, V. A. (2010). A tabu search procedure for coordinating production, inventory and distribution routing problems. *International Transactions in Operational Research*, 17(2):179–195.
- Solyali, O., Süral, H., and Neogy, S. (2009). A relaxation based solution approach for the inventory control and vehicle routing problem in vendor managed systems. In SK, N., AK, D., and Bapat, R., editors, *Modeling, Computation and Optimization*, number 2000, pages 171–189, Singapore.
- Tibben-Lembke, R. S. and Rogers, D. S. (2002). Differences between forward and reverse logistics in a retail environment. *Supply Chain Management: An International Journal*, 7(5):271–282.
- Tripathi, M., Agrawal, S., Pandey, M. K., Shankar, R., and Tiwari, M. (2009). Real world disassembly modeling and sequencing problem: Optimization by Algorithm of Self-Guided Ants (ASGA). *Robotics and Computer-Integrated Manufacturing*, 25(3):483–496.
- Xu, Y., Wang, L., and Yang, Y. (2012). A New Variable Neighborhood Search Algorithm for the Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows. *Electronic Notes in Discrete Mathematics*, 39:289–296.
- Yeh, W.-C. (2012). Simplified swarm optimization in disassembly sequencing problems with learning effects. *Computers & Operations Research*, 39(9):2168–2177.

- Zarandi, M. H. F., Sisakht, A. H., and Davari, S. (2011). Design of a closed-loop supply chain (CLSC) model using an interactive fuzzy goal programming. *The International Journal of Advanced Manufacturing Technology*, 56(5-8):809–821.
- Zhao, J. and Zhu, F. (2015). A multi-depot vehicle-routing model for the explosive waste recycling. *International Journal of Production Research*, 54(2):550–563.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2017LYSEM005

Muhammad Khoirul Khakim HABIBI

INTEGRATED OPTIMISATION OF THE COLLECTION OF END-OF-LIFE PRODUCTS AND THEIR
DISASSEMBLY IN REVERSE SUPPLY CHAIN

Speciality : Industrial Engineering

Keywords : Reverse Supply Chain, End-of-Life Products, Disassembly Process, Production-Distribution Problem, Approximate Methods, Stochastic Programming.

Abstract :

This dissertation supports and proposes better management in the implementation of the circular economy by integrating activities of reverse supply chains. The hypothesis states that integrating decisions of at least two activities in reverse supply chain leads to better decisions notably the collection of End-of-Life products and their disassembly.

A deterministic problem, called *Collection-Disassembly Problem*, integrating both collection and disassembly processes as well as its formulation are introduced and developed. Due to lack of available instances in the literature, some instances are generated. Another non-integrated formulation is developed and solved using the commercial solver CPLEX. The obtained results show that the integrated model proposes better decisions in terms of total cost. Some approximate methods are developed because the commercial solver CPLEX is unable to provide optimal solutions under acceptable CPU times notably for large size instances.

An extended version of the problem is introduced due to the fact that reverse supply chains frequently deal with the uncertainty of certain parameters such as the quality and the quantity of End-of-Life products as well as the demands of components. Also, there is often more than one vehicle available to collect the products. Thus, this second problem suggested which is called *Stochastic Multi-Vehicle Collection-Disassembly Problem* and its formulation is developed. The problem is formalised as two-stage stochastic programming by assuming that the parameters under uncertainty follow some known probability distributions and their realisation comes after the planning stage. To provide the solutions, two methods combined with an algorithmic framework of Sample Average Approximation are developed.

Another problem called *Integrated Procurement-Disassembly Problem* is also studied. Along with the decisions on collection process, this problem emphasises on the decisions of disassembly line balancing problem.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2017LYSEM005

Muhammad Khoirul Khakim HABIBI

OPTIMISATION INTÉGRÉE DE LA COLLECTE DE PRODUITS EN FIN DE VIE ET DE LEUR
DÉSASSEMBLAGE DANS UNE CHAÎNE LOGISTIQUE INVERSE

Spécialité : Génie Industriel

Mots clefs : Chaîne logistique inverse, produits en fin de vie, processus de désassemblage, problème de production et distribution, méthodes approchées, programmation stochastique.

Résumé :

Il est connu que l'intégration des décisions dans les chaînes logistiques directes permet de proposer de meilleures décisions. Dans cette thèse, une approche similaire est proposée pour une chaîne logistique inverse. Nous supposons que l'intégration de décisions concernant la collecte des produits en fin de vie avec celles de leur désassemblage permet d'optimiser la chaîne logistique inverse.

D'abord, un problème déterministe, appelé le *problème de collecte et désassemblage*, intégrant les décisions de collecte de produits en fin de vie et la planification de leur désassemblage a été proposé et étudié. Étant donné qu'il n'y a pas d'instance de ce problème dans la littérature, les instances sont générées pour pouvoir effectuer les tests des modèles développés. Une comparaison de la formulation intégrée avec celle non-intégrée est effectuée. Selon les résultats obtenus, il s'avère que l'intégration permet d'optimiser les coûts totaux. Des méthodes approchées sont également proposées car le solveur CPLEX n'est pas capable de fournir les solutions optimales des instances de grand taille avec les temps de calcul acceptables.

Ensuite, le problème est étendu en considérant l'incertitude sur les paramètres associés à la qualité et la quantité des produits en fin de vie et les demandes en composants. En considérant également la possibilité d'avoir plusieurs véhicules pour la collecte de produits en fin de vie, le problème stochastique de collecte et désassemblage dans sa version multi-véhicule est introduit. Sa formulation est développée sous forme d'une programmation stochastique en bi-niveau. Nous supposons qu'au moment de la planification, les paramètres incertains sont considérés comme des variables aléatoires avec des lois de distribution connues. Les réalisations de ces variables ne sont connues qu'au moment d'exécution du plan. Deux méthodes de résolution en utilisant une procédure d'approximation par échantillonnage sont implémentées afin de fournir les solutions.

Finalement, un autre problème est également posé et étudié où les décisions concernant la collecte de produits en fin de vie sont associées au problème de l'équilibrage de ligne de désassemblage.