



Fixed cardinality linear ordering problem, polyhedral studies and solution methods

Rahimeh Neamatian Monemi

► To cite this version:

Rahimeh Neamatian Monemi. Fixed cardinality linear ordering problem, polyhedral studies and solution methods. Other [cs.OH]. Université Blaise Pascal - Clermont-Ferrand II, 2014. English. NNT : 2014CLF22516 . tel-01166339

HAL Id: tel-01166339

<https://theses.hal.science/tel-01166339>

Submitted on 22 Jun 2015

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: D.U: 2516
EDSPIC: 674

UNIVERSITE BLAISE PASCAL - CLERMONT Fd. II

ÉCOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

Thèse

Présentée par

RAHIMEH NEAMATIAN MONEMI

Pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ
SPÉCIALITÉ: INFORMATIQUE

Titre de la thèse:

**Fixed Cardinality Linear Ordering Problem,
Polyhedral Study and Solution Methods**

Soutenue publiquement le **02 Décembre 2014** devant le jury :

M. Mohamed Didi Biha,
M. Philippe Michelon,
M. Ali Ridha Mahjoub,
M. Herve Kerivin,
M. Mourad Baiou,
M. Philippe Mahey,

Président et examinateur
Rapporteur et examinateur
Rapporteur et examinateur
Examineur
Directeur de thèse
Directeur de thèse

Acknowledgement

First of all, I am thankful of those who have been continuously supporting me so far in my life. There is no doubt that all I achieved so far owes to their endless love, tireless efforts, unconditional support and attention. The outcome of this research, whatever it is, has strongly been influenced by many people who accompanied me during the whole or part of the time in my private and academic life. I take this opportunity to express my sincere thanks to them and appreciate their valuable support and devotion.

The most, I am grateful and owing to my family, my mother, Nouriyeh Moravej Torbati and my father, Ahmad Neamatian Monemi. I am strongly grateful and indebted to my husband, Shahin Gelareh, whom without his unsparing endless and kind support; I could never reach this milestone. In particular, I would like to thank Sourena, the God's gift, who has colored my life. I am also indebted forever to my sisters, Zahra, Tahereh, Marziyeh and Raziye, for the creativity and the curiosity that they have developed in my mind and for their endless love and patience and all what they have taught me during my life. I am very much thankful of my uncle Mehdi, for his worthwhile kind support during my life. I would like to thank Mrs. Sedigheh Makaremi and her family for their endless kind support. In particular, I would like to thank my dear friend, Nathalie Marine Bernard, who has taught a lot.

My special thanks go to my PhD advisors, Prof. Philippe Mahey and Dr. Mourad Baiou, for offering me the opportunity and supporting me to carry out this work. Without their support and patience, this dissertation would not reach to this point.

I am also thankful of the PhD jury members, Prof. Ali Ridha Mahjoub from Université Paris Dauphine (Paris 9), Prof. Philippe Michelon from Université d'Avignon et des Pays de Vaucluse, Prof. Mohamed Didi Biha from Université de Caen Basse-Normandie and Dr. Hervé Kerivin from Université Blaise Pascal, who have accepted evaluating my work.

Last but not least, I also would like to thank the LIMOS CNRS research laboratory for offering me a friendly environment and necessary facilities to carry out this work. In particular, I am thankful of Beatrice Bourdieu for offering me her kind and valuable support in the last few year.

Résumé

Le problème d'ordre linéaire (LOP) a reçu beaucoup d'attention dans différents domaines d'application, allant de l'archéologie à l'ordonnancement en passant par l'économie et même de la psychologie mathématique. Ce problème est aussi connu pour être parmi les problèmes NP-difficiles. Nous considérons dans cette thèse une variante de (LOP) sous contrainte de cardinalité. Nous cherchons donc un ordre linéaire d'un sous-ensemble de sommets du graphe de préférences de cardinalité fixée et de poids maximum. Ce problème, appelé (FCLOP) pour 'fixed-cardinality linear ordering problem', n'a pas été étudié en tant que tel dans la littérature scientifique même si plusieurs applications dans les domaines de macro-économie, de classification dominante ou de transport maritime existent concrètement. On retrouve en fait ses caractéristiques dans les modèles étendus de sous-graphes acycliques. Le problème d'ordre linéaire est déjà connu comme un problème NP-difficile et il a donné lieu à de nombreuses études, tant théoriques sur la structure polyédrale de l'ensemble des solutions réalisables en variables 0-1 que numériques grâce à des techniques de relaxation et de séparation progressive. Cependant on voit qu'il existe de nombreux cas dans la littérature, dans lesquelles des solveurs de Programmation Linéaire en nombres entiers comme CPLEX peuvent résoudre certaines instances en moins de 10 secondes, mais une fois que la cardinalité est limitée, ces mêmes instances deviennent très difficiles à résoudre. Sur les aspects polyédraux, nous avons étudié le polytope de FCLOP, défini plusieurs classes d'inégalités valides et identifié la dimension ainsi que certaines inégalités qui définissent des facettes pour le polytope de FCLOP. Nous avons introduit un algorithme Relax-and-Cut basé sur ces résultats pour résoudre les instances du problème. Dans cette étude, nous nous sommes également concentrés sur la relaxation Lagrangienne pour résoudre ces cas difficiles. Nous avons étudié différentes stratégies de relaxation et nous avons comparé les bornes duales par rapport à la consolidation obtenue à partir de chaque stratégie de relâcher les contraintes afin de détecter le sous-ensemble des contraintes le plus approprié. Les résultats numériques montrent que nous pouvons trouver des bornes duales de très haute qualité. Nous avons également mis en place une méthode de décomposition Lagrangienne. Dans ce but, nous avons décomposé le modèle de FCLOP en trois sous-problèmes (au lieu de seulement deux) associés aux contraintes de 'tournoi', de 'graphes sans circuits' et de 'cardinalité'. Les résultats numériques montrent une amélioration significative de la qualité des bornes duales pour plusieurs cas. Nous avons aussi mis en oeuvre une méthode de plans sécants (cutting plane algorithm) basée sur la relaxation pure des contraintes de circuits. Dans cette méthode, on a relâché une partie des contraintes et on les a ajoutées au modèle au cas où il y a des de/des violations. Les résultats numériques montrent des performances prometteuses quant à la réduction du temps de calcul et à la résolution d'instances

difficiles hors d'atteinte des solveurs classiques en PLNE.

mots-clefs

Optimisation, recherche opérationnelle, NP-difficile, programmation mathématiques, programmation nombres entiers, méthodes de résolution exacte, polyèdres, dimension de polytope, facette, relaxation Lagrange, décomposition Lagrange, décomposition Benders, méthode de relax-and-cut, méthode de plans sécants, méthode de Bundle, borne duale

Abstract

Linear Ordering Problem (LOP) has received significant attention in different areas of application, ranging from transportation and scheduling to economics and even archeology and mathematical psychology. It is classified as a NP-hard problem. Assume a complete weighted directed graph on V_n , $|V_n| = n$. A permutation of the elements of this finite set of vertices is a linear order. Now let p be a given fixed integer number, $0 \leq p \leq n$. The p -Fixed Cardinality Linear Ordering Problem (FCLOP) is looking for a subset of vertices containing p nodes and a linear order on the nodes in S .

Graphically, there exists exactly one directed arc between every pair of vertices in an LOP feasible solution, which is also a complete cycle-free digraph and the objective is to maximize the sum of the weights of all the arcs in a feasible solution. In the FCLOP, we are looking for a subset $S \subseteq V_n$ such that $|S| = p$ and an LOP on these S nodes. Hence the objective is to find the best subset of the nodes and an LOP over these p nodes that maximize the sum of the weights of all the arcs in the solution. Graphically, a feasible solution of the FCLOP is a complete cycle-free digraph on S plus a set of $n - p$ vertices that are not connected to any of the other vertices.

There are several studies available in the literature focused on polyhedral aspects of the linear ordering problem as well as various exact and heuristic solution methods.

The fixed cardinality linear ordering problem is presented for the first time in this PhD study, so as far as we know, there is no other study in the literature that has studied this problem.

The linear ordering problem is already known as a NP-hard problem. However one sees that there exist many instances in the literature that can be solved by CPLEX in less than 10 seconds (when $p = n$), but once the cardinality number is limited to p ($p < n$), the instance is not anymore solvable due to the memory issue.

We have studied the polytope corresponding to the FCLOP for different cardinality values. We have identified dimension of the polytope, proposed several classes of valid inequalities and showed that among these sets of valid inequalities, some of them are defining facets for the FCLOP polytope for different cardinality values. We have then introduced a Relax-and-Cut algorithm based on these results to solve instances of the FCLOP.

To solve the instances of the problem, in the beginning, we have applied the Lagrangian relaxation algorithm. We have studied different relaxation strategies and compared the dual bound obtained from each case to detect the most suitable subproblem. Numerical results show that some of the relaxation strategies result better dual bound and some other contribute more in reducing the computational time and provide a relatively good dual bound in a shorter time.

We have also implemented a Lagrangian decomposition algorithm, decom-

posing the FCLOP model to three subproblems (instead of only two subproblems). The interest of decomposing the FCLOP model to three subproblems comes mostly from the nature of the three subproblems, which are relatively quite easier to solve compared to the initial FCLOP model. Numerical results show a significant improvement in the quality of dual bounds for several instances. We could also obtain relatively quite better dual bounds in a shorter time comparing to the other relaxation strategies. We have proposed a cutting plane algorithm based on the pure relaxation strategy. In this algorithm, we firstly relax a subset of constraints that due to the problem structure, a very few number of them are active. Then in the course of the branch-and-bound tree we verify if there exist any violated constraint among the relaxed constraints or. Then the characterized violated constraints will be globally added to the model. Consequently in the worst case, the algorithm will end after a finite (and polynomial) number of iterations. Numerical instances confirm the efficiency of this exact solution algorithm.

Keywords

Optimization, Operations research, NP-hard, mathematical programming, Integer programming, Exact solution method, polyhedral, polytope, dimension, facet defining inequalities, Lagrangian relaxation, Lagrangian decomposition, Benders decomposition, relax-and-cut, cutting plane algorithm, Bundle method, dual bound

Contents

1	Introduction	13
1.1	Literature review - State of the art of the LOP	15
1.1.1	Linear ordering problem and applications	15
1.1.2	Polyhedral studies for the LOP	18
1.1.3	Solution methods	19
1.2	Summary and conclusion	20
2	Modeling the FCLOP	21
2.1	Definitions and notations	22
2.2	Motivation	23
2.2.1	Voting theory and analysis of ordinal data	23
2.2.2	Transportation and communications network design	24
2.2.3	Resource-constrained scheduling problems	24
2.3	Mathematical modeling	25
2.3.1	Linear ordering problem	25
2.3.2	Fixed cardinality linear ordering problem	25
2.4	Summary and conclusion	26
3	Valid Inequalities	27
3.1	Category 1: The properties of underlying graph	28
3.1.1	Node-Degree constraints	28
3.1.2	TotalX constraint	29
3.1.3	Triangle-free inequalities	30
3.2	Category 2: Dynamic symmetry-breaking inequalities	33
3.2.1	Algorithm for finding symmetry-breaking inequalities	35
3.3	Numerical results	37
3.3.1	FCLOP	38
3.3.2	FCLOP + TotalX	40

3.3.3	FCLOP + Node-Degree	42
3.3.4	FCLOP + TotalX + TF Class 1	44
3.3.5	FCLOP + Node-Degree + TF Class 1	45
3.3.6	FCLOP + TF Class 2	47
3.3.7	Bound quality comparison	49
4	Polyhedral Study	55
4.1	Dimension	57
4.2	Facets	62
4.2.1	Trivial inequalities	62
4.2.2	Triangle-free inequalities	71
4.3	Related studies	79
4.4	Conclusion	84
5	Lagrangian Relaxation	85
5.1	Introduction	85
5.2	Complicating constraints for the FCLOP	92
5.2.1	FCLOP-LR-A (relaxing the Tournament inequalities)	95
5.2.2	FCLOP-LR-B (<i>relaxing triangle-free inequalities</i>) . . .	96
5.2.3	FCLOP-LR-C <i>relaxing cardinality constraints</i>	97
5.2.4	FCLOP-LD1 —Two sub-problems	97
5.2.5	FCLOP-LD2 —Three sub-problems	98
5.3	Numerical results	101
5.3.1	FCLOP-LR-A	103
5.3.2	FCLOP-LR-B	105
5.3.3	FCLOP-LR-C	107
5.3.4	FCLOP-LD1	109
5.3.5	FCLOP-LD2	111
6	Relaxation Based Algorithm	127
6.1	Relaxation Based Exact Algorithm for the FCLOP	132
6.1.1	Algorithm	134
6.1.2	Extended pure relaxation algorithm	136
6.1.3	Relaxation Based Algorithm hybrid with Lagrangian Relaxation	138
6.2	Numerical results	139
6.2.1	Pure Relaxation Based Cutting Plane algorithm . . .	139
6.2.2	Extended pure Relaxation Based Cutting Plane Algo- rithm	141
6.2.3	Cutting Plane Algorithm Hybrid with Lagrangian Re- laxation	143
6.2.4	Comparing the three algorithms	144

7	Facets and Numerical Efficiency	147
7.1	Lagrangian relaxation	147
7.1.1	Numerical results	149
7.2	Relax-and-Cut Algorithm	153
7.2.1	Introduction	153
7.2.2	Algorithm	153
7.2.3	Numerical results	154
8	Summary, conclusions and outlook to future work	157
8.1	Summary and conclusions	157
8.2	Future work	159

CHAPTER 1

Introduction

The Linear Ordering Problem(LOP) is a well known problem in combinatorial optimization which is classified as a NP-hard problem by Garey and Johnson (1979).

The linear ordering problem has received many attention in various applications like economics, tournament theory, voting theory, archeology, scheduling, triangulation problem of input-output matrices and even in mathematical psychology.

By definition, every permutation of the elements of a finite set is a linear order. The Linear Ordering Problem (LOP) consists in finding a linear order such that the sum of the preferences between each element and each 'follower' in the order is maximized. In this PhD study, we are presenting a new variation of the linear ordering problem for the case when the number of nodes which are appearing in a feasible solution is limited to p . The cardinality number p can vary between zero and n , so we have $0 \leq p \leq n$. Hereafter, the new variation of linear ordering problem with fixed (limited) cardinality number will be referred by Fixed Cardinality Linear Ordering Problem (FCLOP).

To our present knowledge, it is the first investigation of this problem with respect to combinatorial optimization, polyhedral representation and algorithmic issues.

In this PhD study, we introduce in detail the new extension of the linear ordering problem that will be refereed as the p -fixed cardinality linear ordering problem (FCLOP). Then an integer programming formulation for the FCLOP will be proposed and will be improved by introducing several valid inequalities for the polytope characterized by the FCLOP linear relaxation

(introduced in chapter 2). We have defined the polytope corresponding to the FCLOP, characterized the dimension of the polytope for different p values and identified several facet defining inequalities for it.

We have also introduced several exact solution algorithms to solve different instances of the problem. Since the FCLOP is introduced for the first time in this PhD thesis, there exist no study and therefore no results on it. As a consequence, we have used CPLEX (ILOG, 2012) the widely used LP software, to validate our numerical experiments.

In this chapter, we review the state of the art on linear ordering problem, description of the problem and solution methods.

In chapter 2, we discuss applications and motivations of the study on p -fixed cardinality linear ordering problem. We also introduce an initial integer programming mathematical formulation for the FCLOP that will be improved later.

In chapter 3, the polytope corresponding to the FCLOP model introduced formerly, is defined. Later, several classes of valid inequalities with respect to the defined polytope are proposed to improve the mathematical formulation of the problem and reduce the integrality gap.

We also introduce an algorithm to find non-valid inequalities with respect to the known polytope, that remove feasible solutions having the same objective values.

Chapter 4 is dedicated to the polyhedral study, identifying first the dimension of the FCLOP polytope for different p values. We discuss then the facets of the FCLOP and identify some classes of facet defining inequalities.

Chapter 5 serves to introduce some solution algorithms to solve the FCLOP. In this chapter, we propose several relaxation strategies based on Lagrangian relaxation. We also propose a Lagrangian decomposition algorithms that decompose the problem into three subproblems. The advantage of such a decomposition is to have relatively easy subproblems comparing to the initial FCLOP and the Lagrangian relaxation algorithms as well. Numerical experiments concerning the proposed solution algorithms are presented at the end of the chapter 5.

In chapter 6, a relaxation algorithm enriched by a cut generator algorithm is proposed to solve the problem. In this chapter, we try to apply particular properties of the FCLOP to introduce an efficient exact solution algorithm. We will show that whenever p is smaller than n , from among the classical triangle-free inequalities there are a few of them active in a feasible solution. We propose a solution algorithm based on relaxing the set of classical triangle-free inequalities and apply a feasibility cut generation algorithm to

generate violated cuts and add them to the model, if any. We will show that the algorithm will find the optimal solution in a finite number of iterations. We also introduce an extended relaxation base algorithm using the same idea. Numerical experiments reported at the end of the chapter 6 confirms the efficiency of these algorithms for the cases when $p \ll n$.

In chapter 7, a relax-and-cut algorithm is proposed inducing part of the valid inequalities proposed earlier in the model. A comparison between a direct Lagrangian relaxation algorithm and the proposed relax-and-cut algorithm is presented and is followed by the numerical results.

Chapter 8 is dedicated to summarize the work, conclude the results and propose some ideas on future researches.

1.1 Literature review - State of the art of the LOP

The literature review is organized as follows. The state-of-the-arts in Linear Ordering Problem (LOP), polyhedral studies and solution methods are reviewed in different subsections. As Fixed Cardinality Linear Ordering Problem (FCLOP) will be introduced in this thesis, therefore, no previous literature is available to report. The present review does not pretend to be exhaustive with respect to theory and application of the LOP, but we focus on the results and methods which may be of interest to extend to the p-fixed cardinality case (see Martí and Reinelt (2011) for a complete textbook on the LOP).

1.1.1 Linear ordering problem and applications

Martí and Reinelt (2011) provides a comprehensive and categorized introduction to different variants, solution methods and polyhedral studies. The interested readers are referred to Martí and Reinelt (2011) for further details.

In economics, input-output models of a multisectorial economy have long been studied by economists to model the exchange flows (of cash, products or activities) between the different sectors of a market or a country. The analysis entails dividing up the economy into n sectors that produce goods and/or services, and determines the quantities that each sector must produce to meet both internal and external demands. These quantities are the coefficients of the so-called input-output matrix and can be supposed positive wlog.

Triangulation of an input-output table is the process of determining a hierarchical ordering of the different industries of production such that the flow of monetary value between them is maximized. The optimal value is proportional to the so-called *linearity factor* of the economy. Since the sum

of all the entries of the matrix A less those elements on the main diagonal is always constant, the triangulation problem is reduced to finding a permutation of the rows and columns such that the sum of the elements above the main diagonal is maximum. Clearly, this is equivalent to the linear ordering problem.

In practice, economists have observed that the linearity measure is not sensitive to the influence of many economic sectors, giving rise to the question of determining the subset of the most influent sectors and their respective order. Moreover, the input-output tables are rather sparse (rows or columns of zeroes are common) inducing many alternative optimal orders indeed difficult to detect (see Grötschel et al. (1983)). If the cardinality parameter p defining the (FCLOP) model is not a priori fixed in that situation, the analysis could have to compare different versions of the (FCLOP) applied to triangulation problem for input-output matrices with different values of p . The study of inter-compartmental models in ecology is another example where the identification of the most influent sectors is critical (Lenzen, 2007).

Another application of the linear ordering problem is in the voting theory. A classical question in voting theory is the search for a common order of candidates from a set of individual orders (binary or complete) given by the votant. Partial orders on a subset of p candidates will as well be of interest, but which subset of p candidates. The fixed-cardinality LOP can thus be viewed as a subproblem of a general ordering problem where one compares different groups of a limited number of candidates.

An other important area of application for many network optimization problems is transportation and telecommunications network design. In some transportation and telecommunications applications, we have situations such that not only the distance/cost and the volume of flow exchange influence the choice of facility locations, but also the mutual attractiveness of locations between each other can play a very important role. Therefore the cost function would be a linear combination of 1) a minimization problem which minimizes the setup and flow transfer cost, and 2) a maximization problem which influences the choice of p facilities in such a way that the total attractiveness associated with the choice of pairs of locations is being maximized.

In Telecommunications and in particular in Internet Service Providers industry, there are several agreements for different situations between every pair of ISPs. When an ISP wants to ensure Internet connectivity to its users, it needs to come to some agreements with others. To decide how many other ISPs should sign such an agreement (i.e. p) is a managerial decision which takes into account some political aspects (e.g. avoiding monopoly, some political compromise etc.). In the lower level decision making, the bandwidth of $i - j$ connection, the technology, compatibility, historical data and past experiences etc. are used to define p_{ij} as the attractiveness of collaboration

between ISP_i and ISP_j .

In maritime transport, particularly in liner shipping, the large vessels call at some major ports based on a published itineraries. The set of port call sequence is known as '*string*'. The containers on the vessels are discharged and then distributed to smaller ports around the called major port. Similarly, the concentrated container of smaller ports are loaded into the ship. The major ports are selected based on draft, average turn around times etc. It is well-known in maritime industry that the time a vessels spends at a port is the unprofitable part of service. Vessel owners try to 1) call minimum possible port along the string to avoid turnaround time, 2) minimize the total flow transport costs, 3) determine the sequence of calls.

In most of these examples, the cardinality and ordering constraints appear as embedded in a larger design model, quite amenable to decomposition techniques which will induce (FCLOP) subproblems.

In Sirdey and Kerivin (2007), Sirdey and Kerivin have addressed the Process Move Programming problem where a set of processes have to be assigned to a set of processors, a process move consisting of a suborder of states which must satisfy capacity constraints due to processors types. A *partial ordering* problems results from the modelling which can be interpreted as a (LOP) with no cardinality constraint at all, thus including all (FCLOP) formulations.

Charon and Hudry (2007) provides a survey on the LOP for weighted/unweighted tournaments. This problem consists in finding a linear order which is at minimum distance from a (weighted or not) tournament.

In general, a tournament T in a graph $G = (V_n, A_n)$ consists of arcs containing for every pair of nodes i and j either arc (i, j) or arc (j, i) , but not both.

Grötschel et al. (1984), Grötschel et al. (1985a) and Grötschel et al. (1985b) explain the fact that a closely related problem to the LOP is the maximum acyclic subgraph problem. In fact the LOP represents a special case of the acyclic subgraph problem (Grötschel et al., 1985a) when the input matrix represents a complete weighted oriented graph. They also explain that the optimal solution of the linear ordering problem is optimal also for the acyclic subgraph problem. Moreover, they precise that the linear ordering polytope is a face of the maximum acyclic subgraph polytope (Grötschel et al., 1985a). Another version of the maximum acyclic subgraph problem is the minimum feedback arc set problem. A feedback arc set in a digraph $D = (V_n, A_n)$ with arc weights c_{ij} for all $i, j \in A_n$, is a set $F \subseteq A_n$ such that $D \setminus F$ is acyclic. Therefore, the minimum feedback arc set problem looks for F such that the sum over all the cost of the arcs present in F is minimized Charbit et al. (2007), Eades et al. (1993), Grötschel et al. (1984).

Bertacco et al. (2008) introduced the so called *Linear Ordering Problem with Cumulative Costs* with practical application in wireless telecommunication systems. They studied a variant of the LOP for which the overall permutation cost can be expressed as the sum of terms α_u associated with each item u , each defined as a linear combination of the values α_v of all items v that follow u in the permutation. They prove that the problem is indeed NP-hard and propose an ad hoc enumerative algorithm as well as a dynamic-programming heuristic.

Buchheim et al. (2010) studied the quadratic linear ordering problem and proposed a positive semi-definite relaxation to solve it. The quadratic linear ordering problem represents a generalization of various optimization problems, such as bipartite crossing minimization or the betweenness problem, which includes linear arrangement problem.

Hungerländer and Rendl (2011) present a generic study on ordering problems, either linear or quadratic ordering problem. In this study, a new quadratic constraint formulation for the (linear or quadratic) ordering problem is introduced and the positive semi-definite relaxation technique applied solve the problem. It is reported in this study that the proposed positive semi-definite relaxation algorithm is also efficient on a variety of optimization problems i.e. bipartite crossing minimization problem, minimum linear arrangement problem, the single-row facility layout problem, Multi-level crossing minimization problem and weighted betweenness problem.

1.1.2 Polyhedral studies for the LOP

Grötschel et al. (1985a) proved that among all the classes of facet defining inequalities of acyclic subgraph polytope that were proposed in Grötschel et al. (1985b), only some classes of valid inequalities such as triangle-free inequalities, simple k-fence and Möbius ladder inequalities are facet-defining. As an example, k-dicycle inequalities which are facet defining for the maximum acyclic subgraph polytope are the groups which are not defining facet for the linear ordering polytope. They also used those inequalities in a cutting plane method for the triangulation of the input-output matrices (Grötschel et al., 1984).

Reinelt (1993) also studied the linear ordering polytope and proposed a list of facet defining inequalities for the linear ordering polytope when the number of vertices is smaller than or equal to 7.

Reinelt (1993) and Grötschel et al. (1985a) both showed that the linear ordering polytope of the problem with n nodes inherits all facets from the polytope of linear ordering polytope of $n - 1$ nodes.

Leung and Lee (1994) also proposed new facets by generalizing facets induced by subgraphs called *fences* which were introduced firstly by Grötschel et al. (1985a) and augmented fences which were introduced by McLennan McLennan (1990).

Bolotashvili et al. (1999) presented a procedure called the rotation method to generate new facet defining inequalities by using the information of the facets which are already known. Applying the rotation method to the linear ordering polytope led them to generalize several facets i.e. facets induced by subgraphs called m-fences, *Möbius* ladders and Z_m facets Reinelt (1985), t-reinforced m-fences (Leung and Lee, 1994) and (m, k) fences Bolotashvili (1986).

1.1.3 Solution methods

Solution methods for the linear ordering problem have been early proposed and tested on many instances so that the problem is considered well solved by exact methods (up to 100 nodes).

Hoogeveen and Velde (1995) deals with a scheduling problem where the Lagrangian relaxation is used to solve the problem. The resulting Lagrangian relaxation sub-problem becomes a LOP which is polynomially solvable under certain conditions.

Grötschel et al. (1984) proposed a new cutting plane algorithm based on the results regarding the facet-defining inequalities of the LOP which they have proposed in their earlier study Grötschel et al. (1985a).

Mitchell and Borchers (2000) proposed a cutting plane algorithm which uses a primal-dual interior point method as well as the simplex method to solve the linear ordering problem. They do a comparison between their own algorithm with one that uses only the simplex method and with one that uses only an interior point method.

Belloni and Lucena (2004) proposed two heuristics for the linear ordering problem embedded within a Lagrangian relaxation framework. They exploited dual information obtained from Lagrangian relaxation to construct feasible solution and then improved the solution using their heuristic. As the number of relaxed constraint becomes prohibitive, they also applied a relax-and-cut approach. Out of 79 instances, they proved optimality in 72 cases.

Belloni (2001) proposed two heuristics. The first one is based on the so called position cost (modified costs) to sequentially build a linear ordering of the nodes. The second one directly used the LR solution.

Schiavinotto and Stützle (2004) presented a comprehensive search space analysis of the LOLIB library instance classes. They conclude that adaptive restart algorithms like iterated local search or memetic algorithms that iteratively generate new starting solutions for a local search based on previous search experience, are promising candidates for obtaining high performing algorithms.

Campos et al. (1999) proposed a scatter search, that could find high quality solutions for the linear ordering problem. The scatter search algorithm proposed in this study, combines solutions in a linear order to create new solutions and maintain a balance between quality and diversification in the reference set.

Laguna et al. (1999) proposed a tabu search to solve the triangulation problem for input-output matrices in economy.

As well, Garcia et al. (2006) introduced a variable neighborhood search and a hybrid method to apply a short term tabu search combined with a principal algorithm based on variable neighborhood search.

1.2 Summary and conclusion

A literature review and the state of the art of the linear ordering problem is presented in this chapter.

As mentioned earlier, the fixed cardinality linear ordering problem is a new combinatorial optimization problem that can be classified as an extension of the linear ordering problem and will be studied for the first time in this PhD thesis. Therefore, there exist no previous study available on this subject. However, there are several studies focused on combinatorial optimization problems with fixed cardinality constraints. Bruglieri et al. (2006) provide an annotated bibliography of the combinatorial optimization problems that have been considered with the cardinality constraints.

CHAPTER 2

Modeling the Fixed Cardinality Linear Ordering Problem

An introduction on the fixed cardinality linear ordering problem is given in this chapter. It is then followed by introducing some applications and motivations of this study.

Given a ground set $V_n = \{1, 2, \dots, n\}$. We define $G = (V_n, A_n)$ to be the complete directed graph induced by the node set V_n . Here, $A_n = \{(i, j) \in V_n \times V_n, i \neq j\}$. For any function $F : V_n \cup A_n \rightarrow \mathbb{R}$ and $T \subseteq V_n \cup A_n$, we define

$$F(T) = \sum_{e \in T} F(e).$$

A linear order π_S on $S \subseteq V_n$, is a bijection from $\{1, 2, \dots, |S|\}$ to $S \subseteq V_n$. When each ordered pair $(i, j) \in V_n \times V_n$ is associated with a weight w_{ij} , the weight of a given linear order π_S is

$$w(\pi_S) = \sum_{i \in S} \sum_{j \in S, j \neq i} f_{ij} w_{ij}, \quad (2.1)$$

where $f_{ij} = 1$ if $\pi_S^{-1}(i) < \pi_S^{-1}(j)$, and $f_{ij} = 0$ otherwise.

When it is necessary and useful, we also denote $\pi_S = (a_1, a_2, \dots, a_p)$ the linear order on $S \subseteq V_n$ with $|S| = p$. In this notation, $\pi_S^{-1}(a_i) = i$ and

$$w(\pi_S) = \sum_{i=1}^p \sum_{i < j \leq p} w_{a_i a_j}.$$

Given a subset $S \subseteq V_n$, let Π_S denotes the set of permutations of the elements of S .

Given a scalar p , the Fixed Cardinality Linear Ordering Problem (FCLOP) seeks a linear ordering $\pi_{S^*}^*$ where,

$$w(\pi_{S^*}^*) = \max_{S \subseteq V, |S|=p} \{w(\pi_S^*)\}, \quad (2.2)$$

and

$$w(\pi_S^*) = \max_{\pi_S \in \Pi_S} \{w(\pi_S)\}. \quad (2.3)$$

2.1 Definitions and notations

Given a linear order π_S , let $G_{\pi_S} = (S, A(\pi_S))$ be the graph having S as the node set and $A(\pi_S)$ the set of arcs such that for all $u, v \in S$, we have $(u, v) \in A(\pi_S)$ if and only if $\pi_S^{-1}(u) < \pi_S^{-1}(v)$.

Let $G[S]$ be the complete directed graph induced by S .

Among all the possible $S \subseteq V$, such that $|S|=p$, the FCLOP may be seen as the problem of determining the *set* S and the *maximum weighted acyclic tournament* (a LOP) in $G[S]$.

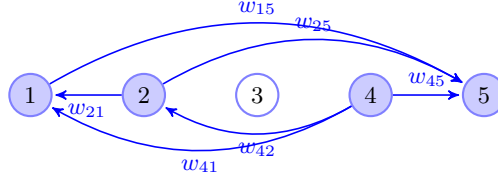


Figure 2.1: An example of G_{π_S} with $S = \{1, 2, 4, 5\}$ and $\pi_S = (4, 2, 1, 5)$.

Given a linear order π_S , its *incidence vector* (x^{π_S}, y^{π_S}) is defined as follows: $x_{ij}^{\pi_S} = 1$ if $\pi_S^{-1}(i) < \pi_S^{-1}(j)$ and $\{i, j\} \subseteq S$; $x_{ij}^{\pi_S} = 0$ if $\pi_S^{-1}(i) > \pi_S^{-1}(j)$ or $\{i, j\} \not\subseteq S$; $y_i^{\pi_S} = 1$ if $i \in S$ and $y_i^{\pi_S} = 0$ if $i \notin S$. The Fixed cardinality linear ordering polytope is the convex hull of the incidence vectors of the linear order π_S , $\forall S \subseteq V_n$, $|S|=p$.

That is

$$FCLOP(G_n, p) = \text{conv}\{(x^{\pi_S}, y^{\pi_S}) : S \subseteq V_n, |S|=p, \pi_S \in \Pi_S\}.$$

Given a linear order π_S for an arbitrary $S \subseteq V_n$, we denote by $\delta_{\pi_S}^+(i)$, $\forall i \in S$ the set of arcs (i, j) with $\pi_S^{-1}(i) < \pi_S^{-1}(j)$.

A *Cut* is a set of edges when removed, or increases the number of connected components of the underlying graph).

Let $G = (V, A)$ be any directed graph. For $i \in V$, the set of arcs of G with tail (head) i is represented by $\delta_G^+(i)$ ($\delta_G^-(i)$), and let $\delta_G(i) = \delta_G^+(i) \cup \delta_G^-(i)$.

As well, for a given subset $W \subset V$, $\delta_G^+(W)$ denotes the set of arcs $(u, v) \in A$, with $u \in W$ and $v \in V \setminus W$. Also we denote by $\delta_G^-(W)$ the set of arcs (u, v)

with $v \in W$ and $u \in V \setminus W$. When there is no confusion about the graph, we omit the subscript G .

If each arc a of G is associated with a weight $w(a)$, then the weight of the cut $\delta^+(W)$ is

$$w(\delta^+(W)) = \sum_{a \in \delta^+(W)} w(a).$$

2.2 Motivation

Input-output models of a multisectorial economy have long been studied by economists to model the exchange flows (of cash, products or activities) between the different sectors of a market or a country. The analysis entails dividing up the economy into n sectors that produce goods and/or services, and determines the quantities that each sector must produce to meet both internal and external demands. These quantities are the coefficients of the so-called input-output matrix and can be supposed positive wlog.

Triangulation of an input-output table is the process of determining a hierarchical ordering of the different industries of production such that the flow of monetary value between them is maximized. The optimal value is proportional to the so-called *linearity factor* of the economy. Since the sum of all the entries of the matrix A less those elements on the main diagonal is always constant, the triangulation problem is reduced to finding a permutation of the rows and columns such that the sum of the elements above the main diagonal is maximum. Clearly, this is equivalent to the linear ordering problem.

In practice, economists have observed that the linearity measure is not sensitive to the influence of many economic sectors, giving rise to the question of determining the subset of the most influent sectors and their respective order. Moreover, the input-output tables are rather sparse (rows or columns of zeroes are common) inducing many alternative optimal orders indeed difficult to detect (see Groetschel et al Grötschel et al. (1983)). If the cardinality parameter p defining the FCLOP model is not a priori fixed in that situation, the analysis could have to compare different versions of the FCLOP applied to input-output matrices triangulation with different values of p . The study of inter-comportamental models in ecology is another example where the identification of the most influent sectors is critical (Lenzen Lenzen (2007)).

2.2.1 Voting theory and analysis of ordinal data

A classical question in voting theory is the search for a common order of candidates from a set of individual orders (binary or complete) given by the votant. Partial orders on a subset of p candidates will as well be of interest, but which subset of p candidates. The FCLOP can thus be viewed

as a subproblem of a general ordering problem where one compares different groups of a limited number of candidates.

2.2.2 Transportation and communications network design

In some transportation and telecommunications applications, not only the distance/cost and the volume of flow exchange influence the choice of facility locations, but also the mutual attractiveness of locations between each other can play a very important role. Therefore the cost function would be a linear combination of 1) a minimization problem which minimizes the setup and flow transfer cost, and 2) a maximization problem which influences the choice of p facilities in such a way that the total attractiveness associated with the choice of pairs of locations is being maximized.

In Telecommunications and in particular in Internet Service Providers industry, there are several agreements for different situations between every pair of ISPs. When an ISP wants to ensure Internet connectivity to its users, it needs to come to some agreements with others. To decide how many other ISPs should sign such an agreement (i.e. p) is a managerial decision which takes into account some political aspects (e.g. avoiding monopoly, some political compromise etc.). In the lower level decision making, the bandwidth of $i - j$ connection, the technology, compatibility, historical data and past experiences etc. are used to define p_{ij} as the attractiveness of collaboration between ISP_i and ISP_j .

In maritime transport, particularly in liner shipping, the large vessels call at some major ports based on a published itineraries. The set of port call sequence is known as '*string*'. The containers on the vessels are discharged and then distributed to smaller ports around the called major port. Similarly, the concentrated container of smaller ports are loaded into the ship. The major ports are selected based on draft, average turn around times etc. It is well-known in maritime industry that the time a vessels spends at a port is the unprofitable part of service. Vessel owners try to 1) call minimum possible port along the string to avoid turnaround time, 2) minimize the total flow transport costs, 3) determine the sequence of calls.

In most of these examples, the cardinality and ordering constraints appear as embedded in a larger design model, quite amenable to decomposition techniques which will induce FCLOP subproblems.

2.2.3 Resource-constrained scheduling problems

In Sirdey and Kerivin (2007), Sirdey and Kerivin have addressed the Process Move Programming problem where a set of processes have to be assigned to a set of processors, a process move consisting of a suborder of states which must satisfy capacity constraints due to processors types. A *partial ordering*

problems results from the modelling which can be interpreted as a LOP with no cardinality constraint at all, thus including all FCLOP formulations.

2.3 Mathematical modeling

In this section, we firstly recall the known mathematical formulation of the linear ordering problem and continue by introducing an integer linear programming formulation for the FCLOP. At the end of this chapter, we also propose a quadratic programming formulation for the FCLOP.

2.3.1 Linear ordering problem

For a given linear order π_S (when $|S|=n$), let the decision variables x_{ij} , $\forall i, j \in V_n$, $i \neq j$ be defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if } \pi_S^{-1}(i) < \pi_S^{-1}(j) \\ 0 & \text{otherwise} \end{cases}$$

An integer programming formulation of the linear ordering problem can be proposed as follows:

$$\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \quad (2.4)$$

s.t.

$$x_{ij} + x_{ji} = 1 \quad \forall i, j \in V_n, i \neq j \quad (2.5)$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall i, j, k \in V_n, i \neq j \neq k \quad (2.6)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V_n, i \neq j \quad (2.7)$$

In this formulation, constraints (2.5) ensure that i is ordered before j or (exclusive) j is ordered before i in any feasible solution. Inequalities (2.6) when combined with (2.5) avoid the existence of any directed cycle in a feasible solution.

2.3.2 Fixed cardinality linear ordering problem

Integer programming formulation

For a given linear order π_S , $|S|=p$, let the decision variables x_{ij} and y_i for all $i, j \in V_n$, $i \neq j$ be defined as follow:

$$x_{ij} = \begin{cases} 1 & \text{if } \pi_S^{-1}(i) < \pi_S^{-1}(j) \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad y_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

Then an integer programming formulation for the FCLOP can be proposed as follow:

$$\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \quad (2.8)$$

s.t.

$$x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V_n, i < j \quad (2.9)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V_n, i < j \quad (2.10)$$

$$x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall i, j \in V_n, i < j \quad (2.11)$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall i, j, k \in V_n : i \neq j \neq k \quad (2.12)$$

$$\sum_{i \in V_n} y_i = p \quad (2.13)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n, i \neq j \quad (2.14)$$

Inequalities (2.9) and (2.10) ensure that there exist at most one arc between every pair of nodes i and j , only if i and j both are selected in a linear order. Inequalities (2.11) ensure the existence of at least one arc between i and j , when both are in a linear order. In fact (2.11) together with (2.9) and (2.10) guarantee the existence of exactly one arc between every pair of vertices in S .

Inequalities (2.12) ensure that there exist no triangle in a graph corresponding to a feasible solution. Inequalities (2.12) are known as the classical triangle-free inequality (Grötschel et al., 1985b) and are sufficient to eliminate all the potential directed cycles including more than three arcs in every feasible solution of the FCLOP.

In the next lemma, we will show that inequalities (2.12) together with (2.9) - (2.11) avoid directed cycles of size greater than three.

Remark 1 *Inequalities (2.12) together with (2.9) - (2.11) are sufficient to avoid any directed cycle of size greater than three.*

2.4 Summary and conclusion

In the second chapter, a brief definition of the fixed cardinality linear ordering problem and several applications and motivations for the study has been proposed. Finally an integer programming formulation of the FCLOP has been proposed that will be improved later in the following chapters.

CHAPTER 3

Integer Programming Formulation and Valid Inequalities for the FCLOP Polytope

Let G_n, A_n be the complete directed graph on n vertices. Each arc $(i, j) \in A_n$ is associated with a variable x_{ij} and each node $i \in V_n$ is associated with a variable y_i .

The following is a linear relaxation for the FCLOP:

$$\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \quad (3.1)$$

s.t.

$$x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V_n, i < j \quad (3.2)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V_n, i < j \quad (3.3)$$

$$x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall i, j \in V_n, i < j \quad (3.4)$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall i, j, k \in V_n : i \neq j \neq k \quad (3.5)$$

$$\sum_{i \in V_n} y_i = p \quad (3.6)$$

$$0 \leq x_{ij} \quad \forall i, j \in V_n, i \neq j \quad (3.7)$$

$$y_i \leq 1 \quad \forall i \in V_n \quad (3.8)$$

Recall that $FCLOP(G_n, p)$ is the polytope associated with the FCLOP problem with n nodes and when the cardinality of the linear order is p . It may be seen as the convex hull of the 0 – 1 solutions satisfying (3.2) - (3.8).

Our polytope $FCLOP(G_n, p)$ is closely related to the polytope studied by Sirdey and Kerivin (2007). In their study the constant p is not fixed and

they look for the convex hull of all the linear orders of any cardinality. That is the polytope they studied called the *Partial Linear Ordering Polytope* P_{PLO}^n is the convex hull of all the 0-1 vectors satisfying inequalities (3.2)-(3.5). Hence any valid inequality for P_{PLO}^n is also valid for $FCLOP(G_n, p)$. In this chapter, we introduce several classes of strengthening valid inequalities for the polytope defined by (3.2) - (3.8). We also introduce non valid inequalities that break symmetry. This symmetry breaking inequalities are not valid for the FCLOP polytope since they cut integer solutions to reduce the number of integer feasible solutions having the same objective value.. In the process of separating the symmetry breaking inequalities, the algorithm ensures the existence of at least one integer feasible solutions from any set of integer feasible solutions that have similar objective value.

We report the impact of the new valid inequalities in separate tables on a selected set of instances of LOLIB¹. These instances are chosen in such a way that instances from different classes of problems in the LOLIB are represented in our testbed. Note that the level of intractability changes significantly from one class of instance to another in LOLIB.

3.1 Category 1: The properties of underlying graph

3.1.1 Node-Degree constraints

These valid inequalities for $FCLOP(G_n, p)$ are obtained from the following facts:

- a) $p = n - 1$. In each pair of nodes, at least one of the nodes belongs to a feasible solution (see Figure 3.1).

$$x_{ij} + x_{ji} = y_i + y_j - 1 \quad \forall i, j \in V_n, i \neq j. \quad (3.9)$$

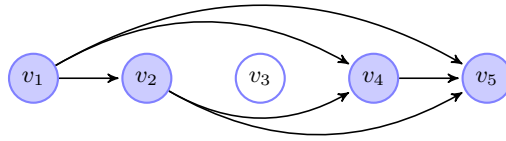
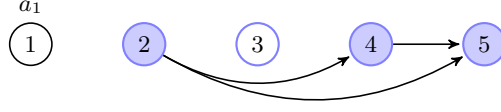


Figure 3.1: An example of G_{π_S} with $n = 5, p = 4$.

- b) $p < n - 1$. In the graph associated with the linear order π_S , G_{π_S} , the number of outgoing and incoming arcs to a vertex i is equal to $p - 1$ if $i \in S$. If $i \notin S$, this number is zero (see Figure 3.2).

$$x(\delta(i)) = (p - 1)y_i \quad \forall i \in V_n. \quad (3.10)$$

¹<http://www.iwr.uni-heidelberg.de/groups/comopt/software/LOLIB/>


 Figure 3.2: An example of G_{π_S} for $p = 3, n = 5$.

Lemma 1 (Node-Degree) *Constraints (3.9) and (3.10) are valid for the $FCLOP(G_n, p)$ polytope.*

Proof *In order to prove that (3.9) and (3.10) are valid for the $FCLOP(G_n, p)$ polytope, we need to consider two cases:*

Case 1) $p = n - 1$: In this case, in any feasible solution of the $FCLOP$, there exists at least one selected vertex between every pair of vertices i and j . Therefore $0 \leq y_i + y_j - 1 \leq 1$. The upper bound is met when both i and j are designated to be among the p vertices on the linear order. As well, the lower bound is met when between i and j , one of them is not selected to be present in the current feasible solution and actually, there exist no other situation since $p = n - 1$.

Case 2) $p < n - 1$: For an arbitrary vertex i , the total number of arcs arriving to i ($|\delta^-(i)|$) plus the total number of arcs leaving i ($|\delta^+(i)|$) are said to be the degree of i (say $|\delta(i)|$).

It is well-known that any feasible solution of the LOP (as well as the $FCLOP$), represents a complete cycle-free digraph (sub-digraph for the $FCLOP$). Hence, the degree of every vertex appearing in a feasible linear order is equal to $p - 1$. In the contrary, if an arbitrary vertex j is not designated among the selected p vertices, then $y_j = 0$ and we have $x(\delta(j)) = 0$. \square

Note 1 *From now on, we distinguish between the two $FCLOP$ cases, $p = n - 1$ and $p < n - 1$.*

3.1.2 TotalX constraint

In a graph associated to a feasible solution of the $FCLOP$ (e.g. Figure 3.1 and Figure 3.2), we always have

$$\sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} = \frac{p(p-1)}{2} \quad (3.11)$$

Lemma 2 *(3.11) is valid with respect to the $FCLOP(G_n, p)$ polytope for all n and $0 \leq p \leq n$.*

Proof The case $p = n$ represents an LOP and is obvious. Given $p = n - 1$. Then (3.11) is equal to the aggregation of all the (3.9) equalities over the i index.

Now consider $0 \leq p < n - 1$. In this case, constraint (3.11) is equal to the aggregation of all the (3.10) equalities over the i and j indices. \square

The TotalX constraint is redundant in presence of (3.9) when $p = n - 1$ for $FCLOP(G_n, n - 1)$. It is also redundant in presence of (3.10) for $FCLOP(G_n, p)$ when $0 \leq p < n - 1$. However, our extensive computational experiments revealed that, in fact, it often contributes significantly in reducing the computation time as reported in Table 3.5.

3.1.3 Triangle-free inequalities

In the sequel, we are proposing three classes of inequalities to strengthen Triangle-Free (TF) inequalities (3.5).

TF - Class 1

The first class of triangle-free inequalities is a natural generalization of the classical triangle-free inequalities (3.5):

$$x_{ij} + x_{jk} + x_{ki} \leq y_r + y_s \quad \forall \{r, s\} \subset \{i, j, k\} \subseteq V_n, i \neq j \neq k \quad (3.12)$$

The number of these inequalities is equal to $n(n - 1)(n - 2)$. Triangle-free inequalities ensure that there is no directed cycle composed of three arcs in any feasible solution to the FCLOP (see Figure 3.3).

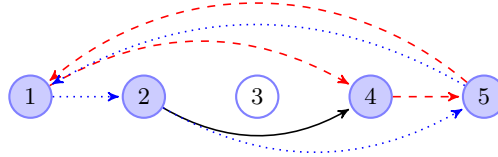


Figure 3.3: Example of an infeasible solution with two directed triangles (dashed and dotted) which are removed by adding constraints (3.12).

Lemma 3 Inequalities (3.12) are valid for $FCLOP(G_n, p)$, for all $n \geq 3$ and $1 \leq p \leq n$.

Proof It is clear that we might have $x_{ij} + x_{jk} + x_{ki} = 2$ only if $i, j, k \in S$, that means $y_i = y_j = y_k = 1$. Since $y_i \leq 1, \forall i \in V_n$; consequently we always have $x_{ij} + x_{jk} + x_{ki} \leq y_r + y_s, \forall \{r, s\} \subseteq \{i, j, k\} \subseteq V_n$. \square

Hence by adding these inequalities to our linear relaxation (3.2) - (3.8), the inequalities (3.5) become redundant.

TF - Class 2

The second class of triangle-free inequalities to eliminate every triangle in a feasible solution of the FCLOP is the following:

For an arbitrary set of vertices $\{i, j, k\}$ in V_n , if we have i before k and k before j , then:

- (i) k must be among the p designated vertices and,
- (ii) there must exist an arc from i to j .

$$x_{ik} + x_{kj} - x_{ij} - y_k \leq 0 \quad \forall i, j, k \in V_n, i \neq j \neq k \quad (3.13)$$

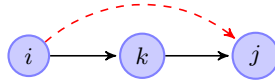


Figure 3.4: Constraints (3.13) ensures that if (i, k) and (k, j) co-exist, then the arc (i, j) is enforced.

Lemma 4 *Inequalities (3.12) are implied by (3.2) and (3.3).*

Proof *Without loss of generality, let us show that inequalities*

$$x_{12} + x_{23} + x_{31} \leq y_1 + y_2 \quad (3.14)$$

$$x_{12} + x_{23} + x_{31} \leq y_2 + y_3 \quad (3.15)$$

$$x_{12} + x_{23} + x_{31} \leq y_1 + y_3 \quad (3.16)$$

are implied by (3.13), (3.2) and (3.3).

From (3.13) we have

$$x_{12} + x_{23} - x_{13} - y_2 \leq 0 \quad (3.17)$$

$$x_{31} + x_{12} - x_{32} - y_1 \leq 0 \quad (3.18)$$

and from (3.2) and (3.3) we have

$$x_{13} + x_{31} - y_1 \leq 0 \quad (3.19)$$

$$x_{13} + x_{31} - y_3 \leq 0 \quad (3.20)$$

$$x_{23} + x_{32} - y_3 \leq 0 \quad (3.21)$$

Now, (3.14), (3.15) and (3.16) are obtained by combining (3.17) with (3.19), (3.17) with (3.20) and (3.18) with (3.21) respectively. \square

TF - Class 3

The third set of proposed triangle-free inequalities are the following:

$$x_{ij} + x_{jk} + x_{ki} + \sum_{i \in V_n \setminus \{i,j,k\}} y \leq \lfloor \frac{2p + (n-3)}{3} \rfloor \quad \forall \{i, j, k\} \subseteq V_n \quad (3.22)$$

Lemma 5 *Inequalities (3.22) are valid inequalities for FCLOP(G_n, p).*

Proof From (3.12) we have:

$$\begin{aligned} x_{ij} + x_{jk} + x_{ki} - y_i - y_j &\leq 0, \\ x_{ij} + x_{jk} + x_{ki} - y_i - y_k &\leq 0, \\ x_{ij} + x_{jk} + x_{ki} - y_j - y_k &\leq 0. \end{aligned}$$

Their sum give the following inequality

$$3(x_{ij} + x_{jk} + x_{ki}) - 2(y_i + y_j + y_k) \leq 0. \quad (3.23)$$

We know that $y_i + y_j + y_k = p - y(V_n \setminus \{i, j, k\})$. By replacing this value of $y_i + y_j + y_k$ in (3.23) we obtain

$$3(x_{ij} + x_{jk} + x_{ki}) + 2y(V_n \setminus \{i, j, k\}) \leq 2p. \quad (3.24)$$

Combining this later inequalities with the trivial inequalities $y_i \leq 1$ for $i \in V_n \setminus \{i, j, k\}$ we get

$$3(x_{ij} + x_{jk} + x_{ki}) + 3y(V_n \setminus \{i, j, k\}) \leq 2p + (n-3). \quad (3.25)$$

Now by dividing both sides of (3.26) by three and rounding down the right hand side we obtain the result.

$$x_{ij} + x_{jk} + x_{ki} + y(V_n \setminus \{i, j, k\}) \leq \lfloor \frac{2p + (n-3)}{3} \rfloor. \quad (3.26)$$

□

In the sequel, some more classes of valid inequalities are presented:

Various valid inequalities For an arbitrary set of vertices $\{i, j, k\}$ of distinct elements, if there exist arcs from i to both j and k and as well, from both j and k to i , then there must exist either (j, k) or (k, j) between k and j . (e.g. Figure 3.5 and Figure 3.6)

$$x_{ij} + x_{ik} - (x_{jk} + x_{kj}) - y_i \leq 0 \quad \forall i, j, k \in V_n : i \neq j \neq k \quad (3.27)$$

$$x_{ki} + x_{ji} - (x_{jk} + x_{kj}) - y_i \leq 0 \quad \forall i, j, k \in V_n : i \neq j \neq k \quad (3.28)$$

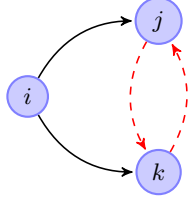


Figure 3.5: Constraints (3.27) ensure that exactly one of the two dashed arcs must be chosen.

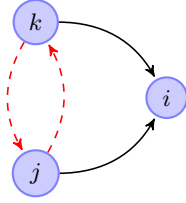


Figure 3.6: Constraints (3.28) ensure that exactly one of the two dashed arcs must be chosen.

Inequalities (3.29) are also valid with respect to $FCLOP(G_n, p)$.

$$x_{ij} + (x_{ik} + x_{ki}) - (x_{kj} + x_{jk}) - y_i \leq 0 \quad \forall i, j, k \in V_n : i \neq j \neq k \quad (3.29)$$

3.2 Category 2: Dynamic symmetry-breaking inequalities

Often computational efficiency is in a direct relationship with the size of the branch-and-bound tree and a good formulation can contribute significantly by providing a tighter bound, which requires minimal branching for proving optimality. However, even with a tight bound, still the combinatorial structure of problem can play a major role. More precisely, the existence of symmetry in the combinatorial structure can lead to situations where branching on a fractional variable would make another variable to take a similar fractional value with more or less the same objective function and

the bound never improves. This would indeed hinder the convergence of the branch-and-bound process.

To break symmetry, the first idea coming to mind is to slightly perturb the input data in such a way that it does not affect the optimal solution. While such a technique can potentially eliminate part of the symmetry, however normally such a technique does not contribute a lot in improving the computational time or reducing the integrality gap (Margot, 2010). In fact, exploiting the information about the structure of symmetry is often ways superior alternative.

Our initial experiments on several instances of the FCLOP have revealed existence of several optimal solutions (different linear orders having the same optimal value) for a big part of the testbed.

Besides the fact that often symmetry is due to the input data, the modeling aspects such as definition of a feasible solution as a directed path may become also a source of severe symmetry in the model. The latter, can be eliminated or reduced by generating a kind of dynamic symmetry breaking inequalities. This type of inequalities cut some integer feasible solutions but they do not affect the optimal solution.

Here, we aim at eliminating such symmetric solutions by generating symmetry-breaking inequalities in a dynamic way as soon as an integer incumbent appears in the course of branch-and-bound process.

Such cuts are not actually valid cuts for the FCLOP polytope because when added directly to the model (as static cuts) they certainly cut some parts of the polytop containing some integer feasible solutions, which might contain optimal solution. But, by adding them dynamically, one ensures that at least one of such solutions which represents the same objective value remains in the feasible space. This technique often has shown to contribute in reducing branch-and-bound tree size and accelerating convergence (see Margot (2010)).

In the sequel, we propose an algorithm to dynamically generate certain type of symmetry breaking inequalities to eliminate some sorts of symmetry for the $FCLOP(G_n, p)$ polytope.

Given a set of nodes $V_n = \{1, 2, \dots, n\}$, and $\pi_s = (a_1, a_2, \dots, a_p)$ any linear order on S when $|S| = p$.

Given $1 \leq k \leq n - 1$. Let S_k and \bar{S}_k be subsets of S defined as below:

$$S_k = \bigcup_{t=1}^k \{a_t\}, \quad \bar{S}_k = S \setminus S_k$$

From $\pi_S = (a_1, a_2, \dots, a_p)$, define the two sub-linear orders π_{S_k} and $\pi_{\bar{S}_k}$ as follows:

$$\pi_{S_k} = (a_1, a_2, \dots, a_k), \quad \pi_{\bar{S}_k} = (a_{k+1}, a_{k+2}, \dots, a_p).$$

Now, we are ready to present the algorithm to generate the symmetry-breaking inequalities for any incumbent solution in the course of branch-and-bound process.

3.2.1 Algorithm for finding symmetry-breaking inequalities

For k from 1 to $p - 1$ do:

- If $w(\delta_{G[S]}^+(S_k)) = w(\delta_{G[\bar{S}]}^-(S_k))$, add the following inequality to the model:

$$x(\delta^-(S_k)) + \sum_{a_i, a_j \in S_k, i < j} x_{a_i a_j} + \sum_{a_i, a_j \in \bar{S}_k, i < j} x_{a_i a_j} \leq \frac{p(p-1)}{2} - 1 \quad (3.30)$$

Remark 2 As mentioned earlier, constraints (3.30) do not represent valid inequalities with respect to $FCLOP(G_n, p)$. Yet, they can cut a feasible solution only if there exists other feasible solution(s) having the same objective value (an isomorphic optimal solution). Therefore, the optimal value does not change.

Figures 3.7 - 3.9 represent an example of a LOP feasible solution with $n = 5$ and $k = 2$, while Figures 3.10 - 3.12 represent a FCLOP example when $n = 5$, $p = 3$ and $k = 2$.

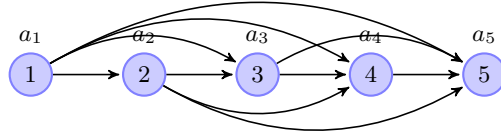


Figure 3.7: A feasible solution to the LOP, $n = 5$.

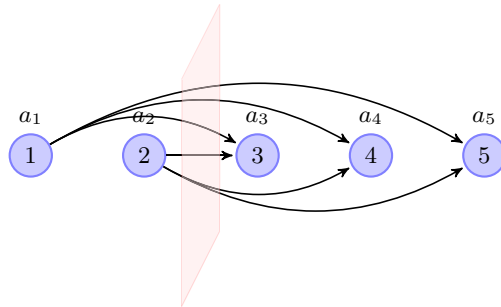


Figure 3.8: A cut example, $k = 2$

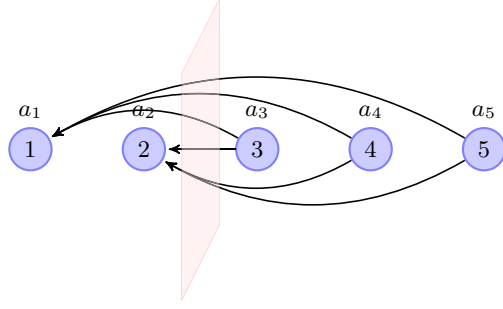


Figure 3.9: The set of arcs in $\delta_{G[S]}^-(S_k)$ when $k = 2$.

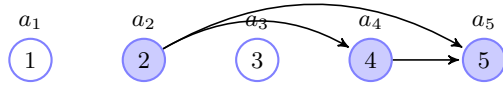


Figure 3.10: An example of G_{π_S} , when $n = 5, p = 3$.

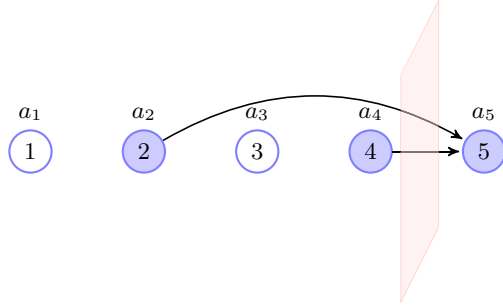


Figure 3.11: The set of arcs $\delta_{G[S]}^+(S_k)$ when $n = 5, p = 3$ and $k = 2$.

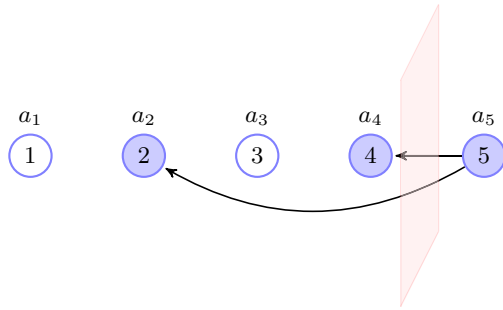


Figure 3.12: The set of arcs in $\delta_{G[S]}^-(S_k)$ when $n = 5, p = 3$ and $k = 2$.

3.3 Numerical results

A number of instances from the LOLIB² have been chosen to form a testbed for conducting our numerical experiments. This set of instances has been constructed in such a way that every class of problem is represented by at least a few instances. The reason for constructing such a restricted list of problem instances is that The LOLIB itself contains a large number of instances and in the FCLOP, we also have a parameter p , which varies in certain range for every instance size. A combination of all values of parameter p and all the instances of LOLIB will produces an unnecessarily huge testbed.

In the following subsections we report numerical results where CPLEX is used to solve instances of the selected testbed. Table 3.1 through Table 3.6 are organized as follows:

The instance names are brought in the first column in the format NAME- n - p where NAME represents the instance name from LOLIB, n stands for the number of nodes and p indicates the cardinality.

We have set a time limit of 3 hours (equivalent to 10800 sec.) for CPLEX. However, it might happen that CPLEX is in the middle of solving an LP at certain node when the time limit has been reached. In this case CPLEX terminates as soon as that process terminates. Therefore, a minor deviation from this time limit might rarely appear in our reports.

The second column reports the best objective value reported by CPLEX. The time elapsed to solve the instances is reported in the next column.

The following column reports the total number of processed nodes in the branch-and-bound tree.

The last column report the CPLEX termination status within the given time limit: 1) *FailFeas* indicates that the process exceeded the available memory, while there is at least one feasible solution that has been found during the solution process, 2) *TimLim* indicates that CPLEX terminated upon reaching the time limit, 3) *Optimal* and *OptimalTol* indicate that CPLEX terminates with the optimal solution within the time limit. Further explanation of CPLEX status can be found in the IBM ILOG Concert 12.5.1. It must be also noted that we have used a 32-bit version of ILOG Concert Technology.

Our testbed is composed of 51 instances from 7 different classes of input

²<http://www.opticom.es/lolib/>

matrices with different appropriate values of p .

3.3.1 FCLOP

In Table 3.1, the initial FCLOP model before adding any new valid inequalities has been considered. For N-atp24, with $n = 24$, the value p varies in $\{5, 10, 15, 20, 23, 24\}$. One observes that all the instances can be solved in reasonable times. The maximum elapsed CPU time is about 58 seconds. Yet, two instances are solved to optimality at the root node (bearing in mind the preprocessing, reduction and cuts added by CPLEX at the root node).

For the input matrix N-atp48, with $n = 48$, when p varies in

$$\{10, 15, 20, 25, 30, 35, 40, 45, 47, 48\},$$

one observes that as much as the problem gets closer to the classical LOP (in the sense that p approaches n) the computational time reduces.

While for smaller values of p the standard solvers run out of memory due to a prohibitive memory usage of the branch-and-bound tree, for moderate values of p solver terminates due to reaching the time limit without memory issues. For larger values of p , CPLEX terminates to optimality within the time limit. An interesting point here is that when p is very close to n , problems can be solved at the root node by CPLEX.

For N-be75eec with $n = 50$ and p varying in $\{20, 30, 40, 50\}$, all the instances are solved to optimality in about 15 seconds, at most.

In N-econ36 with $n = 36$ and p varying in $\{15, 20, 25, 30, 35, 36\}$, again all the instances are solved to optimality within a few seconds.

For N-p50 with $n = 50$ and p varying in $\{10, 15, 20, 25, 30, 35, 40, 45, 49, 50\}$, while LOP instances are not yet solved to optimality, we observe that when $k \ll N$ the prohibitive branch-and-bound tree size is an obstacle and the solver terminates due to the memory issue (out of memory status). However, as p approaches n the time limit is met before proving the optimality for all the instances.

In N-pal19 with $n = 19$ and p varying in $\{5, 10, 15, 18, 19\}$, all the instances are solved to optimality while the computational times are significantly higher compared to N-be75eec, N-econ36 and N-atp24. Moreover, optimality is met after a significant number of nodes being processed, mostly because we have a high level of symmetry causing degeneracy in the course of the branch-and-bound process.

In the same class of problems from LOLIB, i.e. N-pal43 and p varying in $\{10, 15, 20, 25, 30, 35, 40, 42, 43\}$, only one instance could reach the time limit and the rest of instances ran into memory problem. A surprising point is that the maximum number of branch-and-bound nodes upon which memory

issue occurs never exceed 7013 nodes. This might be due to the high degree of degeneracy of LP at every (or certain) node.

For N-be75eec and N-econ36, except for one instance, CPLEX was able to prove the optimality at the root node.

In general, in terms of number of nodes in branch-and-bound tree, the trees tend to become smaller as p approaches n in *almost* all the instances.

The column #Nodes reports the number of nodes in the branch-and-bound tree.

Fname	Obj.Val.	Time (sec.)	#Nodes	Status
N-atp24-24-5	18	7.5	389	Optimal
N-atp24-24-10	53	58.46	847	Optimal
N-atp24-24-15	94	16.51	268	Optimal
N-atp24-24-20	140	9.65	54	Optimal
N-atp24-24-23	165	1.37	0	Optimal
N-atp24-24-24	172	0.76	0	Optimal
N-atp48-48-10	45	2883.39	5620	FailFeas
N-atp48-48-15	91	6469.86	4487	FailFeas
N-atp48-48-20	144	9262.23	5146	FailFeas
N-atp48-48-25	202	15816.2	5419	FailFeas
N-atp48-48-30	264	40152.1	3704	TimeLimit
N-atp48-48-35	321	38817.3	2232	TimeLimit
N-atp48-48-40	386	38385.5	1917	TimeLimit
N-atp48-48-45	447	12870.1	649	Optimal
N-atp48-48-47	472	223.98	0	Optimal
N-atp48-48-48	483	5.87	0	Optimal
N-be75eec-50-20	130506	14.67	12	Optimal
N-be75eec-50-30	191061	6.87	0	Optimal
N-be75eec-50-40	223672	10.68	0	Optimal
N-be75eec-50-50	236464	3.73	0	Optimal
N-econ36-36-15	400547	4.26	0	Optimal
N-econ36-36-20	469584	5.15	0	Optimal
N-econ36-36-25	506268	7.56	0	Optimal
N-econ36-36-30	534406	6.37	0	Optimal
N-econ36-36-35	547542	11.45	0	Optimal
N-econ36-36-36	548574	2.29	0	Optimal
N-p50-20-50-10	2653	9813.16	3926	FailFeas
N-p50-20-50-15	5948	15935.1	4777	FailFeas
N-p50-20-50-20	9908	17895.9	4133	FailFeas
N-p50-20-50-25	12394	10811.4	3468	FailFeas
N-p50-20-50-30	18654	28182.8	4372	FailFeas
N-p50-20-50-35	25565	38952.4	2888	TimeLimit
N-p50-20-50-40	31457	37498.2	1611	TimeLimit
N-p50-20-50-45	38416	36687.8	1468	TimeLimit

N-p50-20-50-49	45280	37315.3	2005	TimeLimit
N-p50-20-50-50	46694	39038.3	1578	TimeLimit
N-pal19-19-10	36	383.03	50002	Optimal
N-pal19-19-15	72	2297.2	127463	Optimal
N-pal19-19-18	98	1072.09	67993	Optimal
N-pal19-19-19	107	396.17	25624	Optimal
N-pal19-19-5	10	77.59	9829	Optimal
N-pal43-43-10	40	40918.2	17248	TimeLimit
N-pal43-43-15	81	12174.1	6308	FailFeas
N-pal43-43-20	138	19752.7	6839	FailFeas
N-pal43-43-25	204	22625.9	7013	FailFeas
N-pal43-43-30	280	32759.6	6341	FailFeas
N-pal43-43-35	372	38437.9	5490	FailFeas
N-pal43-43-40	466	38360.5	5371	FailFeas
N-pal43-43-42	509	26864.6	4987	FailFeas
N-pal43-43-43	535	20711	5396	FailFeas

Table 3.1: Numerical results of solving FCLOP

3.3.2 FCLOP + TotalX

The numerical experiments of solving the problem after adding **TotalX** constraint to the FCLOP model is reported in Table 3.2. For the instance **N-atp24**, one observes that computational times as well as the maximum number of nodes in the branch-and-bound tree needed to prove the optimality, decrease significantly. The same as before, all the instances of **N-atp24** are optimally solved in reasonable times and faster than before (see Table 3.1). The results concerning **N-atp48** shows that after adding **TotalX** constraint to the model, all the instances except two of them (i.e. $p = 20$ and $p = 25$) have been solved to optimality.

Table 3.1 and Table 3.2 show that the behavior of several instances like **N-be75eec**, **N-econ36**, **N-pal19** and **N-p50-20** for any p that we have examined, remain quite the same, before and after adding **TotalX** constraint.

The comparison between computational times in each set of instances are available in Figure 5.6 through Figure 3.19.

In the set of instances of **N-pal43**, we have again similarity between Table 3.1 and Table 3.2. However, there is a slight difference with regard to the number of nodes in branch-and-bound tree. One observes that in formulation with **TotalX**, CPLEX is usually able to process more nodes before running out of memory.

Fname	Obj.Val.	Time (sec.)	#Nodes	Status
N-atp24-24-5	18	1.41	5	Optimal
N-atp24-24-10	53	11.81	11	Optimal

N-atp24-24-15	94	15.96	11	Optimal
N-atp24-24-20	140	5.42	0	Optimal
N-atp24-24-23	165	1.5	0	Optimal
N-atp24-24-24	172	1	0	Optimal
N-atp48-48-10	53	1238.52	1193	Optimal
N-atp48-48-15	97	14217.3	8422	Optimal
N-atp48-48-20	144	25700.7	7196	FailFeas
N-atp48-48-25	205	40980	8846	FailFeas
N-atp48-48-30	264	33676	5645	Optimal
N-atp48-48-35	324	33302.2	4320	Optimal
N-atp48-48-40	387	5203.27	257	Optimal
N-atp48-48-45	447	1025.41	3	Optimal
N-atp48-48-47	472	261.51	0	Optimal
N-atp48-48-48	483	5.34	0	Optimal
N-be75eec-50-20	130506	15.29	8	Optimal
N-be75eec-50-30	191061	12.93	0	Optimal
N-be75eec-50-40	223672	10.56	0	Optimal
N-be75eec-50-50	236464	5.31	0	Optimal
N-econ36-36-15	400547	3.76	0	Optimal
N-econ36-36-20	469584	5.56	0	Optimal
N-econ36-36-25	506268	8.64	0	Optimal
N-econ36-36-30	534406	7.7	0	Optimal
N-econ36-36-35	547542	9.85	0	Optimal
N-econ36-36-36	548574	3.5	0	Optimal
N-p50-20-50-10	3134	3420.38	4447	FailFeas
N-p50-20-50-15	5685	5443.19	3775	FailFeas
N-p50-20-50-20	8935	11790.4	4314	FailFeas
N-p50-20-50-25	14354	27010.3	4217	FailFeas
N-p50-20-50-30	19670	28743.3	3705	FailFeas
N-p50-20-50-35	25886	37099	2088	TimeLimit
N-p50-20-50-40	32348	37414.7	2074	TimeLimit
N-p50-20-50-45	39302	34785.2	1279	TimeLimit
N-p50-20-50-49	45417	36139.6	1642	TimeLimit
N-p50-20-50-50	46597	38924.8	1639	TimeLimit
N-pal19-19-10	36	307.59	24533	Optimal
N-pal19-19-15	72	3652.47	147459	Optimal
N-pal19-19-18	98	996.75	65365	Optimal
N-pal19-19-19	107	431.93	36187	Optimal
N-pal19-19-5	10	0.53	0	Optimal
N-pal43-43-10	40	41427.8	310771	TimeLimit
N-pal43-43-15	82	13358.3	31817	FailFeas
N-pal43-43-20	137	12558.6	9908	FailFeas
N-pal43-43-25	203	17064.8	5713	FailFeas

N-pal43-43-30	282	37034.3	5794	FailFeas
N-pal43-43-35	369	37061.4	5099	FailFeas
N-pal43-43-40	468	31281.4	4987	FailFeas
N-pal43-43-42	511	24258.6	4587	FailFeas
N-pal43-43-43	533	26369.6	5143	FailFeas

Table 3.2: Numerical results for FCLOP+TotalX

3.3.3 FCLOP + Node-Degree

In Table 3.3, we examined simple FCLOPI after adding **Node-Degree** constraints. Compared to the earlier results, in some cases (e.g. **N-atp48** with $p = 15$), the instance is not solvable anymore on the same machine. That means, the model becomes more intractable after adding a significant number of constraints **Node-Degree** to the model. Also regarding **N-atp48** with $p = 20$ or $p = 25$, we are still not able to solve these instances to optimality. However, the new results in Table 3.3 show that the minimum number of branch-and-bound nodes needed now (in the FCLOP + Node-Degree) to prove optimality is less than what we needed earlier in Table 3.2. For instance the results concerning sets **N-be75eec** and **N-econ36** show that for all the instances of these sets, the computational times increase, while the number of branch-and-bound nodes decrease to only 3 nodes for **N-be75eec** with $p = 20$ and do not change much for the rest of instances.

An important point here is that we could solve **N-atp48-48-15** to optimality after adding the constraint TotalX (see Table 3.2). However it ran out of memory when we applied the set of Node-Degree constraints. That is why we prefer to use TotalX instead of the Node-Degree inequalities henceforward (keeping in mind that TotalX is equal to the aggregation of all the constraints in Node-Degree and in fact they are redundant). For the set of instances **N-p50-20** and **N-pal43**, we still did not succeed to solve any of them to optimality.

Fname	Obj.Val.	Time (sec.)	#Nodes	Status
N-atp24-24-10	53	12.12	48	Optimal
N-atp24-24-15	94	14.07	11	Optimal
N-atp24-24-20	140	6.76	0	Optimal
N-atp24-24-23	165	1.45	0	Optimal
N-atp24-24-24	172	0.95	0	Optimal
N-atp24-24-5	18	2.09	3	Optimal
N-atp48-48-10	53	836.57	487	Optimal
N-atp48-48-15	94	29603.7	11222	FailFeas
N-atp48-48-20	147	39260.2	6893	TimeLimit
N-atp48-48-25	204	38755.9	5276	TimeLimit
N-atp48-48-30	264	32398.3	5225	Optimal

N-atp48-48-35	324	28352.3	4494	Optimal
N-atp48-48-40	387	3543.59	155	Optimal
N-atp48-48-45	447	800.96	5	Optimal
N-atp48-48-47	472	264.62	0	Optimal
N-atp48-48-48	483	5.32	0	Optimal
N-be75eec-50-20	130506	20.79	3	Optimal
N-be75eec-50-30	191061	25.54	0	Optimal
N-be75eec-50-40	223672	35.84	0	Optimal
N-be75eec-50-50	236464	4.85	0	Optimal
N-econ36-36-15	400547	7.78	0	Optimal
N-econ36-36-20	469584	9.03	0	Optimal
N-econ36-36-25	506268	10.04	0	Optimal
N-econ36-36-30	534406	9.65	0	Optimal
N-econ36-36-35	547542	10.75	0	Optimal
N-econ36-36-36	548574	3.65	0	Optimal
N-p50-20-50-10	3025	9190.95	5615	FailFeas
N-p50-20-50-15	6035	15110.5	5645	FailFeas
N-p50-20-50-20	9639	22155.6	4462	FailFeas
N-p50-20-50-25	14370	37007.6	3748	TimeLimit
N-p50-20-50-30	19597	36197.2	2412	TimeLimit
N-p50-20-50-35	25615	34824.3	1421	TimeLimit
N-p50-20-50-40	31806	34811.8	1179	TimeLimit
N-p50-20-50-45	38775	33620.4	538	TimeLimit
N-p50-20-50-49	45194	35836	1597	TimeLimit
N-p50-20-50-50	46597	38458.2	1645	TimeLimit
N-pal19-19-10	36	418.17	19210	Optimal
N-pal19-19-15	72	3163.77	98493	Optimal
N-pal19-19-18	98	991.43	81540	Optimal
N-pal19-19-19	107	431.2	36187	Optimal
N-pal19-19-5	10	0.43	0	Optimal
N-pal43-43-10	40	3765.72	7015	FailFeas
N-pal43-43-15	81	3411.11	6004	FailFeas
N-pal43-43-20	136	7625.92	5883	FailFeas
N-pal43-43-25	204	30248.8	6498	FailFeas
N-pal43-43-30	280	37178	4148	TimeLimit
N-pal43-43-35	372	34785.1	1939	TimeLimit
N-pal43-43-40	471	36472.4	3895	TimeLimit
N-pal43-43-42	515	18870	4939	FailFeas
N-pal43-43-43	533	25873.4	5148	FailFeas

Table 3.3: Numerical results for the FCLOP + Node-Degree

3.3.4 FCLOP + TotalX + TF Class 1

The model FCLOP after adding TotalX constraints as well as the set of TF - Class 1 constraints is considered here.

Table 3.4 reports the statistics on the computational experiments with this model. The CPU times required to solve instances of **N-atp24**, in general, increases compared to all other earlier results while this is not the case regarding the set of **N-atp48** instances. Concerning the set of **N-atp48** instances, the number of instances solved to optimality is decreased to only 5 instances.

The behavior of the set of **N-be75eec** and **N-econ36** instances is almost the same as what we had in Table 3.3.

Concerning instances of **N-p50-20**, we are still unable to solve any of these instances. However, the number of nodes in branch-and-bound tree is quite less than what we observed in Table 3.3.

All the instances **N-pal19** are again optimally solved while the number of nodes in branch-and-bound trees increased. Finally for the set of instances **N-pal43**, we see that except the last one, CPLEX has terminated upon reaching the time limit.

Fname	Obj.Val.	Time	#Nodes	Status
N-atp24-24-10	53	120.09	10	Optimal
N-atp24-24-15	94	57.64	0	Optimal
N-atp24-24-20	140	6.14	0	Optimal
N-atp24-24-23	165	1.82	0	Optimal
N-atp24-24-24	172	1.26	0	Optimal
N-atp24-24-5	18	7.82	0	Optimal
N-atp48-48-10	53	25361.1	963	Optimal
N-atp48-48-15	94	35598.2	453	TimeLimit
N-atp48-48-20	134	21579.6	0	TimeLimit
N-atp48-48-25	195	21581.3	0	TimeLimit
N-atp48-48-30	258	25597.6	26	TimeLimit
N-atp48-48-35	324	31556.5	143	TimeLimit
N-atp48-48-40	387	19407.5	47	Optimal
N-atp48-48-45	447	688.1	0	Optimal
N-atp48-48-47	472	60.06	0	Optimal
N-atp48-48-48	483	8.84	0	Optimal
N-be75eec-50-20	130506	56.29	0	Optimal
N-be75eec-50-30	191061	25.26	0	Optimal
N-be75eec-50-40	223672	31.67	0	Optimal

N-be75eec-50-50	236464	7.71	0	Optimal
N-econ36-36-15	400547	8.75	0	Optimal
N-econ36-36-20	469584	4.82	0	Optimal
N-econ36-36-25	506268	7.12	0	Optimal
N-econ36-36-30	534406	6.45	0	Optimal
N-econ36-36-35	547542	12.21	0	Optimal
N-econ36-36-36	548574	4.82	0	Optimal
N-p50-20-50-10	3170	31818.1	528	TimeLimit
N-p50-20-50-15	5625	27025.9	56	TimeLimit
N-p50-20-50-20	9366	21577.4	0	TimeLimit
N-p50-20-50-25	14511	21579.2	0	TimeLimit
N-p50-20-50-30	19023	21580	0	TimeLimit
N-p50-20-50-35	25201	21596.9	0	TimeLimit
N-p50-20-50-40	31731	21581.8	0	TimeLimit
N-p50-20-50-45	38363	22907.7	11	TimeLimit
N-p50-20-50-49	42351	30239.1	65	TimeLimit
N-p50-20-50-50	46615	38730.2	1317	TimeLimit
N-pal19-19-10	36	443.71	6201	Optimal
N-pal19-19-15	72	4182.84	54933	Optimal
N-pal19-19-18	98	3182.83	71069	Optimal
N-pal19-19-19	107	546.17	45416	Optimal
N-pal19-19-5	10	1.23	0	Optimal
N-pal43-43-10	40	36271	603	TimeLimit
N-pal43-43-15	81	35185.6	596	TimeLimit
N-pal43-43-20	135	33396.2	592	TimeLimit
N-pal43-43-25	204	34969.3	416	TimeLimit
N-pal43-43-30	279	21070.4	2	TimeLimit
N-pal43-43-35	370	35437	558	TimeLimit
N-pal43-43-40	472	32510.5	526	TimeLimit
N-pal43-43-42	510	34393.1	731	TimeLimit
N-pal43-43-43	534	25559	4111	FailFeas

Table 3.4: Numerical results of solving FCLOP + TotalX + TF Class 1

3.3.5 FCLOP + Node-Degree + TF Class 1

Table 3.5 reports the experimental results of a model composed of FCLOP and those constraints TF Class 1 and Node-Degree.

One observes in Table 3.5 that the instances N-atp24 for $k = 10$ and 15, the computational times needed to prove optimality do not significantly change when compared to Table 3.4.

For the instance set **N-atp48**, we observe that the number of instances solved to optimality is the same as what we had in Table 3.4 while it has decreased compared to observations in Table 3.2 and Table 3.3.

All the instances of sets **N-be75eec** and **N-econ36** are optimally solved in reasonable times.

Comparison of computational time required for any of the individual instances are available in Figure 5.6 through Figure 3.19.

Table 3.5 shows that except for **N-pal43-43-43**, for all the instances of sets **N-p50-20** and **N-pal43**, CPLEX has terminated upon reaching the time limit.

Fname	Obj.Val.	Time (sec.)	#Nodes	Status
N-atp24-24-10	53	53.75	11	Optimal
N-atp24-24-15	94	50.09	0	Optimal
N-atp24-24-20	140	2.18	0	Optimal
N-atp24-24-23	165	1.75	0	Optimal
N-atp24-24-24	172	1.28	0	Optimal
N-atp24-24-5	18	8.53	0	Optimal
N-atp48-48-10	53	19893.3	318	Optimal
N-atp48-48-15	86	27475.7	71	TimeLimit
N-atp48-48-20	113	21594.5	0	TimeLimit
N-atp48-48-25	196	21594.2	0	TimeLimit
N-atp48-48-30	238	20913.2	2	TimeLimit
N-atp48-48-35	295	27986.6	46	TimeLimit
N-atp48-48-40	387	2470.77	5	Optimal
N-atp48-48-45	447	87.65	0	Optimal
N-atp48-48-47	472	58.65	0	Optimal
N-atp48-48-48	483	8.6	0	Optimal
N-be75eec-50-20	130506	34.03	0	Optimal
N-be75eec-50-30	191061	44.18	0	Optimal
N-be75eec-50-40	223672	50.95	0	Optimal
N-be75eec-50-50	236464	7.4	0	Optimal
N-econ36-36-15	400547	6.76	0	Optimal
N-econ36-36-20	469584	9.32	0	Optimal
N-econ36-36-25	506268	13.1	0	Optimal
N-econ36-36-30	534406	11.5	0	Optimal
N-econ36-36-35	547542	12.28	0	Optimal
N-econ36-36-36	548574	4.65	0	Optimal
N-p50-20-50-10	3342	34457.4	397	TimeLimit
N-p50-20-50-15	5616	23004.5	16	TimeLimit
N-p50-20-50-20	9131	21617.1	0	TimeLimit
N-p50-20-50-25	13981	21594.7	0	TimeLimit
N-p50-20-50-30	17337	21594.3	0	TimeLimit
N-p50-20-50-35	23905	21594.6	0	TimeLimit
N-p50-20-50-40	30859	21596.9	0	TimeLimit

N-p50-20-50-45	37338	25485.7	24	TimeLimit
N-p50-20-50-49	44419	26888.6	42	TimeLimit
N-p50-20-50-50	46615	39237.4	1415	TimeLimit
N-pal19-19-10	36	412.85	4030	Optimal
N-pal19-19-15	72	4997.97	45018	Optimal
N-pal19-19-18	98	3624.03	84875	Optimal
N-pal19-19-19	107	542.34	45416	Optimal
N-pal19-19-5	10	1.61	0	Optimal
N-pal43-43-10	39	33032.2	175	TimeLimit
N-pal43-43-15	83	35462.6	391	TimeLimit
N-pal43-43-20	137	36134.1	337	TimeLimit
N-pal43-43-25	200	33440.2	301	TimeLimit
N-pal43-43-30	278	21596.5	0	TimeLimit
N-pal43-43-35	357	30175.8	49	TimeLimit
N-pal43-43-40	471	35009.3	376	TimeLimit
N-pal43-43-42	512	35876.3	836	TimeLimit
N-pal43-43-43	534	25691.2	4153	FailFeas

Table 3.5: Numerical results of solving FCLOP + Node-Degree + TF Class 1

3.3.6 FCLOP + TF Class 2

In Table 3.6, one observes that **N-atp48** for $p = 20$ and $p = 25$ are two instances that have been solved to optimality for the first time. Moreover, the instance **N-p50-20** for $p = 10$ also has been solved to optimality, which has not been solved to optimality before, for any p .

Another interesting fact is that for 18 instances of **N-p50-20** and **N-pal43** – which their optimal solution are not known – we could significantly improve the upper bound as well as the lower bound.

We have improved the gap from 2.94% to 1.58% for LOP model of **N-p50-20**. The best known upper and lower bounds available for **N-p50-20** in the literature is [46779, 48155] and the new upper bound we have obtained is 47518.

Fname	Obj.Val.	Time	#Nodes	LP bound	Optimal*	Status
N-atp24-24-5	18	1.23	3	41.04	18	Optimal
N-atp24-24-10	53	18.01	3	82.08	53	Optimal
N-atp24-24-15	94	11.65	0	123.12	94	Optimal
N-atp24-24-20	140	2.31	0	153.16	140	Optimal
N-atp24-24-23	165	0.90	0	168.25	165	Optimal
N-atp24-24-24	172	0.57	0	172	172	Optimal
N-atp48-48-10	53	8727.7	42	118.54	53	Optimal
N-atp48-48-15	97	24993.7	87	177.81	97	Optimal

N-atp48-48-20	148	46292.3	133	237.08	148	Optimal
N-atp48-48-25	205	35265.6	37	296.35	205	Optimal
N-atp48-48-30	264	5863.47	0	355.62	264	Optimal
N-atp48-48-35	324	22453.3	33	400.65	324	Optimal
N-atp48-48-40	387	78.5	0	436.16	387	Optimal
N-atp48-48-45	447	71.65	0	466.62	447	Optimal
N-atp48-48-47	472	49.51	0	477.7	472	Optimal
N-atp48-48-48	483	8.40	0	483	483	Optimal
N-be75eec-50-20	130506	24.09	0	132966	130506	Optimal
N-be75eec-50-30	191061	13	0	191168	191061	Optimal
N-be75eec-50-40	223672	14.23	0	224006	223672	Optimal
N-be75eec-50-50	236464	5.57	0	236464	236464	Optimal
N-econ36-36-15	400547	8.06	0	403222	400547	Optimal
N-econ36-36-20	469584	3.04	0	470033	469584	Optimal
N-econ36-36-25	506268	5.42	0	507568	506232	Optimal
N-econ36-36-30	534406	3.81	0	535017	534406	Optimal
N-econ36-36-35	547542	6.18	0	547984	547501	Optimal
N-econ36-36-36	548574	3.34	0	548588	548574	OptimalTol
N-p50-20-50-10	3342	173259	1530	13282.6	2311*	Optimal
N-p50-20-50-15	6296	401847	1045	19923.9	5106*	FailFeas
N-p50-20-50-20	10049	381111	659	26565.2	7397*	TimeLimit
N-p50-20-50-25	14746	374291	580	33206.5	13697*	TimeLimit
N-p50-20-50-30	20144	345978	395	39847.8	17052*	TimeLimit
N-p50-20-50-35	25706	368863	525	44672.5	23298*	TimeLimit
N-p50-20-50-40	32584	374226	585	45864.2	31337*	TimeLimit
N-p50-20-50-45	39147	334478	913	47055.8	38099*	FailFeas
N-p50-20-50-49	44932	138277	986	47985.7	42450*	FailFeas
N-p50-20-50-50	47518	104270	6380	48155.7	46779*	FailFeas
N-pal19-19-5	10	0.62	0	45	10	Optimal
N-pal19-19-10	36	203.73	1612	90	36	Optimal
N-pal19-19-15	72	5685.59	64098	114	72	Optimal
N-pal19-19-18	98	2244.66	52019	114	98	Optimal
N-pal19-19-19	107	398.20	35861	114	107	Optimal
N-pal43-43-10	40	95272	2041	210	40*	FailFeas
N-pal43-43-15	82	111463	2119	315	80*	FailFeas
N-pal43-43-20	137	148396	1719	420	139*	FailFeas
N-pal43-43-25	207	296861	2605	525	199*	FailFeas
N-pal43-43-30	281	220975	2590	602	276*	FailFeas
N-pal43-43-35	371	248244	2546	602	361*	FailFeas
N-pal43-43-40	473	181788	2629	602	459*	FailFeas
N-pal43-43-42	511	90379.1	2693	602	500*	FailFeas
N-pal43-43-43	535	14477	6371	602	535*	FailFeas

Table 3.6: Numerical results of solving FCLOP + TotalX +
TF Class 2

3.3.7 Bound quality comparison

In Table 3.7, the first column is the instance name as before, the second column (*FCLOP*) represent the LP bound of the basic FCLOP model. The next columns *TF1*, *TF2* and *TF3* represent the LP bound after adding the inequalities TF Class 1 (TF1), TF Class 2 (TF2) and TF Class 3 (TF3) to the model, respectively. The next column, *TF2 + TF3*, reports the bound obtained by the LP relaxation of the FCLOP model after adding the two sets of inequalities TF2 and TF3. The reason for reporting this column is to compare the quality of linear relaxation of the model with the two previous cases that we only add TF2 (or TF3) to the basic FCLOP model. The next column, *TF2+Var* reports the LP bound obtained after adding the set of inequalities TF2, (3.27) and (3.28) to the FCLOP model.

One observes that triangle-free inequalities TF2 are producing tighter bound compared to the TF1 and TF3. When we add the inequalities TF2 together with (3.27) and (3.28) we improve the LP bound for some difficult instances e.g. N-p50-20_50 with various p values.

Fname	FCLOP	TF1	TF2	TF3	TF2+TF3	TF2+Var
N-atp24_24_5	41.04	21.08	19.17	21.26	19.17	18.11
N-atp24_24_10	82.08	65.79	53.93	68.1	53.93	53
N-atp24_24_15	123.12	101.36	94.43	107.65	94.43	94
N-atp24_24_20	153.16	140	140	157	140	140
N-atp24_24_22	161.1	157	157	157	157	157
N-atp24_24_23	168.25	165	165	165	165	165
N-atp24_24_24	172 172	172	172	172	172	
N-atp48_48_10	118.54	71.26	58.20	72.2	58.2	54.9
N-atp48_48_15	177.81	137.81	104.36	142.33	104.36	99.65
N-atp48_48_20	237.08	201.57	155.21	222.17	155.21	150.43
N-atp48_48_25	296.35	253.41	209.3	288.24	209.3	205.55
N-atp48_48_30	355.62	299.82	265.86	334.28	265.86	264
N-atp48_48_35	400.65	344.62	324.64	382.09	324.64	324
N-atp48_48_40	436.16	391.34	387	449	387	387
N-atp48_48_45	466.62	447	447	525	447	447
N-atp48_48_46	471	459	459	459	459	459
N-atp48_48_47	477.7	472	472	472	472	472
N-atp48_48_48	483	483	483	483	483	483
N-be75eec_50_20	132966	130506	130506	132979	130506	130506
N-be75eec_50_30	191168	191061	191061	195826	191061	191061

N-be75eec_50_40	224006	223672	223672	230885	223672	223672
N-be75eec_50_48	236464	236345	236345	236345	236345	236345
N-be75eec_50_49	236464	236464	236464	236464	236464	236464
N-be75eec_50_50	236464	236464	236464	236464	236464	236464
N-econ36_36_15	403222	400547	400547	401589	400547	400547
N-econ36_36_20	470033	469584	469584	473682	469584	469584
N-econ36_36_25	507568	506268	506268	514152	506268	506268
N-econ36_36_30	535017	534406	534406	544978	534406	534406
N-econ36_36_34	546766	545576	545576	545576	545576	545576
N-econ36_36_35	547984	547556	547556	547556	547556	547556
N-econ36_36_36	548588	548588	548588	548588	548588	548588
N-p50-20_50_10	13282.6	4451.88	4001.91	4491.63	4001.91	3961.62
N-p50-20_50_15	19923.9	9487.72	8083.14	9750.19	8083.14	8053.35
N-p50-20_50_20	26565.2	15428.6	12542.3	16498.8	12542.3	12501.8
N-p50-20_50_25	33206.5	21645.4	17370.7	24303.5	17370.7	17149.7
N-p50-20_50_30	39847.8	26990.6	22619.6	29889.6	22619.6	22374.5
N-p50-20_50_35	44672.5	32557.2	28370.7	36807.3	28370.7	28192.1
N-p50-20_50_40	45864.2	37747.7	34634	45330.9	34634	34558.5
N-p50-20_50_45	47055.8	42673.6	41379.4	55601	41379.4	41376.8
N-p50-20_50_48	47771.1	45820.5	45518	45518	45518	45518
N-p50-20_50_49	47985.7	46959.7	46856.3	46856.3	46856.3	46856.3
N-p50-20_50_50	48155.7	48155.7	48155.7	48155.7	48155.7	48155.7
N-pal19_19_5	45	10	10	10	10	10
N-pal19_19_10	90	45	45	45	45	45
N-pal19_19_15	114	90	80	105	80	80
N-pal19_19_17	114	102	96.33	96	96	96.33
N-pal19_19_18	114	108	105	105	105	105
N-pal19_19_19	114	114	114	114	114	114
N-pal43_43_10	210	45	45	45	45	45
N-pal43_43_15	315	105	105	105	105	105
N-pal43_43_20	420	190	190	190	190	190
N-pal43_43_25	525	300	275	300	275	275
N-pal43_43_30	602	420	355	435	355	355
N-pal43_43_35	602	490	443.33	595	443.333	443.33
N-pal43_43_40	602	560	540	780	540	540
N-pal43_43_41	602	574	560.33	560	560	560.33
N-pal43_43_42	602	588	581	581	581	581
N-pal43_43_43	602	602	602	602	602	602

Table 3.7: numerical experiment-Integrality gap comparison.

In the following, Figure 3.13 through Figure 3.19 correspond to 7 groups of instances N-atp24, N-atp48, N-be75ec, N-econ36 N-p50-20 N-pal19 and N-pal19 are presented, which compare the computational time required to solve every instances with different p values. From what we have observed so far, the behavior of different instances are highly different. Hence, we consider instances separately and report for different p values in each figure. Note that some of the instances are not solved in optimality and runs have been terminated by time limit or out of memory status. We have also considered these instances in these diagrams.

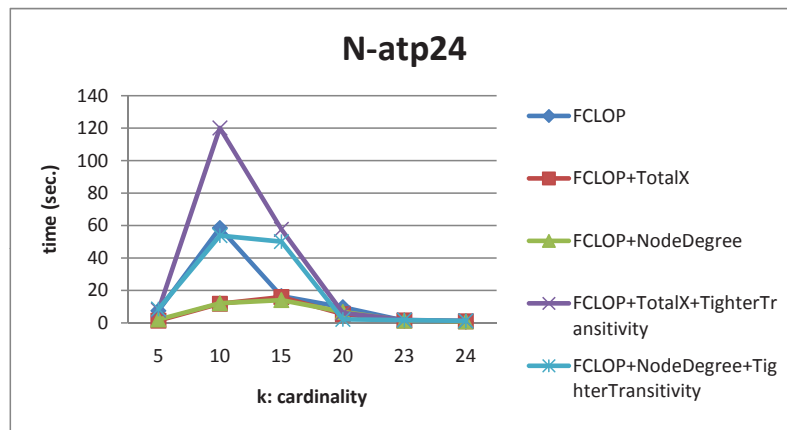


Figure 3.13: N-atp24

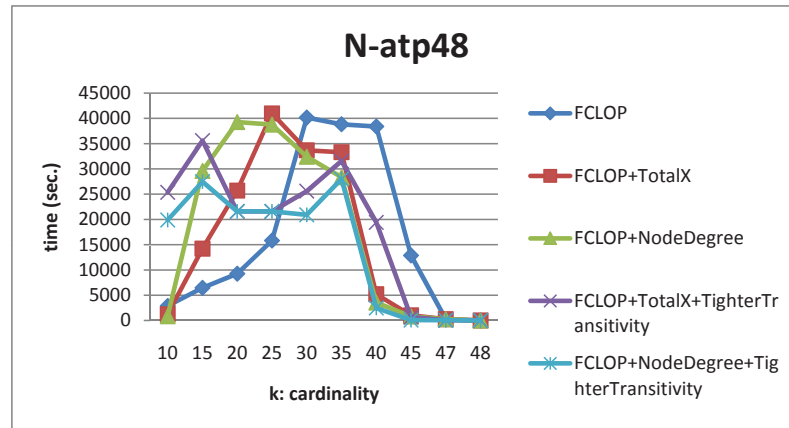


Figure 3.14: N-atp48

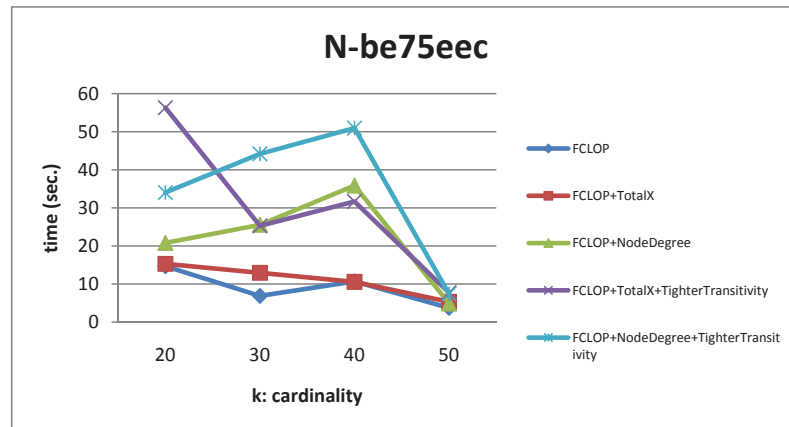


Figure 3.15: N-be75eec

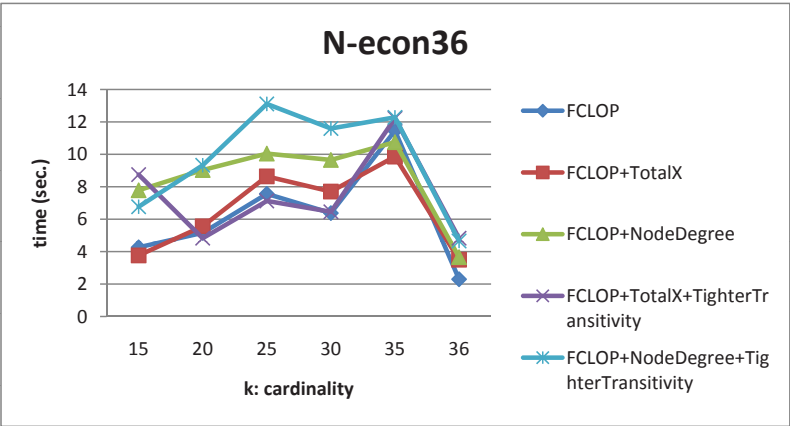


Figure 3.16: N-econ36

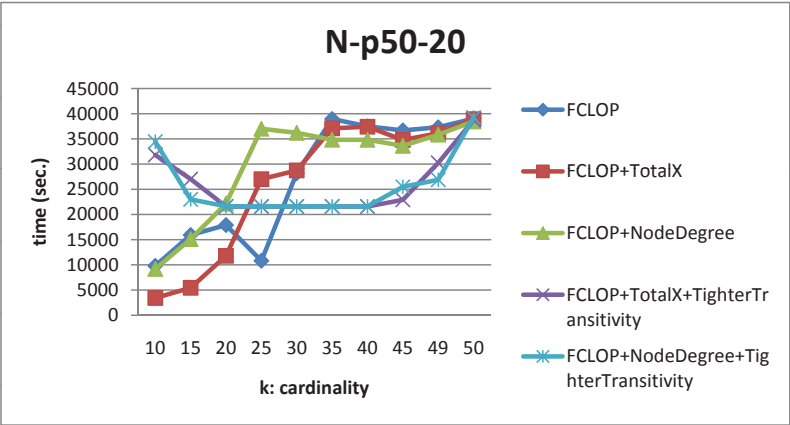


Figure 3.17: N-p50-20

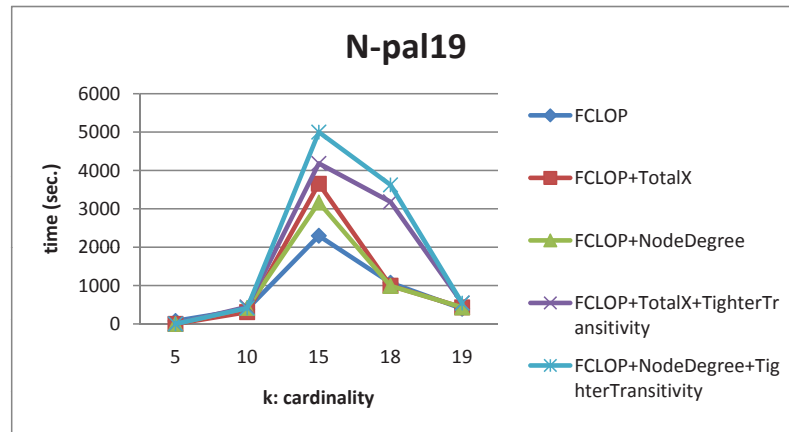


Figure 3.18: N-pal19

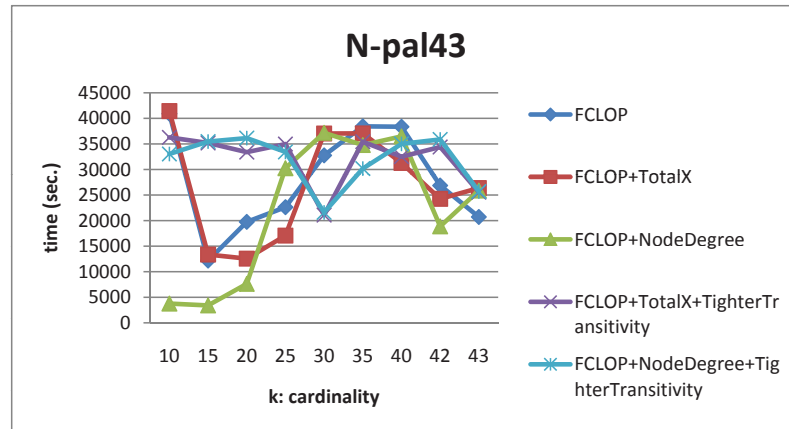


Figure 3.19: N-pal43

CHAPTER 4

Polyhedral Study for $FCLOP(G_n, p)$

In chapter 3, we have proposed several sets of valid inequalities for the FCLOP polytope. In this chapter, our investigation on the polytopes of the FCLOP for different p values is presented. We have determined the dimension of the FCLOP polytope and characterized some classes of facet defining inequalities.

Numerical results already proved the significant impact of adding these facet defining inequalities to the model. Later in Chapter chapter 6 a Relax-and-Cut algorithm will be presented to apply the results presented here. In the following we present an integer programming formulation for the FCLOP for the cases (i) $p = n - 1$ and (ii) $p \leq n - 2$:

- (i) $FCLOP_{n-1}$:

$$\max \sum_{i=1}^n \sum_{j=1, j \neq i}^n w_{ij} x_{ij} \quad (4.1)$$

s.t.

$$x_{ij} + x_{ji} = y_i + y_j - 1 \quad \forall i, j \in V_n, i \leq j, \quad (4.2)$$

$$\sum_{i=1}^n y_i = n - 1, \quad (4.3)$$

$$x_{ij} + x_{jk} - x_{ik} - y_j \leq 0 \quad \forall \{i, j, k\} \subseteq V_n, \quad (4.4)$$

$$0 \leq x_{ij} \quad \forall i, j \in V_n, i \neq j, \quad (4.5)$$

$$y_i \leq 1 \quad \forall i \in V_n, \quad (4.6)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n. \quad (4.7)$$

- (ii) FCLOP_p:

$$\max \sum_{i=1}^n \sum_{j=1, j \neq i}^n w_{ij} x_{ij} \quad (4.8)$$

s.t.

$$x(\delta(i)) = (p-1)y_i \quad \forall i, j \in V_n, i \leq j, \quad (4.9)$$

$$\sum_{i=1}^n y_i = p, \quad (4.10)$$

$$x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V_n, i \leq j, \quad (4.11)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V_n, i \leq j, \quad (4.12)$$

$$x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall i, j \in V_n, i \leq j, \quad (4.13)$$

$$x_{ij} + x_{jk} - x_{ik} - y_j \leq 0 \quad \forall \{i, j, k\} \subseteq V_n, \quad (4.14)$$

$$x_{ij} + x_{jk} + x_{ki} + \sum_{i \in V_n \setminus \{i, j, k\}} y_i \leq \lfloor \frac{2p + (n-3)}{3} \rfloor \quad \forall \{i, j, k\} \subseteq V_n, \quad (4.15)$$

$$0 \leq x_{ij} \quad \forall i, j \in V_n, i \neq j, \quad (4.16)$$

$$y_i \leq 1 \quad \forall i \in V_n, \quad (4.17)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n. \quad (4.18)$$

Some basic definitions related to polyhedra are followed.

The set of solutions of a finite system of linear (in)equalities, is called a polyhedron. A *polytope* is a bounded polyhedron. An inequality $\gamma^T x \leq \gamma_0$, is *valid* for the polytope P if $P \subset \{x : \gamma^T x \leq \gamma_0\}$. If $\gamma^T x \leq \gamma_0$ is a valid inequality for P , then the set $F = \{x \in P : \gamma^T x = \gamma_0\}$ is called a face of P . The dimension of a polytope P , denoted by $\dim(P)$ is the maximum number of affinely independent points in P minus 1. A face of dimension 0 is called an extreme point and a face of dimension $\dim(P) - 1$ is called a facet.

If there is no risk of confusion, we may simplify the notation by using V instead of V_n (A instead of A_n).

4.1 Dimension

Remark 3 If $\alpha^T x + \beta^T y = \rho$ is an hyperplan containing $FCLOP(G_n, p)$, then $\alpha_{ij} = \alpha_{ji}$ for any $i, j \in V_n, i \neq j$.

Theorem 1 Given $G_n = (V_n, A_n)$ and p , with $n \geq 3$ and $p = n - 1$, then the system

$$x_{ij} + x_{ji} = y_i + y_j - 1 \text{ for all } i, j \in V_n, i < j, \quad (4.19)$$

$$\sum_{i \in V_n} y_i = p, \quad (4.20)$$

is a minimal equation system for $FCLOP(G_n, p)$.

Proof It is easy to check that equations (4.19) and (4.20) are valid for the $FCLOP$ polytope and are linearly independent. Therefore to prove the theorem it suffices to show that any other equation

$$\alpha^T x + \beta^T y = \rho, \quad (4.21)$$

with $FCLOP(G_n, p) \subseteq \{(x, y) \in \mathbb{R}^{|n^2|} : \alpha^T x + \beta^T y = \rho\}$, is a linear combination of equations (4.19)-(4.20).

Let $S_i = V_n \setminus \{i\}$, for each $i \in V_n$. Let π_{S_1} be any LO on S_1 . Let $\pi_{S_i}, i \neq 1$, be the LO defined from π_{S_1} as follows: $\pi_{S_i}^{-1}(j) = \pi_{S_1}^{-1}(j)$ if $j \notin \{1, i\}$ and $\pi_{S_i}^{-1}(1) = \pi_{S_1}^{-1}(i)$. From (4.21), we have

$$\alpha^T x^{\pi_{S_1}} + \beta^T y^{\pi_{S_1}} = \rho, \quad (4.22)$$

$$\alpha^T x^{\pi_{S_i}} + \beta^T y^{\pi_{S_i}} = \rho, \quad \text{for each } i \in V_n \setminus \{1\}. \quad (4.23)$$

Now combining carefully (4.22) with (4.23) for each $i \in V_n \setminus \{1\}$ and with the fact that $\alpha_{ij} = \alpha_{ji}$ for any i and j , we obtain

$$\beta_1 + \frac{1}{2}\alpha(\delta(1)) = \beta_i + \frac{1}{2}\alpha(\delta(i)) \quad \text{for each } i \in V_n \setminus \{1\} \quad (4.24)$$

Now multiply each equation (4.19) by α_{ij} and do the sum over all $i, j \in V_n, i < j$, then using the fact $\alpha_{ij} = \alpha_{ji}$ we get

$$\alpha^T x = \frac{1}{2}\alpha(\delta(1))y_1 + \sum_{i \in V_n \setminus \{1\}} \frac{1}{2}\alpha(\delta(i))y_i - \frac{1}{2}\alpha(A_n) \quad (4.25)$$

For each $i \in V_n \setminus \{1\}$, replace $\alpha(\delta(i))$ in (4.25) by its value in (4.24),

$$\alpha^T x + \sum_{i \in V_n \setminus \{1\}} \beta_i y_i = \frac{1}{2}\alpha(\delta(1))y_1 + (\beta_1 + \frac{1}{2}\alpha(\delta(1))) \sum_{i \in V_n \setminus \{1\}} y_i - \frac{1}{2}\alpha(A_n) \quad (4.26)$$

But from (4.20) we know that $\sum_{i \in V_n \setminus \{1\}} y_i = (n-1) - y_1$ and hence (4.26) may be rewritten as

$$\alpha^T x + \beta^T y = (n-1)(\beta_1 + \frac{1}{2}\alpha(\delta(1))) - \frac{1}{2}\alpha(A_n)$$

□

Theorem 2 Given $G_n = (V_n, A_n)$ and p , with $n \geq 4$ and $2 \leq p \leq n-2$, then the system

$$x(\delta(i)) = (p-1)y_i \text{ for all } i \in V_n, \quad (4.27)$$

$$\sum_{i \in V_n} y_i = p, \quad (4.28)$$

is a minimal equation system for $FCLOP(G_n, p)$.

Proof As in Theorem 1 it easy to see that (4.27)-(4.28) are valid for the $FCLOP$ polytope and that are linearly independent. It suffices to show that any equation

$$\alpha^T x + \beta^T y = \rho, \quad (4.29)$$

with $FCLOP(G_n, p) \subseteq \{(x, y) \in \mathbb{R}^{|n^2|} : \alpha^T x + \beta^T y = \rho\}$, is a linear combination of equations (4.27)-(4.28).

We have two cases to consider (i) $p = n-2$ and (ii) $p \leq n-3$.

(i) $p = n-2$. Let us fix a node $l \in V_n$ and consider the $FCLOP$ instance on $G_{n-1}^l = (V_{n-1}^l, A_{n-1}^l)$ with the constant p unchanged that is $p = n-2$, where $V_{n-1}^l = V_n \setminus \{l\}$ and $A_{n-1}^l = \{(i, j) \in V_{n-1}^l \times V_{n-1}^l : i \neq j\}$. Any linear ordering π_S with respect to G_{n-1}^l and p is also a linear ordering with respect to G_n and p . Therefore

$$\begin{aligned} & FCLOP(G_{n-1}^l, p) \\ & \subseteq \\ & \{(x, y) \in \mathbb{R}^{|(n-1)^2|} : \sum_{i, j \in V_{n-1}^l, i \neq j} \alpha_{ij} x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i = \rho\}. \end{aligned}$$

From Theorem 1, the equality

$$\sum_{i, j \in V_{n-1}^l, i \neq j} \alpha_{ij} x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i = \rho \quad (4.30)$$

may be obtained as a linear combination of the following equalities

$$x_{ij} + x_{ji} = y_i + y_j - 1 \text{ for all } i, j \in V_{n-1}^l \text{ and } i < j, \quad (4.31)$$

$$\sum_{i \in V_{n-1}^l} y_i = p, \quad (4.32)$$

Define such a linear combination as follows: multiply each equality (4.31) by the corresponding scalar α_{ij} and multiply equality (4.32) by a scalar σ_l . The sum of the resulting equations gives

$$\sum_{i,j \in V_{n-1}^l, i \neq j} \alpha_{ij} x_{ij} + \sum_{i \in V_{n-1}^l} (\sigma_l - \frac{1}{2}\alpha(\delta(i)) + \alpha_{il}) y_i = \sigma_l p - \frac{1}{2}\alpha(A_{n-1}^l). \quad (4.33)$$

From (4.30) and (4.33) we have

$$\rho = \sigma_l p - \frac{1}{2}\alpha(A_{n-1}^l), \quad (4.34)$$

$$\beta_i = \sigma_l - \frac{1}{2}\alpha(\delta(i)) + \alpha_{il} \quad \text{for each } i \in V_{n-1}^l. \quad (4.35)$$

Notice that these equalities are true for any $l \in V_n$.

Hence let $i \in V_n \setminus \{l, k\}$ and write (4.35) with respect to i and k , we get

$$\beta_i = \sigma_k - \frac{1}{2}\alpha(\delta(i)) + \alpha_{ik}. \quad (4.36)$$

The combination of (4.35) and (4.36) gives

$$\alpha_{il} - \alpha_{ik} = \sigma_k - \sigma_l. \quad (4.37)$$

This is true for any $i \in V_n$ and $k, l \in V_n \setminus \{i\}$, $k \neq l$. Thus we may write

$$\alpha_{ij} = \lambda_i - \sigma_j \quad \text{for any } i, j \in V_n \text{ and } i \neq j,$$

where λ_i for $i \in V_n$ is a fixed scalar.

Notice that for any $i, j \in V_n$ we have

$$\begin{aligned} \alpha_{ij} &= \lambda_i - \sigma_j, \\ \alpha_{ji} &= \lambda_j - \sigma_i. \end{aligned}$$

With Remark 3 we have that

$$\alpha_{ij} = \frac{\lambda_i - \sigma_i}{2} + \frac{\lambda_j - \sigma_j}{2}$$

Finally we may assume that

$$\alpha_{ij} = \mu_i + \mu_j \text{ for any } i, j \in V_n, i \neq j, \quad (4.38)$$

where $\mu_i = \frac{\lambda_i - \sigma_i}{2}$ for any $i \in V_n$.

The combination of (4.34) and (4.35) for some $i \in V_{n-1}^l$ gives

$$\begin{aligned} \beta_i &= \frac{\rho}{p} + \frac{\alpha(A_{n-1}^l)}{2p} - \frac{1}{2}\alpha(\delta(i)) + \alpha_{il} \\ &= \frac{\rho}{p} + \frac{2(n-2) \sum_{j \in V_{n-1}^l} \mu_j}{2(n-2)} - (n-2)\mu_i - \sum_{j \in V_{n-1}^l} \mu_j + \mu_i \\ &= \frac{\rho}{p} - (n-2)\mu_i + \mu_i, \end{aligned}$$

and since $p = n - 2$ we obtain the following,

$$\beta_i = \frac{\rho}{p} - \mu_i(p-1). \quad (4.39)$$

Now it is straightforward to check, by using (4.38) and (4.39), that equality (4.29) is obtained as a linear combination of (4.27) and (4.28), where each equality (4.27) with respect to i is multiplied by μ_i and equality (4.28) is multiplied by $\frac{\rho}{p}$.

(ii) $p \leq n - 3$. This case will be proved by induction on the size of V_n . Notice that here we must have $n \geq 5$. The basic case of the induction is obtained when $p = n - 2$, which is true from the previous case (i).

By the induction hypothesis the theorem is true for any instance of the FCLOP, $G_{n'} = (V_{n'}, A_{n'})$ with $2 \leq p \leq n - 3$ and $p + 2 \leq n' < n$. Therefore we may assume that the theorem is true for $G_{n-1}^1 = (V_{n-1}^1, A_{n-1}^1)$ and p where $p \leq (n-1) - 2$. As before $V_{n-1}^1 = \{2, \dots, n\}$ and $A_{n-1}^1 = \{(i, j) \in V_{n-1}^1 \times V_{n-1}^1 : i \neq j\}$.

Notice that any LO π_S with respect to the instance G_{n-1}^1 and p is also a LO with respect to the instance G_n and p and hence

$$\begin{aligned} &FCLOP(G_{n-1}^1, p) \\ &\subseteq \\ &\{(x, y) \in \mathbb{R}^{(n-1)^2} : \sum_{i=2}^n \sum_{j=2, i \neq j}^n \alpha_{ij} x_{ij} + \sum_{i=2}^n \beta_i y_i = \rho\}. \end{aligned}$$

Thus by the induction hypothesis, the equation

$$\sum_{i=2}^n \sum_{j=2, i \neq j}^n \alpha_{ij} x_{ij} + \sum_{i=2}^n \beta_i y_i = \rho, \quad (4.40)$$

may be obtained as a linear combination of the following equations

$$\sum_{j=2, i \neq j}^n x_{ij} + x_{ji} = (p-1)y_i \quad \text{for all } i = 2, \dots, n, \quad (4.41)$$

$$\sum_{i=2}^n y_i = p. \quad (4.42)$$

Thus if we sum the equations (4.41) with respect to $i = 2, \dots, n$ previously multiplied by a scalar λ_i with the equation (4.42) multiplied by the scalar σ , and we combine with (4.40), we get the following:

$$\alpha_{ij} = \lambda_i + \lambda_j \quad \text{for } i = 2, \dots, n, \ j = 2, \dots, n, \ i \neq j, \quad (4.43)$$

$$\beta_i = \sigma - \lambda_i(p-1) \quad \text{for } i = 2, \dots, n. \quad (4.44)$$

Now let S be any subset of V_{n-1}^1 with $|S|=p$. Let π_S be any LO on S . Consider the subset $S_k = (S \setminus \{k\}) \cup \{1\}$ and let π_k be any LO on S_k . From (4.29), we get

$$\begin{aligned} \alpha^T x^{\pi_S} + \beta^T y^{\pi_S} &= \rho, \\ \alpha^T x^{\pi_{S_k}} + \beta^T y^{\pi_{S_k}} &= \rho, \quad \text{for each } k \in S. \end{aligned}$$

Fix k and combine these last two equations with (4.43), then using Remark 3, we obtain

$$\beta_1 + \sum_{j \in S \setminus \{k\}} \alpha_{1j} = \beta_k + (p-1)\lambda_k + \sum_{j \in S \setminus \{k\}} \lambda_j$$

Replacing by the value of β_k in (4.44), we finally obtain that

$$\beta_1 + \sum_{j \in S \setminus \{k\}} \alpha_{1j} = \sigma + \sum_{j \in S \setminus \{k\}} \lambda_j, \quad \text{for each } S \subseteq V_{n-1}^1, \ k \in S. \quad (4.45)$$

Take some element $l \in S$ and write (4.45) with respect to S and l . Then by combining it with (4.45) which is with respect to S and k , we get $\alpha_{1l} - \alpha_{1k} = \lambda_l - \lambda_k$. This hold for any two elements l and k in V_{n-1}^1 . Hence we may assume that there is a constant scalar, that we call λ_1 for convenience, such that

$$\alpha_{1j} = \lambda_1 + \lambda_j \quad \text{for each } j = 2, \dots, n$$

and with Remark 3 we also have

$$\alpha_{j1} = \lambda_1 + \lambda_j \quad \text{for each } j = 2, \dots, n$$

Thus replacing the values of α_{1j} in (4.45) we have $\beta_1 = \sigma - (p-1)\lambda_1$. This complete the proof of case (ii) when $p \leq n-3$. \square

It is well known from Grötschel et al. (1985a) that the dimension of the $FCLOP$ polytope when $p = n$ is $\frac{1}{2}n(n-1)$. Also notice that when $p = 1$ this dimension is $n-1$. From Theorems 1 and 2 we have the following

Corollary 1 $\dim(FCLOP(G_n, p)) = \frac{1}{2}n(n-1) + n - 1$, when $2 \leq p = n-1$, and $\dim(FCLOP(G_n, p)) = n(n-1) - 1$, when $2 \leq p \leq n-2$. \square

4.2 Facets

4.2.1 Trivial inequalities

In the following, we show that the two sets of trivial inequalities (4.5) and (4.6) are defining facets for $FCLOP(G_n, n-1)$. As well we show that (4.16) are defining facets for $FCLOP(G_n, p)$.

Theorem 3 *The trivial inequalities $x_{ij} \geq 0$ define facets for the $FCLOP(G_n, p)$, for any $n \geq 3$ and $p \leq n-1$.*

Proof *For simplicity we will show that*

$$x_{12} \geq 0. \quad (4.46)$$

Let $F = \{(x, y) \in \text{the } FCLOP(G_n, p) : x_{12} = 0\}$. Let

$$\alpha^T x + \beta^T y \leq \rho, \quad (4.47)$$

be a valid inequality defining a facet for $FCLOP(G_n, p)$ such that

$$F \subseteq F' = \{(x, y) \in \text{the } FCLOP(G_n, p) : \alpha^T x + \beta^T y = \rho\}.$$

It is straightforward to check that $\alpha_{ij} = \alpha_{ji}$ when $\{i, j\} \neq \{1, 2\}$. This will be used implicitly in the sequel. The proof is by induction. This why we will distinguish the cases (i) $p = n-1$, (ii) $p = n-2$ and (iii) $p \leq n-3$. Let $G' = (V_n, A'_n)$ be the graph obtained from G_n by removing the arcs $(1, 2)$ and $(2, 1)$.

(i) $p = n-1$. We will show that (4.47) may be obtained as a linear combination of (4.2), (4.3) and (4.46). Define $S_i = V_n \setminus \{i\}$. Let S_k , for $k \in V_n \setminus \{1, 2\}$ be any linear order such that $\pi_{S_k}(1) = 2$ and $\pi_{S_k}(2) = 1$. Let π_{S_1} and π_{S_2} any linear orders on S_1 and S_2 , respectively. Notice that the incidence vectors of these linear orders belong to F' . Therefore the following hold:

$$\beta_1 + \frac{1}{2}\alpha(\delta_{G'}(1)) = \beta_2 + \frac{1}{2}\alpha(\delta_{G'}(2)), \quad (4.48)$$

$$\beta_1 + \frac{1}{2}\alpha(\delta_{G'}(1)) + \alpha_{21} = \beta_k + \frac{1}{2}\alpha(\delta(k)) \quad \forall k \in V_n \setminus \{1, 2\}. \quad (4.49)$$

Consider the following linear combination:

$$\begin{aligned} \alpha_{ij} & \quad \times (x_{ij} + x_{ji} = y_i + y_j - 1) & \quad \forall i < j, j \notin \{1, 2\}, \\ \alpha_{21} & \quad \times (x_{12} + x_{21} = y_1 + y_2 - 1), \\ \alpha_{21} - \alpha_{12} & \quad \times (-x_{12} \leq 0). \end{aligned}$$

Their sum may be written as follows:

$$\begin{aligned} \alpha^T x - \frac{1}{2} \alpha(\delta(i)) \sum_{i \in V_n \setminus \{1, 2\}} y_i + (-\frac{1}{2} \alpha(\delta_{G'}(1)) - \alpha_{21}) y_1 \\ + (-\frac{1}{2} \alpha(\delta_{G'}(2)) - \alpha_{21}) y_2 \leq -\frac{1}{2} \alpha(A'_n) - \alpha_{21} \end{aligned}$$

Now considering (4.49) we obtain

$$\begin{aligned} \alpha^T x + \sum_{i \in V_n \setminus \{1, 2\}} \beta_i y_i + (-\beta_1 - \frac{1}{2} \alpha(\delta_{G'}(1)) - \alpha_{21}) \sum_{i \in V_n \setminus \{1, 2\}} y_i \\ + (-\frac{1}{2} \alpha(\delta_{G'}(1)) - \alpha_{21}) y_1 + (-\frac{1}{2} \alpha(\delta_{G'}(2)) - \alpha_{21}) y_2 \\ \leq -\frac{1}{2} \alpha(A'_n) - \alpha_{21} \end{aligned}$$

Recall that equality (4.3) implies that

$$\sum_{i \in V_n \setminus \{1, 2\}} y_i = (n - 1) - (y_1 + y_2)$$

and if we replace in the above inequality we get

$$\begin{aligned} \alpha^T x + \sum_{i \in V_n \setminus \{1, 2\}} \beta_i y_i - (-\beta_1 - \frac{1}{2} \alpha(\delta_{G'}(1)) - \alpha_{21})(y_1 + y_2) \\ + (-\frac{1}{2} \alpha(\delta_{G'}(1)) - \alpha_{21}) y_1 + (-\frac{1}{2} \alpha(\delta_{G'}(2)) - \alpha_{21}) y_2 \\ \leq -\frac{1}{2} \alpha(A'_n) - \alpha_{21} + (n - 1)(\beta_1 + \frac{1}{2} \alpha(\delta_{G'}(1)) + \alpha_{21}) \end{aligned}$$

Now using (4.48) and (4.49) it is easy to check that above inequality may be rewritten as below, for any $i \neq 1, 2$

$$\begin{aligned} \alpha^T x + \beta^T y & \leq -\frac{1}{2} \alpha(A'_n) - \alpha_{21} + (n - 2)(\beta_1 + \frac{1}{2} \alpha(\delta_{G'}(1)) + \alpha_{21}) \\ & \quad + (\beta_1 + \frac{1}{2} \alpha(\delta_{G'}(1)) + \alpha_{21}) \\ & = -\frac{1}{2} \alpha(A'_n) + \sum_{i \in V_n \setminus \{1, 2\}} (\beta_i + \frac{1}{2} \alpha(\delta(i))) + (\beta_1 + \frac{1}{2} \alpha(\delta_{G'}(1))) \end{aligned}$$

$$\begin{aligned}
&= -\frac{1}{2}\alpha(A'_n) + \sum_{i \in V_n \setminus \{1,2\}} \beta_i + \alpha(A'_n) - \frac{1}{2}\alpha(\delta_{G'}(1)) \\
&\quad - \frac{1}{2}\alpha(\delta_{G'}(2)) + (\beta_1 + \frac{1}{2}\alpha(\delta_{G'}(1))) \\
&= \frac{1}{2}\alpha(A'_n) + \sum_{i \in V_n \setminus \{1,2\}} \beta_i + \beta_1 - \frac{1}{2}\alpha(\delta_{G'}(2)).
\end{aligned}$$

Now let us show that

$$\rho = \frac{1}{2}\alpha(A'_n) + \sum_{i \in V_n \setminus \{1,2\}} \beta_i + \beta_1 - \frac{1}{2}\alpha(\delta_{G'}(2)).$$

We know that the incidence vector of the LO π_{S_k} for $k \notin \{1,2\}$, as defined above, is in F' . Therefore

$$\rho = \frac{1}{2}\alpha(A'_n) - \frac{1}{2}\alpha(\delta(k)) + \alpha_{21} + \beta_1 + \beta_2 - \beta_k + \sum_{i \in V_n \setminus \{1,2\}} \beta_i$$

Recall that from (4.48) and (4.49) we have

$$\beta_2 + \frac{1}{2}\alpha(\delta_{G'}(2)) + \alpha_{21} = \beta_k + \frac{1}{2}\alpha(\delta(k)) \forall k \in V_n \setminus \{1,2\}.$$

It follows that ρ is exactly the desired value.

- (ii) $p = n-2$. Let $l \in V_n$ and consider the FCLOP instance with the respect to the graph $G_{n-1}^l = (V_{n-1}^l, A_{n-1}^l)$ with the constant p unchanged that is $p = n-2$, where $V_{n-1}^l = V_n \setminus \{l\}$ and $A_{n-1}^l = \{(i,j) \in V_{n-1}^l \times V_{n-1}^l : i \neq j\}$. Any linear ordering π_S with respect to G_{n-1}^l and p is also a linear ordering with respect to G_n and p . Hence (4.46) define a facet for $FCLOP(G_{n-1}^l, p)$, with any $l \notin \{1,2\}$.

Also notice that any feasible solution of $FCLOP(G_{n-1}^l, p)$ that satisfies (4.46) with equality may be extended to a solution in $FCLOP(G_n, p)$ that also satisfies (4.46) as equation. This can be done by considering $y_l = 0$ and $x_{il} = x_{li} = 0$. . Hence we have the following

$$\begin{aligned}
&\{(x, y) \in \text{theFCLOP}(G_{n-1}^l, p) : x_{12} = 0\} \\
&\subseteq \\
&\{(x, y) \in \text{theFCLOP}(G_{n-1}^l, p) : \sum_{i,j \in V_{n-1}^l, i \neq j} \alpha_{ij}x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i = \rho\}.
\end{aligned}$$

Notice that (4.50) is valid for $FCLOP(G_{n-1}^l, p)$.

$$\sum_{i,j \in V_{n-1}^l, i \neq j} \alpha_{ij}x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i \leq \rho. \quad (4.50)$$

Since from case (i) inequality (4.46) define a facet for $FCLOP(G_{n-1}^l, p)$, then inequality (4.50) may be obtained as a linear combination of equalities (4.2), (4.3) and (4.46), all with respect to G_{n-1}^l and p . Consequently (4.50) is obtained as the sum of the following equalities and inequality:

$$\alpha_{ij} \times (x_{ij} + x_{ji} = y_i + y_j - 1) \quad \forall i < j, j \notin \{1, 2, l\}, i \neq l, \quad (4.51)$$

$$\alpha_{21} \times (x_{12} + x_{21} = y_1 + y_2 - 1), \quad (4.52)$$

$$\epsilon_l \times \left(\sum_{i \in V_n^l} y_i = p \right), \quad (4.53)$$

$$(\alpha_{21} - \alpha_{12}) \times (-x_{12} \leq 0), \quad (4.54)$$

where ϵ_l is a real scalar and σ is a positive scalar. The discussion above is true for any $l \in V_n \setminus \{1, 2\}$. Thus the following hold for any $l \in V_n \setminus \{1, 2\}$ and $i \notin \{1, 2, l\}$.

$$\phi_1 = \beta_1 + \frac{1}{2}\alpha(\delta_{G'}(1)) = -\alpha_{21} + \alpha_{1l} + \epsilon_l, \quad (4.55)$$

$$\phi_2 = \beta_2 + \frac{1}{2}\alpha(\delta_{G'}(2)) = -\alpha_{21} + \alpha_{2l} + \epsilon_l, \quad (4.56)$$

$$\phi_i = \beta_i + \frac{1}{2}\alpha(\delta(i)) = \alpha_{il} + \epsilon_l. \quad (4.57)$$

It results that

$$\alpha_{il} - \alpha_{ik} = \epsilon_k - \epsilon_l \quad \forall i \in V_n \text{ and } k, l \in V_n \setminus \{1, 2\}.$$

This implies that

$$\alpha_{ij} = \lambda_i - \epsilon_j \quad \forall i \in V_n \text{ and } j \in V_n \setminus \{1, 2\}, \quad (4.58)$$

where λ_i is a real scalar. Finally we may conclude that

$$\alpha_{ij} = \mu_i + \mu_j \quad \forall i, j \text{ with } \{i, j\} \neq \{1, 2\}, \quad (4.59)$$

where

$$\mu_i = \frac{\lambda_i - \epsilon_i}{2} \quad \forall i \in V_n \setminus \{1, 2\}, \quad (4.60)$$

$$\mu_i = \lambda_i - \frac{\lambda_l + \epsilon_l}{2} \quad i \in \{1, 2\}, \quad (4.61)$$

Let us show that $\mu_1 + \mu_2 = \alpha_{21}$. From (4.57) we have $\phi_k = \alpha_{kl} + \epsilon_l$ for any $k \in V_n \setminus \{1, 2, l\}$. But $\alpha_{kl} = \alpha_{lk}$ and $\alpha_{lk} = \lambda_l - \epsilon_k$, so

$$\lambda_l = \phi_k + \epsilon_k - \epsilon_l,$$

and since $\epsilon_k - \epsilon_l = \alpha_{il} - \alpha_{ik}$ for any $i \in V_n$ we may write

$$\lambda_l = \phi_k + \alpha_{il} - \alpha_{ik}, \quad \forall i \in V_n \text{ and } l, k \in V_n \setminus \{1, 2\}. \quad (4.62)$$

Notice that from (4.58), (4.61) and (4.62) we have

$$\mu_1 + \mu_2 = \alpha_{1l} + \alpha_{2l} + \epsilon_l - \lambda_l = \alpha_{2l} + \alpha_{1k} + \epsilon_l - \phi_k. \quad (4.63)$$

Let $S' = V_n \setminus \{1, 2\}$. Let $\pi_{S'}$ be any LO on S' . Let $S'' = V_n \setminus \{1, k\}$, with $k \neq 2$, and let $\pi_{S''}$ be any LO on S'' . Obviously both incidence vectors of $\pi_{S'}$ and $\pi_{S''}$ are in F and hence in F' too. This imply that

$$\phi_k - \alpha_{1k} = \phi_2.$$

Recall that $\phi_2 = -\alpha_{21} + \alpha_{2l} + \epsilon_l$. Therefore $\epsilon_l = \phi_k - \alpha_{1k} - \alpha_{2l} + \alpha_{21}$. If we replace in (4.63) we obtain

$$\mu_1 + \mu_2 = \alpha_{21}. \quad (4.64)$$

Since (4.50) is the sum of the equalities (4.51)-(4.54) we have that

$$\rho = -\frac{1}{2}\alpha(A'_n) + \frac{1}{2}\alpha(\delta(l)) + \epsilon_l p - \alpha_{21}. \quad (4.65)$$

From (4.57) we have $\epsilon_l = \beta_i + \frac{1}{2}\alpha(\delta(i)) - \alpha_{il}$. A careful combination by replacing this value of ϵ_l in (4.65) and using (4.59) and (4.64) gives

$$\beta_i = -(p-1)\mu_i + \frac{\rho}{p} \quad (4.66)$$

Using exactly the same combinations by considering the value of ϵ_l in with respect to (4.55) and (4.56) we obtain

$$\beta_1 = -(p-1)\mu_1 + \frac{\rho}{p}, \quad (4.67)$$

$$\beta_2 = -(p-1)\mu_2 + \frac{\rho}{p}. \quad (4.68)$$

Now using (4.66)-(4.68) it is easy to check that (4.47) is obtained as the sum of the following constraints

$$\begin{aligned} \mu_i \times (x(\delta(i)) &= (p-1)y_i) & \forall i \in V_n, \\ \frac{\rho}{p} \times (\sum_{i \in V_n} y_i &= p), \\ (\alpha_{21} - \alpha_{12}) \times (-x_{12} &\leq 0). \end{aligned}$$

(iii) $p \leq n - 3$. This case will be proved by induction on the size of V_n . Notice that here we must have $n \geq 5$. The basic case of the induction is obtained when $p = n - 2$, which is true from the previous case (i). By the induction hypothesis the theorem is true for any instance of the FCLOP, $G_{n'} = (V_{n'}, A_{n'})$ with $2 \leq p \leq n - 3$ and $p + 2 \leq n' < n$. Therefore we may assume that the theorem is true for $G_{n-1}^l = (V_{n-1}^l, A_{n-1}^l)$ and p with $l \notin \{1, 2\}$. Here $p \leq (n - 1) - 2$. As before $V_{n-1}^l = V_n \setminus \{l\}$ and $A_{n-1}^l = \{(i, j) \in V_{n-1}^1 \times V_{n-1}^l : i \neq j\}$.

Any feasible solution of $FCLOP(G_{n-1}^l, p)$ that satisfies (4.46) with equality may be extended to a solution in $FCLOP(G_n, p)$ that also satisfies (4.46) as equation. This can be done by considering $y_l = 0$ and $x_{li} = x_{il} = 0$. Hence we have the following

$$\begin{aligned} & \{(x, y) \in \text{theFCLOP}(G_{n-1}^l, p) : x_{12} = 0\} \\ & \subseteq \\ & \{(x, y) \in \text{theFCLOP}(G_{n-1}^l, p) : \sum_{i,j \in V_{n-1}^l, i \neq j} \alpha_{ij} x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i = \rho\}. \end{aligned}$$

Notice that (4.69) is valid for $FCLOP(G_{n-1}^l, p)$.

$$\sum_{i,j \in V_{n-1}^l, i \neq j} \alpha_{ij} x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i \leq \rho. \quad (4.69)$$

By the induction hypothesis the trivial inequality (4.46) define a facet for $FCLOP(G_{n-1}^l, p)$. Therefore inequality (4.69) may be obtained by the sum of the following constraints:

$$\begin{aligned} & \mu_i \times (x(\delta(i)) = (p - 1)y_i) \quad \text{for all } i \in V_n \setminus \{l\}, \\ & \epsilon_l \times \left(\sum_{i \in V_n \setminus \{l\}} y_i = p \right), \\ & \sigma \times (-x_{12} \leq 0). \end{aligned}$$

These imply the following

$$\mu_i + \mu_j = \alpha_{ij} \quad \forall i, j \in V_n \setminus \{l\}, \{i, j\} \neq \{1, 2\}, \quad (4.70)$$

$$\mu_1 + \mu_2 = \alpha_{21}, \quad (4.71)$$

$$\mu_1 + \mu_2 - \sigma = \alpha_{12}, \quad (4.72)$$

$$\beta_i = -(p - 1)\mu_i + \epsilon_l \quad \forall i \in V_n \setminus \{l\}, \quad (4.73)$$

$$\rho = \epsilon_l p. \quad (4.74)$$

Let $S \subseteq V_n \setminus \{l\}$, with $|S| = p$. Let π_S be any linear order with (x^{π_S}, y^{π_S}) in $F \subseteq F'$. If $S \cap \{1, 2\} = \{1\}$, then set $\pi_S^{-1}(1) = |S|$; and if $S \cap \{1, 2\} =$

$\{2\}$, then set $\pi_S^{-1}(2) = 1$. Let $S' = (S \setminus \{k\}) \cup \{l\}$. From π_S define the following linear order $\pi_{S'}'$: $\pi_{S'}'^{-1}(l) = \pi_S^{-1}(k)$ and $\pi_{S'}'^{-1}(j) = \pi_S^{-1}(j)$ for each $j \in S \setminus \{k\}$. Notice that from the choice of π_S , we have that $(x^{\pi_{S'}'}, y^{\pi_{S'}'}) \in F'$ for any $k \in S$. Notice that for any $k \neq l$, there are two sets S and S' as defined above. When we replace (x^{π_S}, y^{π_S}) and $(x^{\pi_{S'}'}, y^{\pi_{S'}'})$ in F' for any k and we combine the two resulting equations we obtain:

$$\beta_k + \alpha(\delta_{G(S)}^+(k)) + \alpha(\delta_{G(S)}^-(k)) = \beta_l + \sum_{j \in S \setminus \{k\}} \alpha_{lj}. \quad (4.75)$$

We have $\beta_k + \alpha(\delta_{G(S)}^+(k)) + \alpha(\delta_{G(S)}^-(k)) = \beta_k + (p-1)\mu_k + \sum_{j \in S \setminus \{k\}} \mu_j$. If we replace β_k by its value in (4.73) and we combine with (4.75), we get

$$\epsilon_l + \sum_{j \in S} \mu_j - \mu_k = \beta_l + \sum_{j \in S} \alpha_{lj} - \alpha_{lk}. \quad (4.76)$$

If we write (4.76) with respect to another element $k' \in S$, $k \neq k'$, and we combine the resulting equality with (4.76) we get

$$\alpha_{lk} - \alpha_{lk'} = \mu_k - \mu_{k'} \quad \forall k, k' \in S \setminus \{l\}.$$

Notice that such k' exists always by the definition of π_S and $\pi_{S'}$ and since $p \geq 2$. Also for any pair of nodes k and k' different from l , we can build a subset S , S' and the linear orderings π_S and $\pi_{S'}$ as defined above. Therefore

$$\alpha_{lk} - \alpha_{lk'} = \mu_k - \mu_{k'} \quad \forall k, k' \in V_n \setminus \{l\}.$$

From this last equation and the fact that $\alpha_{lj} = \alpha_{jl}$ we may assume that

$$\begin{aligned} \alpha_{lj} &= \mu_l + \mu_j && \text{for any } j \in V_n \setminus \{l\}, \\ \alpha_{jl} &= \mu_l + \mu_j && \text{for any } j \in V_n \setminus \{l\}. \end{aligned}$$

If we replace with the value of α_{lj} in (4.76), we obtain

$$\beta_l = -(p-1)\mu_l + \epsilon_l$$

Now it is straightforward to see that (4.47) is obtained by the sum of the following constraints:

$$\begin{aligned} \mu_i \times (x(\delta(i)) &= (p-1)y_i) && \text{for all } i \in V_n, \\ \epsilon_l \times \left(\sum_{i \in V_n \setminus \{l\}} y_i \right) &= p, \\ \sigma \times (-x_{12} &\leq 0). \end{aligned}$$

□

Theorem 4 *The trivial inequalities $y_i \leq 1$, $\forall i \in V_n$ are defining facets for $FCLOP(G_n, n-1)$.*

Proof *For simplicity we will show that*

$$y_1 \leq 1. \quad (4.77)$$

define a facet for $FCLOP(G_n, n-1)$.

Let $F = \{(x, y) \in FCLOP(G_n, p) : y_1 = 1\}$. Let

$$\alpha^T x + \beta^T y \leq \rho, \quad (4.78)$$

be a valid inequality defining a facet for $FCLOP(G_n, p)$ such that $F \subseteq F' = \{(x, y) \in FCLOP(G_n, p) : \alpha^T x + \beta^T y = \rho\}$.

It is straightforward to check that $\alpha_{ij} = \alpha_{ji}$ for all $i, j \in V_n$. This will be used implicitly in the sequel.

Let $p = n-1$. We will show that (4.78) may be obtained as a linear combination of (4.2), (4.3) and (4.77). Define $S_i = V_n \setminus \{i\}$. Notice that the incidence vector of the linear order π_{S_i} belongs to F' if $i \neq 1$. Therefore the following hold:

$$\beta_2 + \frac{1}{2}\alpha(\delta(2)) = \beta_k + \frac{1}{2}\alpha(\delta(k)), \quad \forall k \in V_n \setminus \{1\}. \quad (4.79)$$

Let us define ϕ_k and ϕ_1 as follows:

$$\begin{aligned} \phi_k &= \beta_k + \frac{1}{2}\alpha(\delta(k)), & \forall k \in V_n \setminus \{1\}, \\ \phi_1 &= \beta_1 + \frac{1}{2}\alpha(\delta(1)). \end{aligned}$$

Then we have $\phi_i = \phi_j$ $\forall i, j \in V_n \setminus \{1\}$. Let $\sigma = \beta_2 + \frac{1}{2}\alpha(\delta(2))$ and $\sigma^1 = \beta_1 + \frac{1}{2}\alpha(\delta(1))$. Consider the following linear combination:

$$\alpha_{ij} \times (x_{ij} + x_{ji} = y_i + y_j - 1) \quad \forall i, j \in V_n, \quad (4.80)$$

$$\sigma \times \left(\sum_{i \in V_n} y_i = (n-1) \right), \quad (4.81)$$

$$(\sigma^1 - \sigma) \times (y_1 \leq 1). \quad (4.82)$$

Their sum may be written as follows:

$$\begin{aligned} &\alpha^T x - \sum_{i \in V_n \setminus \{1\}} \frac{1}{2}\alpha(\delta(i))y_i - \frac{1}{2}\alpha(\delta(1))y_1 + \sigma \sum_{i \in V_n \setminus \{1\}} y_i \\ &+ \sigma y_1 + \sigma^1 y_1 - \sigma y_1 \leq -\frac{1}{2}\alpha(A_n) + \sigma(n-1) + \sigma^1 - \sigma \end{aligned}$$

Now considering (4.79) we obtain

$$\alpha^T x + \beta^T y \leq \left[-\frac{1}{2}\alpha(A_n) + (n-2)\sigma + \beta_1 + \frac{1}{2}\alpha(\delta(1)) \right]$$

Now let us show that

$$\rho = -\frac{1}{2}\alpha(A_n) + (n-2)\sigma + \sigma^1. \quad (4.83)$$

We know that the incidence vector of the linear order π_{S_k} for $k \notin \{1\}$, as defined above, is in F' . Therefore

$$\rho = \frac{1}{2}\alpha(A_n) - \frac{1}{2}\alpha(\delta(k)) - \beta_k + \sum_{i \in V_n} \beta_i \quad (4.84)$$

Recall that from (4.79) we have

$$\sigma = \beta_k + \frac{1}{2}\alpha(\delta(k)) \quad \forall k \in V_n \setminus \{1\}.$$

Therefor we may write

$$(n-2)\sigma = \sum_{i \in V_n \setminus \{1, k\}} (\beta_i + \frac{1}{2}\alpha(\delta(i))) \quad \forall k \in V_n \setminus \{1\} \quad (4.85)$$

We also know that

$$\alpha(A_n) = \frac{1}{2}\alpha(\delta(k)) + \frac{1}{2} \sum_{i \in V_n \setminus \{k\}} \alpha(\delta(i))$$

Therefor, we may write

$$\frac{1}{2}\alpha(A_n) - \frac{1}{2}\alpha(\delta(k)) = -\frac{1}{2}\alpha(A_n) + \frac{1}{2} \sum_{i \in V_n \setminus \{k\}} \alpha(\delta(i)) \quad (4.86)$$

By substituting (4.86) in (4.84) and combining the resulting equation in (4.85) we obtain (4.84). \square

Remark 4 The inequality $y_1 \leq 1$ does not define facet for the $FCLOP(G_n, p)$ with $p \leq n-2$ since it is implied by $x_{12} + x_{21} \leq y_1$, $x_{12} + x_{21} \leq y_2$ and $x_{12} + x_{21} \geq y_1 + y_2 - 1$ and that the face define by $y_1 = 1$ is completely contained in the face defined by $x_{12} + x_{21} \leq y_1$. As a consequence, $y_i \leq 1$, $\forall i \in V_n$ is not defining facets for the $FCLOP$ polytope when $p \leq n-2$.

4.2.2 Triangle-free inequalities

Theorem 5 *Inequalities (4.15) define facets for the $FCLOP(G_n, p)$, for $p = n - 1$ and $p = n - 2$ with $p \geq 3$.*

Proof When $p = n$, $FCLOP(G_n, p)$ is the linear ordering polytope and inequalities (4.15) are exactly the triangle-free inequalities and the result follows from Grötschel et al. (1985a). Without loss of generality, we may assume that inequality (4.15) is considered with respect to the arcs $(1, 2)$, $(2, 3)$ and $(3, 1)$ and hence is written as follows

$$x_{12} + x_{23} + x_{31} + y(V_n \setminus \{1, 2, 3\}) \leq \lfloor \frac{2p + (n - 3)}{3} \rfloor. \quad (4.87)$$

Let F be the face of $FCLOP(G_n, p)$ defined by (4.87), that is $F = \{(x, y) \in \text{the } FCLOP(G_n, p) : x_{12} + x_{23} + x_{31} + y(V_n \setminus \{1, 2, 3\}) = \lfloor \frac{2p + (n - 3)}{3} \rfloor\}$. Let

$$\alpha^T x + \beta^T y \leq \rho, \quad (4.88)$$

be a facet of $FCLOP(G_n, p)$ and let $F' = \{(x, y) \in \text{the } FCLOP(G_n, p) : \alpha^T x + \beta^T y = \rho\}$. Assume that $F \subseteq F'$. To complete the proof we need to show that (4.88) is a linear combination of (4.87), (4.2) and (4.3) in case where $p = n - 1$. When $p \leq n - 2$, we need to show that (4.88) is a linear combination of (4.87), (4.9) and (4.10).

We will consider three cases (i) $p = n - 1$ and (ii) $p = n - 2$. Let us before state an important property with some notation useful for the three cases. Notice that if we take any LO having its incidence vector in $F \subseteq F'$, then if we switch the order of two any elements i and j , $i, j \notin \{1, 2, 3\}$, then the resulting LO still in F' consequently we have $\alpha_{ij} = \alpha_{ji}$. The same is true if we consider that exactly one of the two elements i and j is in $\{1, 2, 3\}$. There is always a linear ordering satisfying (4.87) as equation and if we switch an element $i \notin \{1, 2, 3\}$ with an element $j \in \{1, 2, 3\}$, the resulting LO satisfies again (4.87) with equality. We summarize this discussion in the following remark.

Remark 5 *We have $\alpha_{ij} = \alpha_{ji}$ for any $i, j \in V_n$ with $\{i, j\} \not\subseteq \{1, 2, 3\}$.*

For each $i \in \{1, 2, 3\}$, denote by G^i the graph obtained by removing the arcs (i, j) , (j, i) for each $j \in \{1, 2, 3\} \setminus \{i\}$.

Let $A'_n = A_n \setminus \{(1, 2), (2, 1), (2, 3), (3, 2), (1, 3), (3, 1)\}$.

- (i) $p = n - 1$. Notice that the right hand side of (4.87) is equal to $n - 2$ in this case. Let $S_i = V_n \setminus \{i\}$. Assume that $i \notin \{1, 2, 3\}$, and let $\pi_{S_i}^1$ be any LO on S_i such that $\pi_{S_i}^1(j) = j$, for $j \in \{1, 2, 3\}$. Now define

the LO $\pi_{S_1}^1$ which is the same as $\pi_{S_i}^1$ except that $\pi_{S_1}^1(1) = i$. It is a simple matter to check that the incidence vectors of both $\pi_{S_i}^1$ and $\pi_{S_1}^1$ belong to F and thus to F' . Consequently

$$\begin{aligned}\alpha^T x^{\pi_{S_i}^1} + \beta^T y^{\pi_{S_i}^1} &= \rho, \\ \alpha^T x^{\pi_{S_1}^1} + \beta^T y^{\pi_{S_1}^1} &= \rho.\end{aligned}$$

The combination of these equalities implies that

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \beta_1 + \frac{1}{2}\alpha(\delta_{G^1}(1)) + \alpha_{12} + \alpha_{13}, \quad (4.89)$$

Apply the same procedure to the LO $\pi_{S_i}^2$ and $\pi_{S_1}^2$, where $\pi_{S_i}^2$ is any LO on S_i such that $\pi_{S_i}^2(1) = 2$, $\pi_{S_i}^2(2) = 3$ and $\pi_{S_i}^2(3) = 1$ and $\pi_{S_1}^2$ is the same as $\pi_{S_i}^2$ except that $\pi_{S_1}^2(3) = i$, we get

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \beta_1 + \frac{1}{2}\alpha(\delta_{G^1}(1)) + \alpha_{21} + \alpha_{31}. \quad (4.90)$$

Using the same reasoning as above with respect to some appropriate LO we get the following

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \beta_2 + \frac{1}{2}\alpha(\delta_{G^2}(2)) + \alpha_{23} + \alpha_{21}, \quad (4.91)$$

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \beta_2 + \frac{1}{2}\alpha(\delta_{G^2}(2)) + \alpha_{32} + \alpha_{12}, \quad (4.92)$$

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \beta_3 + \frac{1}{2}\alpha(\delta_{G^3}(3)) + \alpha_{31} + \alpha_{32}, \quad (4.93)$$

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \beta_3 + \frac{1}{2}\alpha(\delta_{G^3}(3)) + \alpha_{13} + \alpha_{23}, \quad (4.94)$$

The combination of (4.89) and (4.94) gives

$$\begin{aligned}\alpha_{21} - \alpha_{12} &= \alpha_{13} - \alpha_{31} = \sigma, \\ \alpha_{13} - \alpha_{31} &= \alpha_{32} - \alpha_{23} = \sigma,\end{aligned}$$

Therefore we may assume that there exists scalars λ_{12} , λ_{13} and λ_{23} such that,

$$\alpha_{21} = \lambda_{12} + \sigma, \quad (4.95)$$

$$\alpha_{12} = \lambda_{12}, \quad (4.96)$$

$$\alpha_{13} = \lambda_{13} + \sigma, \quad (4.97)$$

$$\alpha_{31} = \lambda_{13}, \quad (4.98)$$

$$\alpha_{32} = \lambda_{23} + \sigma, \quad (4.99)$$

$$\alpha_{23} = \lambda_{23}. \quad (4.100)$$

From (4.89)-(4.94) we may write

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \phi = \beta_1 + \frac{1}{2}\alpha(\delta_{G^1}(1)) + \alpha_{12} + \alpha_{13} \quad (4.101)$$

$$= \beta_2 + \frac{1}{2}\alpha(\delta_{G^2}(2)) + \alpha_{23} + \alpha_{21}, \quad (4.102)$$

$$= \beta_3 + \frac{1}{2}\alpha(\delta_{G^3}(3)) + \alpha_{31} + \alpha_{32}, \quad (4.103)$$

Let us show that $\sigma \leq 0$. Recall that $\pi_{S_i}^1$ is a LO on S_i for $i \notin \{1, 2, 3\}$, where $\pi_{S_i}^1(j) = j$, for $j \in \{1, 2, 3\}$. Let π_{S_i} the same as $\pi_{S_i}^1$, except that $\pi_{S_i}(1) = 3$ $\pi_{S_i}(2) = 2$ $\pi_{S_i}(3) = 1$. It is easy to check that the incidence vector of $\pi_{S_i}^1$ belongs to F' whereas the incidence vector of π_{S_i} does not but it must satisfy (4.88) since this is a valid inequality. Consequently,

$$\begin{aligned} \alpha^T x^{\pi_{S_i}^1} + \beta^T y^{\pi_{S_i}^1} &= \alpha_{12} + \alpha_{23} + \alpha_{13} + \xi = \rho, \\ \alpha^T x^{\pi_{S_i}} + \beta^T y^{\pi_{S_i}} &= \alpha_{31} + \alpha_{32} + \alpha_{21} + \xi \leq \rho. \end{aligned}$$

The combination of the two constraints above implies that

$$\alpha_{32} - \alpha_{23} + \alpha_{31} - \alpha_{13} + \alpha_{21} - \alpha_{12} = \sigma - \sigma + \sigma \leq 0.$$

Consider the following linear combination:

$$\alpha_{ij} \times (x_{ij} + x_{ji} = y_i + y_j - 1) \quad \forall i < j, \{i, j\} \not\subset \{1, 2, 3\}, \quad (4.104)$$

$$(\lambda_{12} + \sigma) \times (x_{12} + x_{21} = y_1 + y_2 - 1), \quad (4.105)$$

$$(\lambda_{13} + \sigma) \times (x_{13} + x_{31} = y_1 + y_3 - 1), \quad (4.106)$$

$$(\lambda_{23} + \sigma) \times (x_{23} + x_{32} = y_2 + y_3 - 1), \quad (4.107)$$

$$(\sigma + \phi) \times \left(\sum_{i \in V_n} y_i = n - 1 \right), \quad (4.108)$$

$$-\sigma \times (x_{12} + x_{23} + x_{31} + \sum_{i \in V_n \setminus \{1, 2, 3\}} y_i \leq n - 2). \quad (4.109)$$

The sum of (4.104)-(4.109) is the following

$$\begin{aligned} &\alpha^T x + \left(-\frac{1}{2}\alpha(\delta(i)) + \phi\right) + \sum_{i \in V_n \setminus \{1, 2, 3\}} y_i + (-\lambda_{12} - \lambda_{13} - \sigma + \phi - \frac{1}{2}\alpha(\delta_{G^1}(1)))y_1 \\ &+ (-\lambda_{12} - \lambda_{23} - \sigma + \phi - \frac{1}{2}\alpha(\delta_{G^2}(2)))y_2 + (-\lambda_{13} - \lambda_{23} - \sigma + \phi - \frac{1}{2}\alpha(\delta_{G^3}(3)))y_3 \end{aligned}$$

$$\leq -\frac{1}{2}\alpha(A'_n) - \lambda_{12} - \lambda_{13} - \lambda_{23} - 3\sigma + (\sigma + \phi)(n-1) - \sigma(n-2),$$

By choosing the appropriate value of ϕ using (4.101)-(4.103) and replacing in the above inequality, we get

$$\begin{aligned} \alpha^T x + \sum_{i \in V_n \setminus \{1,2,3\}} \beta_i y_i &+ (-\lambda_{12} - \lambda_{13} - \sigma + \beta_1 + \alpha_{12} + \alpha_{13})y_1 \\ &+ (-\lambda_{12} - \lambda_{23} - \sigma + \beta_2 + \alpha_{23} + \alpha_{21})y_2 \\ &+ (-\lambda_{13} - \lambda_{23} - \sigma + \beta_3 + \alpha_{31} + \alpha_{32})y_3 \\ &\leq -\frac{1}{2}\alpha(A'_n) - \lambda_{12} - \lambda_{13} - \lambda_{23} - 3\sigma + (\sigma + \phi)(n-1) - \sigma(n-2) \end{aligned}$$

Finally by considering (4.95)-(4.100) we get

$$\alpha^T x + \beta^T y \leq -\frac{1}{2}\alpha(A'_n) - \sigma - \alpha_{12} - \alpha_{23} - \alpha_{13} + \phi(n-1). \quad (4.110)$$

The last step is to show that the right hand side of (4.110) is ρ .

Consider the LO $\pi_{S_i}^1$ as defined above, its incidence vector belong to F' . Hence

$$\rho = \frac{1}{2}\alpha(A'_n) - \frac{1}{2}\alpha(\delta(i)) + \alpha_{12} + \alpha_{23} + \alpha_{13} + \sum_{j=1, j \neq i}^n \beta_j,$$

Replace $\frac{1}{2}\alpha(\delta(i))$ by its value in (4.55), we get

$$\rho = \frac{1}{2}\alpha(A'_n) + \alpha_{12} + \alpha_{23} + \alpha_{13} - \phi + \sum_{j=1}^n \beta_j,$$

Replace all the β_i for $i = 1, \dots, n$, by their values in (4.101)-(4.103), we then obtain

$$\begin{aligned} \rho &= \frac{1}{2}\alpha(A'_n) + \alpha_{12} + \alpha_{23} + \alpha_{13} - \phi + \phi n - \frac{1}{2} \sum_{i \in V_n \setminus \{1,2,3\}} \alpha(\delta(i)) \\ &\quad - \frac{1}{2}\alpha(\delta_{G^1}(1)) - \frac{1}{2}\alpha(\delta_{G^2}(2)) - \frac{1}{2}\alpha(\delta_{G^3}(3)) \\ &\quad - \alpha_{12} - \alpha_{13} - \alpha_{23} - \alpha_{21} - \alpha_{31} - \alpha_{32} \end{aligned}$$

A careful sum with the relationships (4.95)-(4.100) imply

$$\begin{aligned} \rho &= \frac{1}{2}\alpha(A'_n) + \alpha_{12} + \alpha_{23} + \alpha_{13} + \phi(n-1) - \alpha(A'_n) \\ &\quad - 2(\alpha_{12} + \alpha_{23} + \alpha_{13}) - \sigma, \\ &= -\frac{1}{2}\alpha(A'_n) - \sigma - \alpha_{12} - \alpha_{23} - \alpha_{13} + \phi(n-1). \end{aligned}$$

(ii) $p = n - 2$. Notice that the right hand side is equal to $n - 3$. Let $l \in V_n$ and consider the FCLOP instance with respect to the graph $G_{n-1}^l = (V_{n-1}^l, A_{n-1}^l)$ with the constant p unchanged that is $p = n - 2$. Here $V_{n-1}^l = V_n \setminus \{l\}$ and $A_{n-1}^l = \{(i, j) \in V_{n-1}^l \times V_{n-1}^l : i \neq j\}$. Any linear ordering π_S with respect to G_{n-1}^l and p is also a linear ordering with respect to G_n and p . The following inequality obtained from (4.87) is a facet of $FCLOP(G_{n-1}^l, p)$, this is the result of the previous case

$$x_{12} + x_{23} + x_{31} + y(V_n \setminus \{1, 2, 3, l\}) \leq \lfloor \frac{2p + ((n-1) - 3)}{3} \rfloor = n - 3. \quad (4.111)$$

Also notice that any feasible solution of $FCLOP(G_{n-1}^l, p)$ that satisfies (4.111) with equality may be extended to a solution in $FCLOP(G_n, p)$ by considering $y_l = 0$ and $x_{il} = x_{li} = 0$. This solution belongs to F and thus it belongs to F' too. Hence we have the following

$$\begin{aligned} & \{(x, y) \in \text{the } FCLOP(G_{n-1}^l, p) : x_{12} + x_{23} + x_{31} + y(V_n \setminus \{1, 2, 3, l\}) = n - 3\} \\ & \subseteq \\ & \{(x, y) \in \text{the } FCLOP(G_{n-1}^l, p) : \sum_{i, j \in V_{n-1}^l, i \neq j} \alpha_{ij} x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i = \rho\}. \end{aligned}$$

It is easy to see that the inequality

$$\sum_{i, j \in V_{n-1}^l, i \neq j} \alpha_{ij} x_{ij} + \sum_{i \in V_{n-1}^l} \beta_i y_i \leq \rho, \quad (4.112)$$

is valid for $FCLOP(G_{n-1}^l, p)$. Since from case (i), inequality (4.111) define a facet of $FCLOP(G_{n-1}^l, p)$, then inequality (4.112) define the same facet and may be obtained as a linear combination of equalities (4.2), (4.3) and inequality (4.111), all with respect to G_{n-1}^l and p . Recall that when i and j do not belong at the same time to $\{1, 2, 3\}$, we have $\alpha_{ij} = \alpha_{ji}$. Consequently (4.112) is obtained as the sum of the following equalities and inequality:

$$\alpha_{ij} \times (x_{ij} + x_{ji} = y_i + y_j - 1) \quad \forall i < j, j \notin \{1, 2, 3, l\}, i \neq l, \quad (4.113)$$

$$\alpha_{21} \times (x_{12} + x_{21} = y_1 + y_2 - 1), \quad (4.114)$$

$$\alpha_{13} \times (x_{13} + x_{31} = y_1 + y_3 - 1), \quad (4.115)$$

$$\alpha_{32} \times (x_{23} + x_{32} = y_2 + y_3 - 1), \quad (4.116)$$

$$\epsilon_l \times \left(\sum_{i \in V_n \setminus \{l\}} y_i = p \right), \quad (4.117)$$

$$\sigma \times (x_{12} + x_{23} + x_{31} + \sum_{i \in V_n \setminus \{1,2,3,l\}} y_i \leq n - 3), \quad (4.118)$$

Where ϵ_l is a real scalar and σ is a positive scalar. The discussion above is true for any $l \in V_n \setminus \{1, 2, 3\}$.

Thus the following hold for any $l \in V_n \setminus \{1, 2, 3\}$.

$$\alpha_{12} - \alpha_{21} = \alpha_{31} - \alpha_{13} = \alpha_{23} - \alpha_{32} = \sigma, \quad (4.119)$$

$$\beta_i + \frac{1}{2}\alpha(\delta(i)) = \phi_i = \alpha_{il} + \epsilon_l + \sigma \quad \forall i \in V_n \setminus \{1, 2, 3, l\}, \quad (4.120)$$

$$\beta_1 + \frac{1}{2}\alpha(\delta_{G^1}(1)) = \phi_1 = \alpha_{1l} - \alpha_{21} - \alpha_{13} + \epsilon_l, \quad (4.121)$$

$$\beta_2 + \frac{1}{2}\alpha(\delta_{G^2}(2)) = \phi_2 = \alpha_{2l} - \alpha_{21} - \alpha_{32} + \epsilon_l, \quad (4.122)$$

$$\beta_3 + \frac{1}{2}\alpha(\delta_{G^3}(3)) = \phi_3 = \alpha_{3l} - \alpha_{13} - \alpha_{32} + \epsilon_l. \quad (4.123)$$

If we combine each of the equalities (4.120)-(4.123) with respect to l and k and both l and k are not in $\{1, 2, 3\}$ we get

$$\alpha_{il} - \alpha_{ik} = \epsilon_k - \epsilon_l \quad \forall i \in V_n \text{ and } k, l \in V_n \setminus \{1, 2, 3\}. \quad (4.124)$$

Following the above equality we may assume that

$$\alpha_{ij} = \lambda_i - \epsilon_j \quad \forall i \in V_n \text{ and } j \in V_n \setminus \{1, 2, 3\}, \quad (4.125)$$

where λ_i is a real scalar.

Following the same lines as in (4.37)-(4.38) we may assume that

$$\alpha_{ij} = \mu_i + \mu_j \quad \forall i, j \text{ with } \{i, j\} \not\subset \{1, 2, 3\}, \quad (4.126)$$

with

$$\mu_i = \frac{\lambda_i - \epsilon_i}{2} \quad \forall i \in V_n \setminus \{1, 2, 3\}, \quad (4.127)$$

$$\mu_i = \lambda_i - \frac{\lambda_l + \epsilon_l}{2} \quad \forall i \in \{1, 2, 3\}, \quad (4.128)$$

where l is any element among $V_n \setminus \{1, 2, 3\}$ this is coherent since $\lambda_i + \epsilon_i = \lambda_j + \epsilon_j$ for any $i, j \in V_n \setminus \{1, 2, 3\}$. This come from (4.125) and the fact that in this case $\alpha_{ij} = \alpha_{ji}$.

Let us show that $\mu_1 + \mu_3 = \alpha_{13}$, $\mu_1 + \mu_2 = \alpha_{21}$ and $\mu_2 + \mu_3 = \alpha_{32}$. Before we began the proof for each case let us state some facts.

If we consider (4.120) with respect to some $k \in V_n \setminus \{1, 2, 3, l\}$ instead of the element i we get $\phi_k = \alpha_{kl} + \epsilon_l + \sigma$. But since $\alpha_{kl} = \alpha_{lk}$ and by (4.125) $\alpha_{lk} = \lambda_l - \epsilon_k$ we get

$$\lambda_l = \phi_k + \epsilon_k - \epsilon_l - \sigma,$$

and since $\epsilon_k - \epsilon_l = \alpha_{il} - \alpha_{ik}$ for any $i \in V_n$ we may write

$$\lambda_l = \phi_k + \alpha_{il} - \alpha_{ik} - \sigma, \quad \forall i \in V_n \text{ and } l, k \in V_n \setminus \{1, 2, 3\}. \quad (4.129)$$

Notice that from (4.125), (4.128) and (4.129) we have

$$\mu_1 + \mu_3 = \alpha_{1l} + \alpha_{3l} + \epsilon_l - \lambda_l = \alpha_{3l} + \alpha_{1k} + \sigma + \epsilon_l - \phi_k, \quad (4.130)$$

$$\mu_1 + \mu_2 = \alpha_{1l} + \alpha_{2l} + \epsilon_l - \lambda_l = \alpha_{2l} + \alpha_{1k} + \sigma + \epsilon_l - \phi_k, \quad (4.131)$$

$$\mu_2 + \mu_3 = \alpha_{2l} + \alpha_{3l} + \epsilon_l - \lambda_l = \alpha_{3l} + \alpha_{2k} + \sigma + \epsilon_l - \phi_k. \quad (4.132)$$

- Let $S' = V_n \setminus \{1, 3\}$. Let $\pi_{S'}$ be any LO on S' . Let $S'' = V_n \setminus \{1, k\}$ and $\pi_{S''}$ be any LO on S'' with $\pi_{S''}(1) = 2$ and $\pi_{S''}(2) = 3$. It is easy to check that both incidence vectors of $\pi_{S'}$ and $\pi_{S''}$ are in F and hence in F' too. This imply that

$$\phi_k - \alpha_{1k} = \phi_3 + \alpha_{23}$$

Recall from (4.123) that $\phi_3 = \alpha_{3l} - \alpha_{13} - \alpha_{32} + \epsilon_l$. Therefore $\epsilon_l = \phi_k - \alpha_{1k} - \alpha_{23} - \alpha_{3l} + \alpha_{13} + \alpha_{32}$. If we replace in (4.130) we obtain $\mu_1 + \mu_3 = \sigma + \alpha_{32} - \alpha_{23} + \alpha_{13}$. But recall from (4.119) we have $\alpha_{23} - \alpha_{32} = \sigma$, consequently

$$\mu_1 + \mu_3 = \alpha_{13}. \quad (4.133)$$

- Define $S' = V_n \setminus \{1, 2\}$. Let $\pi_{S'}$ be any LO on S' . Let $S'' = V_n \setminus \{1, k\}$ and $\pi_{S''}$ be any LO on S'' with $\pi_{S''}(1) = 2$ and $\pi_{S''}(2) = 3$. It is easy to check that both incidence vectors of $\pi_{S'}$ and $\pi_{S''}$ are in F and hence in F' too. This imply that

$$\phi_k - \alpha_{1k} = \phi_2 + \alpha_{23}.$$

Combining this equality with the value of ϕ_2 in (4.122) we obtain

$$\epsilon_l = \phi_k - \alpha_{1k} - \alpha_{23} - \alpha_{2l} + \alpha_{21} + \alpha_{32}.$$

Now if we replace ϵ_l in (4.131) by its value in the above equality we obtain

$$\mu_1 + \mu_2 = \alpha_{21}. \quad (4.134)$$

- Define $S' = V_n \setminus \{2, 3\}$. Let $\pi_{S'}$ be any LO on S' . Let $S'' = V_n \setminus \{2, k\}$ and $\pi_{S''}$ be any LO on S'' with $\pi_{S''}(1) = 3$ and $\pi_{S''}(2) = 1$. It is easy to check that both incidence vectors of $\pi_{S'}$ and $\pi_{S''}$ are in F and hence in F' too. This imply that

$$\phi_k - \alpha_{2k} = \phi_3 + \alpha_{31}$$

Combining this equality with the value of ϕ_3 in (4.123) we obtain

$$\epsilon_l = \phi_k - \alpha_{2k} - \alpha_{31} - \alpha_{3l} + \alpha_{13} + \alpha_{32}.$$

Now if we replace ϵ_l in (4.132) by its value in the above equality we obtain

$$\mu_2 + \mu_3 = \alpha_{32}. \quad (4.135)$$

Since (4.112) is the sum of the equalities (4.113)-(4.118) with a careful counting one may see that

$$\rho = -\frac{1}{2}\alpha(A'_n) + \frac{1}{2}\alpha(\delta(l)) + \epsilon_l p - \alpha_{21} - \alpha_{13} - \alpha_{32} + \sigma(n-3). \quad (4.136)$$

Recall that from (4.120) we have $\epsilon_l = \beta_i + \frac{1}{2}\alpha(\delta(i)) - \alpha_{il} - \sigma$, for $i \notin \{1, 2, 3\}$. Replacing by this value of ϵ_l in (4.136) and using the fact that $\alpha_{ij} = \mu_i + \mu_j$ for $\{i, j\} \not\subseteq \{1, 2, 3\}$, we obtain

$$\beta_i = -(p-1)\mu_i + \frac{\rho - \sigma(n-3)}{p} + \sigma. \quad (4.137)$$

Using exactly the same combinations with respect to the value of ϵ_l in (4.121)-(4.123) and the fact that $\alpha_{21} = \mu_1 + \mu_2$, $\alpha_{13} = \mu_1 + \mu_3$ and $\alpha_{32} = \mu_2 + \mu_3$ we obtain

$$\beta_1 = -(p-1)\mu_1 + \frac{\rho - \sigma(n-3)}{p}, \quad (4.138)$$

$$\beta_2 = -(p-1)\mu_2 + \frac{\rho - \sigma(n-3)}{p}, \quad (4.139)$$

$$\beta_3 = -(p-1)\mu_3 + \frac{\rho - \sigma(n-3)}{p}. \quad (4.140)$$

Now using (4.119), (4.133), (4.134), (4.135) and (4.137)-(4.140) it is easy to check that (4.88) is obtained by the sum of the following linear constraints:

$$\begin{aligned} \mu_i & \times (x(\delta(i)) = (p-1)y_i) & \forall i \in V_n, \\ \left(\frac{\rho - \sigma(n-3)}{p}\right) & \times \left(\sum_{i \in V_n} y_i = p\right), \\ \sigma & \times (x_{12} + x_{23} + x_{31} + \sum_{i \in V_n \setminus \{1,2,3\}} y_i \leq n-3). \end{aligned}$$

4.3 Related studies

In the section we will discuss the cases where the valid inequalities introduced in the previous section are not defining facets for the FCLOP polytope. Let us first discuss the case of the triangle-free inequalities (4.14) and (4.15). The following Lemmas establish the relation between these two set of valid inequalities.

Let us consider the two inequalities below corresponding to (4.14) and (4.15) with respect to $i = 1, j = 2, k = 3$.

$$x_{12} + x_{23} + x_{31} + \sum_{i \in V_n \setminus \{1,2,3\}} y_i \leq \lfloor \frac{2p + (n-3)}{3} \rfloor, \quad (4.141)$$

$$x_{12} + x_{23} - x_{13} - y_2 \leq 0. \quad (4.142)$$

Let F_1 and F_2 be the faces of $FCLOP(G_n, p)$ defined by inequalities (4.141) and (4.142), respectively.

Lemma 6 *Inequalities (4.14) and (4.15) are defining the same facets for $FCLOP(G_n, n-1)$.*

Proof *Without loss of generality, we will show that (4.142) and (4.141) define the same facet for $FCLOP(G_n, n-1)$. In fact, we will prove that $F_1 = F_2$.*

Let $S \subseteq V_n$ with $|S| = n-1$ and π_S a linear order on S . It is easy to check that the incidence vector of π_S is in F_1 (resp. F_2) if and only if one of the following statements holds:

- (i) $\{1, 2, 3\} \subseteq S$ and $\pi^{-1}(1) < \pi^{-1}(2) < \pi^{-1}(3)$,
- (ii) $\{1, 2, 3\} \subseteq S$ and $\pi^{-1}(2) < \pi^{-1}(3) < \pi^{-1}(1)$,
- (iii) $\{1, 2, 3\} \subseteq S$ and $\pi^{-1}(3) < \pi^{-1}(1) < \pi^{-1}(2)$,
- (iv) $3 \notin S$ and $\pi^{-1}(1) < \pi^{-1}(2)$,
- (v) $2 \notin S$ and $\pi^{-1}(3) < \pi^{-1}(1)$,
- (vi) $1 \notin S$ and $\pi^{-1}(2) < \pi^{-1}(3)$.

It is also easy to check that the incidence vector of any linear order satisfying (i)-(vi) is in F_2 (resp. F_1). And since any vector in F_1 (resp. F_2) may be obtained as a convex combination of the 0-1 vectors in F_1 (resp. F_2) we conclude that $F_1 = F_2$. \square

Lemma 7 *Inequalities (4.15) do not define facets for $FCLOP(G_n, n-2)$.*

Proof Here we have $p = n - 2$. We will prove that $F_2 \subset F_1$. Let us list all the linear orders π_S having their incidence vectors in F_2 .

- (i) $\{1, 2, 3\} \subseteq S$ and $\pi^{-1}(1) < \pi^{-1}(2) < \pi^{-1}(3)$,
- (ii) $\{1, 2, 3\} \subseteq S$ and $\pi^{-1}(2) < \pi^{-1}(3) < \pi^{-1}(1)$,
- (iii) $\{1, 2, 3\} \subseteq S$ and $\pi^{-1}(3) < \pi^{-1}(1) < \pi^{-1}(2)$,
- (iv) $2 \notin S$, $\{1, 3\} \subseteq S$ and $\pi^{-1}(3) < \pi^{-1}(1)$,
- (v) $2 \notin S$ and $1 \notin S$,
- (vi) $2 \notin S$ and $3 \notin S$,
- (vii) $3 \notin S$, $\{1, 2\} \subseteq S$ and $\pi^{-1}(1) < \pi^{-1}(2)$,
- (viii) $1 \notin S$, $\{2, 3\} \subseteq S$ and $\pi^{-1}(2) < \pi^{-1}(3)$.

The incidence vector of any linear order satisfying (i)-(viii) is in F_1 . Consequently, $F_2 \subseteq F_1$. Now consider $S = V \setminus \{1, 3\}$. Then the incidence vector of any linear order π_S is in F_1 but not in F_2 . \square

The next lemma shows that (4.15) triangle-free inequalities define facet only for $FCLOP(G_n, n - 1)$ and for $FCLOP(G_n, n - 2)$.

Lemma 8 *Inequalities (4.15) are not defining facets for $FCLOP(G_n, p)$ when $p \leq n - 3$. Moreover, the following hold*

- (i) $\dim(F_1) = \frac{(n-p)(n-(p+1))}{2}$, when $n - 5 \leq p \leq n - 3$, and
- (ii) $F_1 = \emptyset$, when $p \leq n - 6$.

Proof We will show statements (i)-(ii). Then any one of these statements implies that inequalities (4.15) are not facet defining for $FCLOP(G_n, p)$ when $p \leq n - 3$.

- (i) $n - 5 \leq p \leq n - 3$. Observe that the right hand side of inequality (4.142) in this case is $n - p$. Notice that for any linear order π_S having its incidence vector in F_1 , we must have $S \cap \{1, 2, 3\} = \emptyset$. In fact, when $1 \leq |S \cap \{1, 2, 3\}| \leq 3$ then the left hand side of inequality (4.142) is at most $n - (p + 1)$. Hence the unique linear orders lying on the face F_1 are those with $S \cap \{1, 2, 3\} = \emptyset$, these are exactly the linear orders on $n - p$ elements. From Grötschel et al. (1985a) we know that we have at most $\frac{(n-p)(n-(p+1))}{2} + 1$ such a linear orders that are affinely independent.

(ii) $p \leq n - 6$. Let $p = n - j$. Notice that in any case the left hand side of inequality (4.142) is at most $n - j$. now to evaluate the right hand side we will distinguish the three cases below with $k \geq 2$.

1. $j = 3k$. The right hand side is equal to $n - (2k + 1)$,
2. $j = 3k + 1$. The right hand side is equal to $n - (2k + 2)$,
3. $j = 3k + 2$. The right hand side is equal to $n - (2k + 3)$.

In all the three cases we have $F_1 = \emptyset$. □

Observe from the lemma above the only cases where F_1 is not empty is when $S \cap \{1, 2, 3\} = \text{set}$. But in this case the incidence vector of any linear order belongs to F_2 . One can easily construct linear orders in F_2 but not in F_1 . Hence $F_1 \subset F_2$.

From the discussion above, one may expect that inequalities (4.141) define facet for $FCLOP(G_n, p)$ when $p \leq n - 3$. Unfortunately this is not the case and we have shown it by a numerical calculation.

To do so, we create a matrix B that contains all the active feasible solutions (the feasible solutions that hold as equality) as the columns (or rows) and then calculate the rank of B matrix. The fact that $\text{rank}(B) \leq \dim(FCLOP(G_n, p)) - 1$ with $p \leq n - 3$ confirms that these inequalities are not defining facet for the $FCLOP(G_n, p)$ with $p \leq n - 3$. As an example, we are going to show that the inequality $y_i \leq 1$ with $p = n - 2$ is not defining facet for the FCLOP polytope. the examples corresponding to other inequalities mostly need a huge B matrix. That is the reason why we show the method of calculation on a relatively small matrix and a simple inequality instance.

Example 1 Consider $n = 5$ and $p = 3$. In this case, regarding the corollary 1, we have $\dim(FCLOP(G_5, 3)) = 19$. Without loss of generality, we set $i = 1$ and show that the inequality $y_1 \leq 1$ is not defining facet for the $FCLOP(G_5, 3)$ polytope.

The set $y_1 \leq 1$ hold as equality for 36 feasible solutions that are reported in (Table 4.1). The number of affinely independent solutions that are reported in Table 4.1 is equal to rank of the B matrix that is equal to 16.

Since $\text{rank}(B) \neq \dim(FCLOP(G_5, 3)) - 1$, consequently, $y_1 \leq 1$ does not define a facet for the $FCLOP(G_5, 3)$, and in the same way, $y_i \leq 1$.

	x_{12}	x_{13}	x_{14}	x_{15}	x_{21}	x_{23}	x_{24}	x_{25}	x_{31}	x_{32}	x_{34}	x_{35}	x_{41}	x_{42}	x_{43}	x_{45}	x_{51}	x_{52}	x_{53}	x_{54}	y_1	y_2	y_3	y_4	y_5
1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
2	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0
3	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
4	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
5	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0
6	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1
7	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	0
8	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0
9	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1
10	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1
11	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
12	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
13	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
14	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0
15	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
16	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
17	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0
18	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1
19	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
20	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0
21	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1
22	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
23	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	1	0
24	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1
25	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	1	0
26	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	0
27	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1
28	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	1	0
29	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	0
30	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1
31	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1
32	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1
33	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1
34	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1
35	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1
36	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1	1

Table 4.1: Matrix B contains the feasible solutions such that $y_1 = 1$ for all of them

By doing the same, we have shown for some other valid inequalities that they are not defining facet for the FCLOP polytope. Table 4.2 reports the numerical results to show that some of the proposed classes of valid inequalities are not defining facets for the FCLOP polytope.

In the following, we will discuss the set of valid inequalities that are introduced earlier in this chapter and show that they are not defining facets for the FCLOP polytope by calculation rank of the B matrix as it is explained earlier.

Table (4.2) reports rank of obtained B matrices for different n and p values for different inequalities. tested in the same manner. The list of valid inequalities that have been tested are in the following:

- Const. 1:

$$x_{ij} + (x_{ik} + x_{ki}) - (x_{kj} + x_{jk}) - y_i \leq 0 \quad \forall i, j, k \in V_n : i \neq j \neq k$$

- Const. 2:

$$x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V_n, i \neq j, \quad p < n - 1$$

- Const. 3:

$$x_{ij} + x_{ji} - y_i - y_j + 1 \geq 0 \quad \forall i, j \in V_n, i < j, \quad p < n - 1$$

	#Nodes	#Cardinality	dimension	rank(B)
Const. 1	5	4	14	9
	6	4	29	23
Const. 2	5	3	19	17
Const. 3	5	3	19	17

Table 4.2: Rank of B matrices for different inequalities and different p values

In Table 4.2, the first column refers to the set of inequalities. Second column, $\#Node$ reports n , the number of nodes, and the third column refers to p , the cardinality number. The column *dimension* reports dimension of the FCLOP polytope corresponds to the aforementioned n and p (see Corollary 1). Finally the last column, $rank(B)$, reports rank of the priory defined B matrix.

4.4 Conclusion

In this chapter we discussed the inequalities that define facets for $FCLOP(G_n, p)$. In particular, we show that in some cases we may have inequalities that define the same facets. This means that we may remove one of these sets of inequalities only when a complete description for $FCLOP(G_n, p)$ is known. Since we do not know this description, all the inequalities may be useful for our linear relaxation even when they define the same facets or when the faces defined by some inequalities are included in the faces defined by other set of inequalities.

Lagrangian Relaxation

Lagrangian relaxation is mainly based on identifying the set of *complicating constraints*. That is, one identifies a set of constraints without which the problem can be solved more efficiently. Such constraints are relaxed in the Lagrangian fashion using Lagrange (dual) multipliers and penalize objective function upon violation of the relaxed constraints (Held and Karp, 1970, 1971; Geoffrion, 1974).

In this chapter we propose different Lagrangian relaxations for the FCLOP problem and compare the quality of obtained dual bound.

5.1 Introduction

Relaxation is a classical strategy in constrained optimization (see Geoffrion (1974) or Lemaréchal (2001) for basic results).

When solving a MIP, continuous relaxation is a practical tool as a direct bound can be completed by a performant LP solver.

Different kind of relaxations have been reported in the LOP literature including binary relaxation, semi-definite relaxation, linear relaxation, Lagrangian relaxation etc. Very often it is assumed that the better is the quality of resulting bound from such relaxation, the more likely that branch-and-bound-like methods would perform in converging to optimality.

Lagrangian Relaxation

When dealing with optimization problems which are not convex such as (M)IPs, Lagrangian relaxation is a method to find a dual bound on such optimization problems.

Given a (mixed integer) linear programming problem:

$$(P) : \quad \begin{aligned} & \max \quad f(x) \\ & \quad \quad g(x) \leq 0 \\ & \quad \quad x \in X \end{aligned}$$

where $g(x)$ represents the set of complicating constraints. In this notation, X may contain constraints as well as the sign restrictions. It may also contain real, binary or general integer variables.

The constraints $g(x) \leq 0$ are assumed to be the *complicating constraints*, in the sense that problem (P) without them could be solved much easier (e.g. it possess particular structure for which very sophisticated solution approach exists).

Definition 1 (The Lagrangian Relaxation of (P) relative to $g(x) \leq 0$) *The Lagrangian Relaxation of (P) relative to the complicating constraints $g(x) \leq 0$ with nonnegative multipliers λ (dual variables) is defined below:*
Lagrangian function:

$$L(x, \lambda) = (f(x) - \lambda \cdot g(x)) \quad x \in X, \quad \lambda \geq 0 \quad (5.1)$$

Lagrangian relaxation:

$$LR_\lambda : \quad v(\lambda) = \sup_{x \in X} L(x, \lambda) \quad (5.2)$$

The constraints $g(x) \leq 0$ are said to be dualized in Lagrangian form.

The dual multipliers λ are interpreted here as the prices to pay to force the relaxed constraints.

Due to the weak duality theorem, we have:

$$\forall x \in X, \quad \forall \lambda \geq 0 \quad \text{s.t.} \quad g(x) \leq 0 : \quad f(x) \leq v(\lambda) \quad (5.3)$$

Definition 2 (Lagrangian dual problem) *The problem of finding the tightest Lagrangian dual bound on $v(\lambda)$ is known as the Lagrangian dual problem.*

Dual problem (least upper bound):

$$\inf_{\lambda \geq 0} (v_\lambda) = \inf_{\lambda \geq 0} \left(\sup_{x \in X} L(x, \lambda) \right) \quad (5.4)$$

Observation Let $x(\lambda)$ denote optimal solution to the (LR_λ) for some $\lambda \geq 0$. If $g(x(\lambda)) \leq 0$, then it is optimal for (P) if $\lambda \cdot g(x(\lambda)) = 0$. In this case, $f(x(\lambda)) = v_\lambda$ (due to the strong duality theorem).

Clearly, problem (LR_λ) is a relaxation of (P) and the set of feasible solutions of the primal problem (P) , $FS(P)$, is a subset of those of (LR_λ) , $FS(LR_\lambda)$.

$$FS(P) \subseteq FS(LR_\lambda)$$

Moreover, if we call v_{LP} the optimal value of linear programming relaxation of P , and v_{LR} the optimal dual value of the Lagrangian relaxation problem of P and v_P^* the optimal value of P , then for a maximization problem we always have,

$$v_P^* \leq v_{LR} \leq v_{LP}$$

In addition, whenever the sub-problem has integrality property, then we always have $v_{LR_\lambda} = v_{LP}$ (Lemaréchal, 2001).

Yet, the key step in Lagrangian relaxation is the identification of complicating constraints.

Lagrangian decomposition

In many optimization problems such as network design, routing etc., often relaxing a set of linking constraints in a Lagrangian fashion leads to a decomposable structure when the problem is block separable. In such cases we can have less difficulties to find the optimal of each subproblem. Another way is to decompose the problem by splitting the variables. Guignard and Kim (1987) provides some tips and tricks such as duplicating variables and constraints in Lagrangian fashion facilitate decomposing the problem into smaller subproblems for each of which (or perhaps some of them), efficient solution methods may exist.

It is shown that, under certain circumstances, the bound resulted from the Lagrangian decomposition can be even better than the bound generated by the corresponding Lagrangian relaxation (Guignard and Kim, 1987).

Lagrangian Decomposition - two sub-problems

Given a (mixed integer) linear programming problem:

$$\begin{aligned} (P) : \quad & \max \quad c \cdot x \\ & x \in S_1 \cap S_2 \quad \leftarrow (\text{complicating constraints}) \\ & x \in X \end{aligned}$$

where X may contain constraints as well as the sign restrictions that may also contain real, binary or general integer variables.

By introducing copy variables $y \in X$ when $y = x$, we may have

$$\begin{aligned} (P') : \quad & \max f_1(x) + f_2(y) \\ & x \in S_1 \\ & y \in S_2 \\ & x - y = 0 \\ & x, y \in X \end{aligned}$$

By dualizing the copy constraints we will have:

$$\begin{aligned} (LD_u) : \quad & \max c \cdot x + u(y - x), \\ & x \in S_1 \\ & y \in S_2 \\ & x, y \in X. \end{aligned}$$

Now, it is straightforward to decompose the LD_u as in the following:

$$\begin{aligned} (LD_u(x)) : \quad & \max c \cdot x - ux, \\ & x \in S_1 \qquad \qquad \qquad x \in X. \end{aligned}$$

and

$$\begin{aligned} (LD_u(y)) : \quad & \max uy, \\ & y \in S_2. \end{aligned}$$

Figure 5.1 sheds more light on it (Guignard and Kim, 1987).

As it is shown in Figure 5.1, if one of the sub-problems has integrality property, then the quality of dual bound obtained by lagrangian decomposition method can not be better than the lagrangian relaxation dual bound. In other word, if none of the sub-problems has integrality property, then the LD dual bound can be even tighter than LR dual bound.

Another interest of applying this method is to have several but easier to solve sub-problems.

Furthermore, we may split the set of variables to three groups and introduce a lagrangian decomposition to three sub-problems. This helps to have still more easier to solve sub-problems and may helps to improve the quality of dual bound as it is confirmed by the numerical results presented at the end of the chapter.

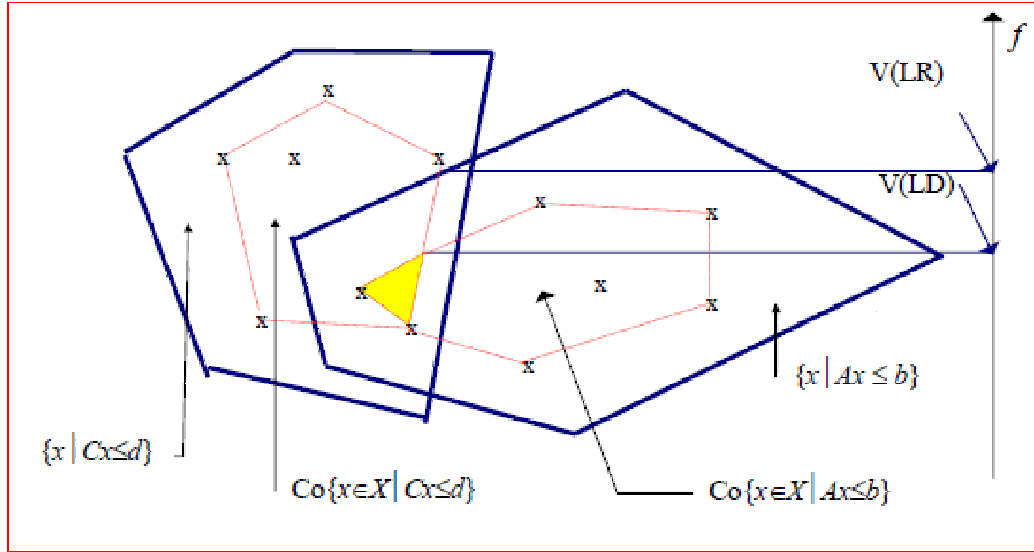


Figure 5.1: Lagrangian decomposition, two sub-problems (Guignard and Kim, 1987).

Lagrangian Decomposition - three sub-problems

Given a (mixed integer) linear programming problem:

$$(P) : \quad \max \quad f_1(x) + f_2(x) + f_3(x) \\ x \in X = X_A \cap X_B \cap X_C$$

X contains a set of constraints as well as the bound restrictions. The optimization problem is assumed easy to solve if we had any of the $f_1(x)$, $f_2(x)$ or $f_3(x)$ as the objective function (or there exist efficient tools to solve them) but it becomes difficult to solve by having all the parts at the same time.

Let us introduce two different sets of copy variables y and z .

$$(P') : \quad \max \quad f_1(x) + f_2(y) + f_3(z) \\ x - y = 0 \\ x - z = 0 \\ x \in X_A, \quad y \in X_B, \quad z \in X_C$$

After relaxing the two copy constraints we have:

$$(P') : \quad \max \quad f_1(x) + f_2(y) + f_3(z) - \lambda(x - y) - \mu(x - z) \\ x \in X_A, \quad y \in X_B, \quad z \in X_C$$

Then by decomposing P' we will have the following three sub-problems:

(P_A)

$$\max \quad f_1(x) - \lambda x - \mu x$$

$$x \in X_A$$

$$(P_B)$$

$$\max_{y \in X_B} f_2(y) + \lambda y$$

$$(P_C)$$

$$\max_{z \in X_C} f_3(z) + \mu z$$

Figures 5.2 and 5.3 shed more lights on this concept.

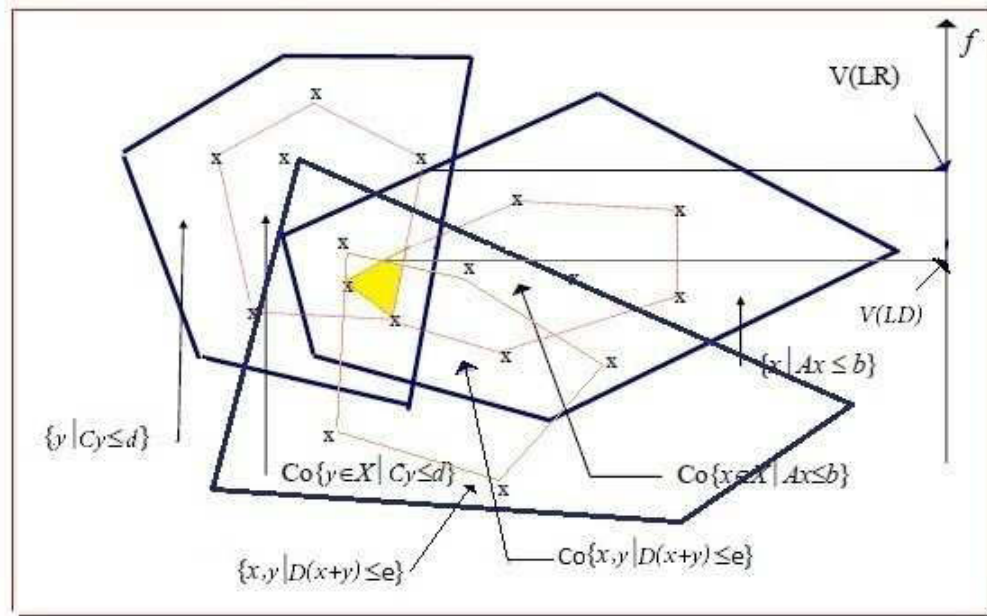


Figure 5.2: Lagrangian decomposition, with three sub-problems, where decomposing the model to three subproblems may improve the LP bound (Guignard and Kim, 1987).

In Figure 5.2, after decomposing the initial IP model to three sub-problems, the dual bound obtained is improved.

Also there are situations, as shown in Figure 5.3, that decomposing the model to three sub-problems does not contribute in improving the dual bound as much as decomposing the model to two sub-problems.

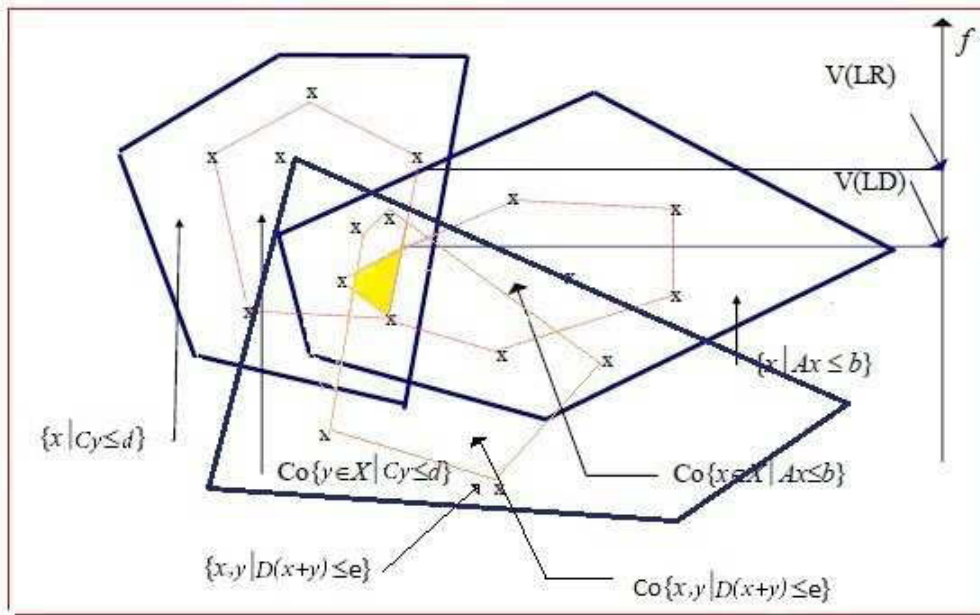


Figure 5.3: Lagrangian decomposition, with three sub-problems where depending on the subproblems properties and the objective function, Lagrangian decomposition technique may not improve the LP bound. It can still improve the computational time (Guignard and Kim, 1987).

Algorithms to solve the Lagrangian dual problem

Lagrangian dual problem LR_λ is a piecewise linear function, and in general it is a non-differential optimization problem. A very successful, simple and well-known method to solve the Lagrangian dual problems is Subgradient method (Polyak, 1969).

Subgradient method was originally proposed in the 60s in the former Soviet Union. Very similar method has also been proposed in Held and Karp (1971) for solving traveling salesman problem. Later, Lemarechal (1975) proposed the well-known *Bundle* methods as an extension of subgradient. The volume algorithm was proposed in Barahona and Anbil (2000) as an algorithm which simultaneously produces a primal feasible solution for the problem as well as a dual bound.

The definition of step size λ in subgradient method is of crucial importance, the speed of convergence depends heavily on it. There are several studies in the literature on proposing different step-size update scheme to speed up the convergence of the algorithm. A very commonly used step size for the subgradient method is proposed in Held et al. (1974).

The bundle method is based on the idea of storing information $\langle v(\lambda), g(x(\lambda_t)) \rangle$ ($v(\lambda)$ is the dual objective and $g(x(\lambda_t))$ is subgradient at iteration t) about

previous iterations in a set β , the *bundle*, in order to exploit such information to chose a good decent direction (when minimizing dual problem). The new point are then chosen along this decent direction. As in contrast to the so called simple *subgradient method*, within which only the information about the present iterations is used to guide the search, the *bundle* is in fact exploiting the current and past memory to guide the search towards optimal values (Lemarechal, 1978).

The major difference between different bundle methods lies in the size of the bundle and the memory.

For the FCLOP problem, we propose different relaxation schemes based on the idea of relaxing complicating constraints. We have studied different relaxations strategies and different sub-problems to identify the best constraints to relax and compared the quality of obtained dual bound.

We have also studied several Lagrangian decomposition schemes, which result notably high quality dual bounds.

To solve all these variants of the Lagrangian dual problems for the FCLOP, we have applied Bundle method and proposed a comparison between different relaxation strategies.

5.2 Complicating constraints for the FCLOP

Linear Ordering Problem is already known as a NP-hard problem and the computational effort needed for solving even moderate size instances increase dramatically. Yet, it is an easier problem comparing to the FCLOP. There are several cases where an easy-to-solve instance of the LOP (which is solved in a few seconds) is not efficiently solvable for some cardinality number $p < n$.

The mathematical formulations corresponding to the FCLOP with $0 \leq p \leq n - 1$ follows:

FCLOP(G_n, p) :

$$\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \quad (5.5)$$

s.t.

$$x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V_n, i < j \quad (5.6)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V_n, i < j \quad (5.7)$$

$$x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall i, j \in V_n, i < j \quad (5.8)$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall i, j, k \in V_n \quad i \neq j \neq k \quad (5.9)$$

$$\sum_{i \in V_n} y_i = p \quad (5.10)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n, i \neq j \quad (5.11)$$

As numerical results show in Tables (3.2), (3.7) and (3.6), we can strengthen the formulation by adding some valid equalities and inequalities. This includes:

$$x_{ij} + x_{ji} - y_i - y_j = -1 \quad \forall i, j \in V_n, i < j \quad (5.12)$$

when $p = n - 1$, and

$$\sum_{j \in V_n, j \neq i} (x_{ij} + x_{ji}) = (p - 1)y_i \quad \forall i \in V_n \quad (5.13)$$

for the cases when $2 \leq p \leq n - 2$.

It is shown in numerical results that we can significantly improve the formulation and reduce integrality gap as well as the computational time required to solve the instances by adding (5.12) and (5.13) to the model (see Table 3.3).

In chapter 4, we have shown that (5.10) together with (5.12) for the case $p = n - 1$ and (5.10) together with (5.13) for the case $p \leq n - 2$ are defining the *minimal equation system* for the corresponding polytope. However comparing the numerical results presented in Tables 3.2 and 3.3 indicates that it is always advantageous if we add

$$\sum_{i \in V_n} \sum_{j \in V_n, j < i} (x_{ij} + x_{ji}) = p(p - 1)/2. \quad (5.14)$$

Numerical experiments have shown that we can improve the integrality gap and reduce the computational time by adding only one equation (5.14) for all the cardinality numbers $0 \leq p \leq n$, instead of adding a large number of equations (5.12) for the case of $p = n - 1$ and (5.13) for the cases with $p \leq n - 1$. The integrality gap remains the same since (5.14) is in fact equal to the aggregation of the equations (5.13) over the index i (also, it is equal to the sum of the (5.12) over all the indices i and j , $i \neq j$).

Let us introduce following notations to facilitate referring to the constraint sets:

$$A^{Tournament} : \begin{cases} x_{ij} + x_{ji} \leq y_i & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} \leq y_j & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} \geq y_i + y_j - 1 & \forall i, j \in V_n, i < j \end{cases}$$

$$B^{Cycle-free} : \begin{cases} x_{ij} + x_{jk} + x_{ki} \leq 2 & \forall i, j, k \in V_n \quad i \neq j \neq k \end{cases}$$

$$C^{Cardinality} : \begin{cases} \sum_{i \in V_n} y_i = p \\ \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} = p(p - 1)/2 \end{cases}$$

Now, we are ready to introduce different Lagrangian dual problems corresponding to the above subsets of constraints. By applying the Lagrangean

relaxation for a maximization problem, one obtains the lowest upper bound when dualizing a given subset of constraints. That upper bound is known to be at least as good as the continuous relaxation of the integer program (see Fisher (1981)). For example, if subset A is chosen to be relaxed, the dual bound will be equal to the maximal value of the objective function on the intersection of the polyhedron defining the continuous relaxation of A with $\text{Conv}(\Omega_B) \cap \text{Conv}(\Omega_C)$ where $\text{Conv}(S)$ is the convex hull of a set S . Let us resume the three options that arise when relaxing each subset of constraints A, B or C :

1. *Relaxing the triangle-free inequalities* : this induces a p -cardinality tournament (subtournament) subproblem which is NP-hard; the number of dual variables is huge, one for each possible triangle-free. Preliminary results showed that the reduction of the LP gap is substantial and improves when p gets smaller.

Observe that the dual iterations will lead to a solution in

$$(P_A \cap P_C) \cap \text{Conv}(\Omega_B)$$

and the upper bound v_B can be better than the continuous relaxation bound v_{LP} .

2. *Relaxing the tournament constraints* : it will induce a decomposition between x and y variables where the cardinality subproblem is trivially solved and integral (see below); on the other hand, the cycle-free subproblem is NP-hard. Preliminary results showed that the corresponding bound and performance are subsumed by the former relaxation.
3. *Relaxing the cardinality constraints* : the subproblem looks like a LOP but it is not, as the dual multiplier can be negative forcing some y_i to be zero. Even if that option does not yield the best upper bound, it is worth analyzing the dual function to identify the favourable cases. Indeed, the dual function is a piecewise affine convex function. We can restrict the study to the case of a non negative multiplier as the cardinality constraint can be relaxed to $\sum_i y_i \leq p$ without altering the optimality conditions. The dual function is defined by :

$$\phi(u) = p \cdot u + \sup \left\{ \sum_{i,j} c_{ij} x_{ij} - u \sum_i y_i \mid \text{s.t. A and B} \right\}$$

Obviously, $\phi(0) = v_n$, the optimal value of LOP for the n nodes. When u increases slightly, the solution will not change and the dual function behaves like the affine function $v_n + (p - n)u$. At the opposite, if u is very large, no node will be selected and the solution will be $x = 0$ with corresponding value $v_0 = 0$, so that the corresponding piece will be linear with slope p . Now, let v_j be the optimal value of (j -FCLOP)

(then $v_n \geq v_{n-1} \geq \dots \geq v_j \geq \dots \geq v_1 = v_0 = 0$). Then, the dual function can be written as a piecewise affine convex function :

$$\phi(u) = \max\{v_j + (p - j)u, j = 0, \dots, n\}$$

Observe that the affine piece corresponding to $j = p$ is horizontal, but it is not necessarily active in the sense that $\phi(u) > v_p, \forall u$ which implies the existence of a duality gap (see Figures 5.4 and 5.4).

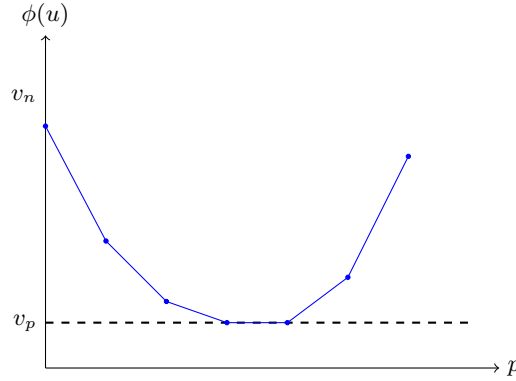


Figure 5.4: A sample of $\phi(u)$, that has an active piece with $j = p$. There is no duality gap in such a case.

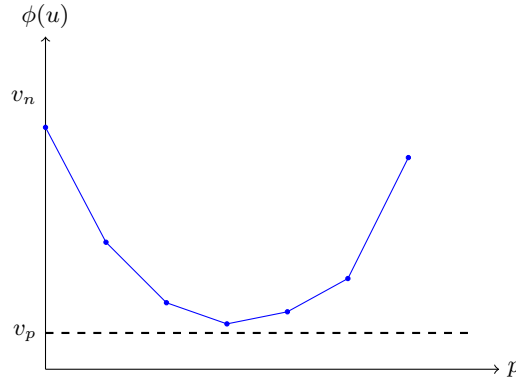


Figure 5.5: A sample of $\phi(u)$, that has no active piece with $j = p$. In such a case, the duality gap is not zero.

The first Lagrangian relaxation model is proposed in the following.

5.2.1 FCLOP-LR-A (relaxing the Tournament inequalities)

$$\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} + \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} u_{ij} (y_i - x_{ij} - x_{ji})$$

$$\begin{aligned}
& + \sum_{i \in V_n} \sum_{j \in V_n, j < i} v_{ij}(-y_i - y_j + 1 + x_{ij} + x_{ji}) \\
& s.t. \\
& (x, y) \in B \cap C \\
& x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n
\end{aligned}$$

In the *FCLOP-LR-A* model we relax all the constraints that link the variables x_{ij} and y_i for all $i, j \in V_n$.

The sub-problem in this case, contains triangle-free inequalities (B) and cardinality constraints (C). In this sub-problem we have considered the two equations to limit the number of nodes and also the number of arcs in any feasible solution.

The sub-problem in this case becomes automatically decomposable to two blocks based on the the two sets of variables x and y .

Bundle method has been applied to solve the dual problem and the numerical results is reported in Table 5.1.

The second lagrangian relaxation of the FCLOP model that we studied is presented in the following.

5.2.2 FCLOP-LR-B (*relaxing triangle-free inequalities*)

$$\begin{aligned}
& \max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} - \sum_{k \in V_n} \sum_{j \in V_n, j \neq k} \sum_{i \in V_n, i < j, i < k} v_{ijk}(x_{ij} + x_{jk} + x_{ki} - 2) \\
& s.t. \\
& (x, y) \in A \cap C \\
& x_{ij}, y_n \in \{0, 1\} \quad \forall i, j \in V_n
\end{aligned}$$

In this model, we relax B and add the inequalities to the objective function multiplied by the Lagrangian multipliers v_{ijk} .

The sub-problem in this model represents a p-tournament problem Moon (1966) in graph theory which, consists of cardinality constraints (C) and tournament problem (A) as well as bound constraints. Tournament problem in graph theory is looking for a digraph by assigning an orientation to all the edges of an undirected complete graph. A feasible solution in tournament problem in graph theory is a digraph which contains a directed arc between every pair of its vertices. Moreover, a feasible solution in the p-tournament problem consists of a digraph that contains p nodes ($p \leq n$) such that there exists one directed arc between every pair of them.

We have tested "*FCLOP-LR-B*" relaxation model and applied Bundle method to solve it. Numerical results are presented in Table 5.2.

The third Lagrangian relaxation of the FCLOP model that we studied is presented in the following.

5.2.3 FCLOP-LR-C *relaxing cardinality constraints*

The second combination of constraints that have been relaxed are the group C of constraints concerning the restrictions on number of nodes and arcs in any feasible solution.

$$\begin{aligned} & \max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} + u[p - \sum_{i \in V_n} y_i] \\ & s.t. \\ & \quad (x, y) \in A \cap B \\ & \quad x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n \end{aligned}$$

Bundle method is applied to solve this Lagrangian relaxation dual problem and the numerical results are presented in Table 5.3.

5.2.4 FCLOP-LD1 —Two sub-problems

In the sequel, a Lagrangian decomposition model is proposed. In this Lagrangian decomposition model, a set of copy constraints

$$t_{ij} - x_{ij} = 0$$

has been proposed to decouple the x variables to two sets of x and t variables. In this decomposition strategy, the objective is to decompose the set of triangle-free inequalities in a block of constraints and keep all the rest (the groups A and C) in the second block.

FCLOP-LD1

$$\begin{aligned} & \max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \\ & s.t. \\ & A : \begin{cases} x_{ij} + x_{ji} \leq y_i & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} \leq y_j & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} \geq y_i + y_j - 1 & \forall i, j \in V_n, i < j \end{cases} \\ & B : \begin{cases} t_{ij} + t_{jk} + t_{ki} \leq 2 & \forall i, j, k \in V_n \quad i \neq j \neq k \end{cases} \\ & C : \begin{cases} \sum_{i \in V_n} y_i = p \\ \sum_{i, j, i \neq j} x_{ij} = p(p-1)/2 \end{cases} \end{aligned}$$

$$\begin{aligned}
x_{ij} - t_{ij} &= 0 & \forall i, j \in V_n, i \neq j \\
t_{ij}, x_{ij}, y_i &\in \{0, 1\} & \forall i, j \in V_n, i \neq j
\end{aligned}$$

Consider the sub-problem $LD_u(x, y)$ and $LD_u(t)$ as follows:

$$\begin{aligned}
&\mathbf{LD_u(x,y):} \\
&\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} - \sum_{i \in V_n} \sum_{j \in V_n} u_{ij} x_{ij} \\
&s.t. \\
&\quad (x, y) \in A \cap C \\
&\quad x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{LD_u(t):} \\
&\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} u_{ij} t_{ij} \\
&s.t. \\
&\quad (t) \in B \\
&\quad t_{ij} \in \{0, 1\} \quad \forall i, j \in V_n.
\end{aligned}$$

In the *FCLOP-LD1* formulation, by applying a set of copy variables t_{ij} , we have decomposed the FCLOP to two sub-problems $LD_u(t)$ and $LD_u(x, y)$. One observes that the block $LD_u(x, y)$ is quit similar to the *FCLOP-LR-B*. The main difference between FCLOP-LD1 and FCLOP-LR-B is in the objective function of the dual problem corresponding to each of them. Since we obtain relatively high quality dual bound by applying the FCLOP-LR-B relaxation strategy, the objective in implementing this Lagrangian decomposition strategy was to see whether this decomposition strategy produce better dual bound or it becomes more difficult.

Bundle method has been applied to solve the model and the numerical experiments that are reported in Table 5.4 confirm that the Lagrangian relaxation strategy FCLOP-LR-B is still more interesting than FCLOP-LD1 in terms of the quality of obtained dual bound.

5.2.5 FCLOP-LD2 —Three sub-problems

As mentioned above, the objective in lagrangian decomposition is to decouple the constraints in separated blocks and dualize the set of copy constraints somehow to decompose the subproblem to two or more easier sub-problems. We have applied the idea to decompose the model into three blocks to find

the sub-problems that are even easier to solve. In this section, a Lagrangian decomposition model which decomposes the dual problem into three sub-problems LD_A , LD_B and LD_C is proposed.

This is accomplished by introducing the following copy constraints:

$$\begin{aligned} x_{ij} - t_{ij} &= 0 & \forall i, j \in V_n \\ y_i - w_i &= 0 & \forall i \in V_n \end{aligned}$$

By substituting the copy variables t_{ij} instead of x_{ij} in triangle-free inequalities and w_i instead of y_i in cardinality constraint we obtain the LD_2 -FCLOP formulation as follows:

FCLOP-LD-2

$$\begin{aligned} \max \quad & \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} - \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} \lambda_{ij} (x_{ij} - t_{ij}) - \sum_{i \in V_n} \mu_i (y_i - w_i) \\ A : \quad & \begin{cases} x_{ij} + x_{ji} \leq y_i & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} \leq y_j & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} \geq y_i + y_j - 1 & \forall i, j \in V_n, i < j \\ \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} = p(p-1)/2 \end{cases} \\ B : \quad & \begin{cases} t_{ij} + t_{jk} + t_{ki} \leq 2 & \forall i, j, k \in V_n, i \neq j \neq k \end{cases} \\ C : \quad & \begin{cases} \sum_{i \in V_n} w_i = p \end{cases} \\ & x_{ij}, y_i, t_{ij}, w_i \in \{0, 1\} \quad \forall i, j \in V_n, i \neq j \neq k \end{aligned}$$

Now it is straightforward to decompose the $FCLOP$ -LD-2 formulation to the following three sub-problems:

Sub-problem A

$$\begin{aligned} \max \quad & \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} - \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} \lambda_{ij} x_{ij} - \sum_{i \in V_n} \mu_i y_i \\ s.t. \quad & \begin{aligned} x_{ij} + x_{ji} &\leq y_i & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} &\leq y_j & \forall i, j \in V_n, i < j \\ x_{ij} + x_{ji} &\geq y_i + y_j - 1 & \forall i, j \in V_n, i < j \\ \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} &= \frac{p(p-1)}{2} \\ x_{ij}, y_i &\in \{0, 1\} & \forall i, j \in V_n, i \neq j \end{aligned} \end{aligned}$$

Sub-problem B

$$\begin{aligned}
& \max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} \lambda_{ij} t_{ij} \\
& s.t. \\
& t_{ij} + t_{jk} + t_{ki} \leq 2 \quad \forall i, j, k \in V_n, \quad i \neq j \neq k \\
& t_{ij} \in \{0, 1\} \quad \forall i, j \in V_n
\end{aligned}$$

Sub-problem C

$$\begin{aligned}
& \max \sum_{i \in V_n} \mu_i w_i \\
& s.t. \\
& \sum_{i \in V_n} w_i = p \\
& w_i \in \{0, 1\} \quad \forall i \in V_n
\end{aligned}$$

Sub-problem A contains both sets of decision variables x and y , since the group A refers to the Tournament constraints.

Sub-problem B, contains only the t variables which are substituted by x variables in triangle-free inequalities (B). In fact the sub-problem B represents a maximum acyclic subgraph problem when the input matrix corresponds to a complete weighted oriented graph and the objective aims to maximize the sum of the weights of all the arcs present in a feasible solution. There is no other constraint than triangle-free inequalities in this sub-problem. Consequently as a feasible solution, we might have a linear order containing all the n vertices only if 1) $w_{ij} \geq 0$ for all $i, j \in V_n$ and 2) for any pair of vertices i and j , we have $w_{ij} + c_{ji} > 0$. In fact two vertices i and j will never be connected directly if $w_{ij} = c_{ji} = 0$.

In *Sub-problem C*, all what remains is cardinality constraint that restricts the number of vertices in any feasible solution. This sub-problem represents a knapsack problem, while the weights of all the variables are equal to +1. Therefor it becomes trivial and we only need to sort the coefficients of the objective function in decreasing order and choose the first p variables to be equal to one.

One observes that all the sub-problems A, B and C are quite easier to solve compared to the basic FCLOP model.

Numerical results of applying Bundle method to solve the lagrangian decomposition model $LD_2\text{-}FCLOP$ are reported in Table 5.5.

5.3 Numerical results

This section is devoted to report the numerical results of applying the Bundle method to solve different Lagrangian dual problems that are presented previously.

The set of instances are the same as what we have considered in chapter 3. We have used Bundle code developed by Frangioni and presented in (Frangioni, 2000) to solve dual problems. Three different termination conditions have been considered: 1) the maximum number of iteration (100 iterations at most) 2) 10800 seconds as the time limit and 3) the average violation that is to certify the convergence. The bundle size has been considered up to 100 solutions.

The first column of each table indicates name of the instances in the format 'Name-p' where name is the instance name from LOLIB and p indicates the cardinality number.

The second column 'elapsed time' reports the time (in second) elapsed in solving the instance.

The third column reports the best obtained dual bound.

The following column reports the number of iterations that have been carried out before the termination of the procedure.

the next column reports the LP bound corresponding to each instance. The next column 'Optimal*' reports value of the optimal solution for the instances of which their optimal solution is known. There are also instances of which are not solved yet to optimality. In such cases, the best known primal bound has been reported superscripted by a *.

Finally, the last column reports the termination status. *AvgViol* indicates that the run has been terminated after being sufficiently converged to the optimal solution. *MaxIter* means that the run has been terminated after 100 iterations. *TimLim* indicates that the run procedure has been terminated by the time limit termination condition.

There are instances of which are terminated by *TimLim* termination condition, however the elapsed time is slightly different from 10800 seconds. The small deviation from the time limit is due to the fact that the time limit is reached while CPLEX has been in the middle of solving an LP node and does not terminate before the LP is resolved.

Finally the term *O.M.* refers to an unusual termination status that happens due to the memory issue (out of memory).

Note 2 *It may happen in some instances (e.g. N-atp24-23, N-atp24-24, N-atp48-40, N-atp48-45,...) that the LR dual bound obtained from the sub-gradient optimization is greater than its LP bound which is not normal. In fact, in all these instances, the termination mode is either time limit or the number of iterations. Such an incorrect dual bound means that the rate of*

convergence was slow enough to avoid the algorithm finding a better bound in a reasonable time or in other words, the only very few iterations have been completed within the time limit.

5.3.1 FCLOP-LR-A

Table 5.1 contains the numerical results concerning the model "*FCLOP-LR-A*".

As explained earlier, in this relaxation strategy, the group A of inequalities that represent the tournament problem, is dualized. Numerical results in Table 5.1 show that in a very few instances of which we have examined, the Lagrangian relaxation model could improve the LP bound. In fact among all the dual problems that we have examined, the "*FCLOP-LR-A*" had the less impact to improve the dual bound.

Also as it is mentioned in Note 2, in some instances that the procedure has been terminated by MaxIter or TimLim, there are instances (e.g. N-atp24, N-atp48, N-be75eec and N-econ36) of which for some cardinality, the LP bound is slightly better than the dual bound.

Concerning the difficult instances (e.g. N-p50-20 and N-pal43), the procedure has ran out of memory for all the cases.

In general one may say that *FCLOP-LR-A* does not produce sufficiently good dual bounds and relaxing the tournament inequalities may not be the best choice even if the resulting subproblem becomes decomposable and easier to solve.

Fname	elapsed Time	bestDual Bound	nrIteration	LP bound	Optimal*	Relative Gap(%)	BundleStatus
N-atp24-5	40	41.04	90	41.04	18	128	AvgViol
N-atp24-10	52	82.08	85	82.08	53	54.86	AvgViol
N-atp24-15	70	123.12	100	123.12	94	30.97	MaxIter
N-atp24-20	123	153.54	100	153.16	140	9.67	MaxIter
N-atp24-23	145	169.16	100	168.25	165	2.52	MaxIter
N-atp24-24	36	172	57	172	172	0	AvgViol
N-atp48-10	572	118.54	100	118.54	53	123.66	MaxIter
N-atp48-15	718	177.81	100	177.81	97	83.30	MaxIter
N-atp48-20	878	237.08	100	237.08	148	60.18	MaxIter
N-atp48-25	781	296.36	100	296.35	205	44.56	MaxIter
N-atp48-30	1350	357.3	100	355.62	264	35.34	MaxIter
N-atp48-35	4382	401.62	100	400.65	324	23.95	MaxIter
N-atp48-40	11590	451.65	35	436.16	387	16.70	TimLim
N-atp48-45	11140	478.86	53	466.62	447	7.12	TimLim
N-atp48-47	13703	489.8	6	477.7	472	3.77	TimLim
N-atp48-48	15170	485.7	9	483	483	0.55	TimLim
N-be75eec-20	1233	151295	100	132966	130506	15.92	MaxIter
N-be75eec-30	1321	205996	100	191168	191061	7.81	MaxIter
N-be75eec-40	1401	230870	100	224006	223672	3.21	MaxIter
N-be75eec-50	1628	236464	100	236464	236464	0	MaxIter
N-econ36-15	367	450166	100	403222	400547	12.38	MaxIter
N-econ36-20	409	503684	100	470033	469584	7.26	MaxIter
N-econ36-25	480	530660	100	507568	506232	4.82	MaxIter
N-econ36-30	677	548190	100	535017	534406	2.57	MaxIter
N-econ36-35	1152	549164	100	547984	547501	0.30	MaxIter
N-econ36-36	1033	548581	100	548588	548574	0.001	MaxIter
N-p50-20-10	-	-	-	13282.6	2311*	-	O.M.
N-p50-20-15	-	-	-	19923.9	5106*	-	O.M.
N-p50-20-20	-	-	-	26565.2	7397*	-	O.M.
N-p50-20-25	-	-	-	33206.5	13697*	-	O.M.
N-p50-20-30	-	-	-	39847.8	17052*	-	O.M.
N-p50-20-35	-	-	-	44672.5	23298*	-	O.M.
N-p50-20-40	-	-	-	45864.2	31337*	-	O.M.
N-p50-20-45	-	-	-	47055.8	38099*	-	O.M.
N-p50-20-49	-	-	-	47985.7	42450*	-	O.M.
N-p50-20-50	-	-	-	48155.7	45560*	-	O.M.
N-pal19-5	329	45	90	45	10	350	AvgViol
N-pal19-10	4804	90	93	90	36	150	AvgViol
N-pal19-15	6183	107	13	114	72	48.61	AvgViol
N-pal19-18	-	-	-	114	98	-	O.M.
N-pal19-19	-	-	-	114	107	-	O.M.
N-pal43-10	-	-	-	210	40*	-	O.M.
N-pal43-15	-	-	-	315	80*	-	O.M.
N-pal43-20	-	-	-	420	139*	-	O.M.
N-pal43-25	-	-	-	525	199*	-	O.M.
N-pal43-30	-	-	-	602	276*	-	O.M.
N-pal43-35	-	-	-	602	361*	-	O.M.
N-pal43-40	-	-	-	602	459*	-	O.M.
N-pal43-42	-	-	-	602	500*	-	O.M.
N-pal43-43	-	-	-	602	535*	-	O.M.

Table 5.1: numerical experiment - "FCLOP-LR-A" - Bundle method

5.3.2 FCLOP-LR-B

Table 5.2 reports the numerical results of solving "*FCLOP-LR-B*". The same as before, bundle code of Frangioni is used to solve the dual problem. In "*FCLOP-LR-B*", the set of triangle-free inequalities are relaxed and added to the objective function multiplied by Lagrangian multipliers to penalize the objective function in the case of violation.

Comparing to the Table 5.1, the results seem much more promising. As an example, we reach the optimal solution for several instances such as **N-atp24** when cardinality number is equal to 5 or 10, also **N-atp48** when the cardinality number is equal to 10 and **N-pal19** when p is equal to 5. Also we have the optimal solution of **N-atp24**, **N-atp48** and **N-be75eec**, when p is equal to n (the LOP).

Another interesting fact is that we often get a better bound comparing to the LP bound when the cardinality is far smaller than the number of nodes. As soon as p increases, the LR dual bound significantly tends to the LP bound.

Fname	elapsed Time	bestDual Bound	nrIteration	LP bound	Optimal*	Relative Gap(%)	BundleStatus
N-atp24-5	1	18	2	41.04	18	0	AvgViol
N-atp24-10	6	53	3	82.08	53	0	AvgViol
N-atp24-15	71	98.57	32	123.12	94	4.86	AvgViol
N-atp24-20	85	146.33	100	153.16	140	4.52	MaxIter
N-atp24-23	12	168.1	100	168.25	165	1.87	MaxIter
N-atp24-24	4	172.37	100	172	172	0.21	MaxIter
N-atp48-10	44	53	4	118.54	53	0	AvgViol
N-atp48-15	883	101.66	19	177.81	97	4.80	AvgViol
N-atp48-20	11400	158.87	36	237.08	148	7.34	TimLim
N-atp48-25	11551	220.75	27	296.35	205	7.68	TimLim
N-atp48-30	10873	285.49	25	355.62	264	8.14	TimLim
N-atp48-35	10819	349.07	59	400.65	324	7.73	TimLim
N-atp48-40	2499	411.85	100	436.16	387	6.42	MaxIter
N-atp48-45	283	462.79	100	466.62	447	3.53	MaxIter
N-atp48-47	42	478.19	100	477.7	472	1.31	MaxIter
N-atp48-48	18	484.17	100	483	483	0.24	MaxIter
N-be75eec-20	119	130977	56	132966	130506	0.36	AvgViol
N-be75eec-30	24	191165	45	191168	191061	0.05	AvgViol
N-be75eec-40	97	223996	100	224006	223672	0.14	MaxIter
N-be75eec-50	19	236607	100	236464	236464	0.06	MaxIter
N-econ36-15	20	400722	55	403222	400547	0.043	AvgViol
N-econ36-20	33	469709	100	470033	469584	0.02	MaxIter
N-econ36-25	47	508110	100	507568	506232	0.37	MaxIter
N-econ36-30	29	537581	100	535017	534406	0.59	MaxIter
N-econ36-35	26	552096	100	547984	547501	0.83	MaxIter
N-econ36-36	11	552257	100	548588	548574	0.67	MaxIter
N-p50-20-10	449	3680	13	13282.6	2311*	59.23	AvgViol
N-p50-20-15	10177	7606.29	15	19923.9	5106*	48.96	TimLim
N-p50-20-20	13210	12860.9	8	26565.2	7397*	73.86	TimLim
N-p50-20-25	13728	19155.8	9	33206.5	13697*	39.85	TimLim
N-p50-20-30	11601	26532.1	13	39847.8	17052*	55.59	TimLim
N-p50-20-35	11355	34745.4	21	44672.5	23298*	49.13	TimLim
N-p50-20-40	10829	43426.3	64	45864.2	31337*	38.57	TimLim
N-p50-20-45	602	47187.7	100	47055.8	38099*	23.85	MaxIter
N-p50-20-49	46	48115.1	100	47985.7	42450*	13.34	MaxIter
N-p50-20-50	19	48318.4	100	48155.7	45560*	6.05	MaxIter
N-pal19-5	0	10	3	45	10	0	AvgViol
N-pal19-10	0	45	3	90	36	25	AvgViol
N-pal19-15	3	105	13	114	72	45.83	AvgViol
N-pal19-18	8	114	74	114	98	16.32	AvgViol
N-pal19-19	0	114	4	114	107	6.54	AvgViol
N-pal43-10	1	45	3	210	40*	12.5	AvgViol
N-pal43-15	2	105	3	315	80*	31.25	AvgViol
N-pal43-20	1	190	3	420	139*	36.69	AvgViol
N-pal43-25	24	300	4	525	199*	50.75	AvgViol
N-pal43-30	76	435	100	602	276*	57.60	MaxIter
N-pal43-35	79	595.03	100	602	361*	64.82	MaxIter
N-pal43-40	40	602	100	602	459*	31.15	MaxIter
N-pal43-42	31	602	93	602	500*	20.4	AvgViol
N-pal43-43	0	602	4	602	535*	12.52	AvgViol

Table 5.2: numerical experiment for "*FCLOP-LR-B*"

(* Optimal or the best known feasible solution in the case if the optimum value is not known)

5.3.3 FCLOP-LR-C

Table 5.3 presents the numerical results concerning the "*FCLOP-LR-C*" model solved by Bundle code of Frangioni. In this relaxation strategy, the two constraints (5.10) and (5.14) that restrict the number of non-zero y variables and the number of non-zero x variables in any feasible solution are relaxed.

Table 5.3 shows that for all the instances that was not ran out of memory, the algorithm could find a very high quality dual bound. However there are many instances that have been terminated due to the memory issue.

In general, this model gives us the best result (regarding the relative gap) comparing with the other relaxations we have tried for easier instances while the *FCLOP-LR-C* provide better dual bound for the instances that are more difficult e.g. N-atp48 with different cardinality values.

Fname	elapsed Time	bestDual Bound	nrIteration	Optimal*	Relative Gap(%)	BundleStatus
N-atp24-5	45	18	12	18	0	AvgViol
N-atp24-10	27	53	9	53	0	AvgViol
N-atp24-15	71	94	13	94	0	AvgViol
N-atp24-20	20	140	7	140	0	AvgViol
N-atp24-23	12	165	6	165	0	AvgViol
N-atp24-24	0	172	2	172	0	AvgViol
N-atp48-10	-	-	-	53	-	O.M.
N-atp48-15	11903	124.53	4	97	28.38	AvgViol
N-atp48-20	-	-	-	148	-	O.M.
N-atp48-25	-	-	-	205	-	O.M.
N-atp48-30	-	-	-	264	-	O.M.
N-atp48-35	-	-	-	324	-	O.M.
N-atp48-40	12882	415.51	4	387	7.36	AvgViol
N-atp48-45	11101	482.17	4	447	7.86	AvgViol
N-atp48-47	-	-	-	472	-	O.M.
N-atp48-48	60	483	2	483	0	AvgViol
N-be75eec-20	24	130506	6	130506	0	AvgViol
N-be75eec-30	5	191061	4	191061	0	AvgViol
N-be75eec-40	13	223672	8	223672	0	AvgViol
N-be75eec-50	2	236464	2	236464	0	AvgViol
N-econ36-15	21	401383	11	400547	0.2	AvgViol
N-econ36-20	16	469584	4	469584	0	AvgViol
N-econ36-25	22	506268	7	506232	0.007	AvgViol
N-econ36-30	23	534406	7	534406	0	AvgViol
N-econ36-35	19	547542	4	547501	0.007	AvgViol
N-econ36-36	11	548574	2	548574	0	AvgViol
N-p50-20-10	-	-	-	2311*	-	O.M.
N-p50-20-15	-	-	-	5106*	-	O.M.
N-p50-20-20	-	-	-	7397*	-	O.M.
N-p50-20-25	-	-	-	13697*	-	O.M.
N-p50-20-30	-	-	-	17052*	-	O.M.
N-p50-20-35	-	-	-	23298*	-	O.M.
N-p50-20-40	-	-	-	31337*	-	O.M.
N-p50-20-45	-	-	-	38099*	-	O.M.
N-p50-20-49	-	-	-	42450*	-	O.M.
N-p50-20-50	-	-	-	45560*	-	O.M.
N-pal19-5	2349	10	5	10	0	AvgViol
N-pal19-10	2580	36	4	36	0	AvgViol
N-pal19-15	10947	72.19	8	72	0.26	AvgViol
N-pal19-18	7040	98	6	98	0	AvgViol
N-pal19-19	680	107	2	107	0	AvgViol
N-pal43-10	-	-	-	40*	-	O.M.
N-pal43-15	-	-	-	80*	-	O.M.
N-pal43-20	-	-	-	139*	-	O.M.
N-pal43-25	-	-	-	199*	-	O.M.
N-pal43-30	-	-	-	276*	-	O.M.
N-pal43-35	-	-	-	361*	-	O.M.
N-pal43-40	-	-	-	459*	-	O.M.
N-pal43-42	-	-	-	500*	-	O.M.
N-pal43-43	-	-	-	535*	-	O.M.

Table 5.3: numerical experiment for "FCLP-LR-C"

(* Optimal or the best known feasible solution in the case if the optimum value is not known)

5.3.4 FCLOP-LD1

Table 5.4 reports the computational results of applying Bundle method to solve the *FCLOP-LD1* model when we decompose the FCLOP to two sub-problems $LD_u(x,y)$ and $LD_u(t)$.

Regarding the numerical results, *FCLOP-LD1* produces relatively very high quality dual bounds. One may observe that, as an example, for the instance *N-pal43*, for the first time the algorithm terminates after finishing 100 iterations without facing with a memory issue. It is quite promising that we could improve dual bound of some difficult instances. However, it may not be the best choice to deal with easier instances since one sees that even very easy instances (e.g. *N-atp24*, *N-be75eec* and *N-econ36*) had difficulties to be terminated after being sufficiently converged.

Fname	elapsed Time	bestDual Bound	nrIteration	LP bound	Optimal*	Relative Gap(%)	BundleStatus
N_atp24_5	1	18	3	41.04	18	0	AvgViol
N_atp24_10	84	53	100	82.08	53	0	MaxIter
N_atp24_15	108	98.84	100	123.12	94	5.14	MaxIter
N_atp24_20	49	147.31	100	153.16	140	5.22	MaxIter
N_atp24_23	32	168.42	100	168.25	165	2.07	MaxIter
N_atp24_24	29	172.37	100	172	172	0.21	MaxIter
N_atp48_10	821	53	68	118.54	53	0	AvgViol
N_atp48_15	5880	101.83	100	177.81	97	4.97	MaxIter
N_atp48_20	10848	159.99	49	237.08	148	8.10	TimLim
N_atp48_25	11197	224.87	30	296.35	205	9.69	TimLim
N_atp48_30	11430	291.03	36	355.62	264	10.23	TimLim
N_atp48_35	10903	352.28	64	400.65	324	8.72	TimLim
N_atp48_40	3796	414.21	100	436.16	387	7.03	MaxIter
N_atp48_45	2480	464.23	100	466.62	447	3.85	MaxIter
N_atp48_47	3136	481.46	100	477.7	472	2.04	MaxIter
N_atp48_48	3081	486.63	100	483	483	0.75	MaxIter
N_be75eec_20	111	131020	100	132966	130506	0.39	MaxIter
N_be75eec_30	147	191208	100	191168	191061	0.07	MaxIter
N_be75eec_40	704	224659	100	224006	223672	0.44	MaxIter
N_be75eec_50	1753	237698	100	236464	236464	0.52	MaxIter
N_econ36_15	28	401011	100	403222	400547	0.11	MaxIter
N_econ36_20	30	471739	100	470033	469584	0.45	MaxIter
N_econ36_25	65	512272	100	507568	506232	1.19	MaxIter
N_econ36_30	150	539958	100	535017	534406	1.03	MaxIter
N_econ36_35	237	554836	100	547984	547501	1.33	MaxIter
N_econ36_36	243	556363	100	548588	548574	1.41	MaxIter
N-p50-20_10	-	-	-	13282.6	2311*	-	O.M.
N-p50-20_15	-	-	-	19923.9	5106*	-	O.M.
N-p50-20_20	-	-	-	26565.2	7397*	-	O.M.
N-p50-20_25	-	-	-	33206.5	13697*	-	O.M.
N-p50-20_30	-	-	-	39847.8	17052*	-	O.M.
N-p50-20_35	-	-	-	44672.5	23298*	-	O.M.
N-p50-20_40	-	-	-	45864.2	31337*	-	O.M.
N-p50-20_45	-	-	-	47055.8	38099*	-	O.M.
N-p50-20_49	-	-	-	47985.7	42450*	-	O.M.
N-p50-20_50	-	-	-	48155.7	45560*	-	O.M.
N_pal19_5	10	10	100	45	10	0	MaxIter
N_pal19_10	11	45	100	90	36	25	MaxIter
N_pal19_15	14	105	100	114	72	45.83	MaxIter
N_pal19_18	29	114.29	100	114	98	16.62	MaxIter
N_pal19_19	11	114	100	114	107	6.54	MaxIter
N_pal43_10	39	45	100	210	40*	12.5	MaxIter
N_pal43_15	68	105.01	100	315	80*	31.26	MaxIter
N_pal43_20	181	190	100	420	139*	36.69	MaxIter
N_pal43_25	108	300.02	100	525	199*	50.76	MaxIter
N_pal43_30	163	435.01	100	602	276*	57.61	MaxIter
N_pal43_35	1390	595.01	100	602	361*	64.82	MaxIter
N_pal43_40	6000	604.15	100	602	459*	31.62	MaxIter
N_pal43_42	8081	603.94	100	602	500*	20.78	MaxIter
N_pal43_43	8001	602.14	100	602	535*	12.54	MaxIter

Table 5.4: Numerical results of lagrangian decomposition model - *FCLOP*–*LD1*

5.3.5 FCLOP-LD2

Table 5.5 reports the computational results of applying Bundle method to solve FCLOP-LD2.

One observes that here for the first time we could improved the LP bound for some of the very difficult instances (e.g. *N-p50-20*).

As it is reported in Table 5.5, the decomposition scheme FCLOP-LD2 works relatively better for easy instances as well as very difficult instances. Relatively short elapsed time for solving each iteration is another advantage of this decomposition scheme, and particularly short CPU time for instances that terminated by MaxIter termination condition confirms it.

In general, one may say that *FCLOP-LD2* is an efficient decomposition scheme to deal with highly difficult instances regarding the three critical items; the dual bound quality, the number of iterations and the CPU time.

Fname	elapsed Time	bestDual Bound	nrIteration	LP bound	Optimal*	Relative Gap(%)	BundleStatus
N-atp24-5	19	18	22	41.04	18	0	AvgViol
N-atp24-10	52	53	51	82.08	53	0	AvgViol
N-atp24-15	299	98.91	100	123.12	94	5.22	MaxIter
N-atp24-20	115	147.17	100	153.16	140	5.12	MaxIter
N-atp24-23	52	168.79	100	168.25	165	2.29	MaxIter
N-atp24-24	31	172.54	100	172	172	0.31	MaxIter
N-atp48-10	578	53	36	118.54	53	0	AvgViol
N-atp48-15	10842	101.89	94	177.81	97	5.04	TimLim
N-atp48-20	10923	160.25	39	237.08	148	8.27	TimLim
N-atp48-25	11208	225.21	35	296.35	205	9.85	TimLim
N-atp48-30	11055	290.08	37	355.62	264	9.87	TimLim
N-atp48-35	10904	353.61	62	400.65	324	9.13	TimLim
N-atp48-40	8508	414.41	100	436.16	387	7.08	MaxIter
N-atp48-45	5253	468.04	100	466.62	447	4.70	MaxIter
N-atp48-47	5970	481.34	100	477.7	472	1.97	MaxIter
N-atp48-48	7533	490.05	100	483	483	1.45	MaxIter
N-be75eec-20	10	132655	6	132966	130506	1.64	AvgViol
N-be75eec-30	133	191138	67	191168	191061	0.04	AvgViol
N-be75eec-40	1119	224079	77	224006	223672	0.18	AvgViol
N-be75eec-50	2567	238646	100	236464	236464	0.92	MaxIter
N-econ36-15	1	401589	3	403222	400547	0.26	AvgViol
N-econ36-20	1	470082	3	470033	469584	0.10	AvgViol
N-econ36-25	1	514151	3	507568	506232	1.56	AvgViol
N-econ36-30	2	544975	4	535017	534406	1.97	AvgViol
N-econ36-35	6	560517	5	547984	547501	2.37	AvgViol
N-econ36-36	7	561570	5	548588	548574	2.36	AvgViol
N-p50-20-10	2750	3683.25	45	13282.6	2311*	59.37	AvgViol
N-p50-20-15	10878	7667.97	26	19923.9	5106*	50.17	TimLim
N-p50-20-20	11406	12927	8	26565.2	7397*	74.76	TimLim
N-p50-20-25	10872	19294.5	8	33206.5	13697*	40.86	TimLim
N-p50-20-30	11321	26651.7	18	39847.8	17052*	56.29	TimLim
N-p50-20-35	11565	34895.7	30	44672.5	23298*	49.77	TimLim
N-p50-20_40	15748	45161.3	4	45864.2	31337*	44.11	TimLim
N-p50-20_45	18510	55571.6	2	47055.8	38099*	45.86	TimLim
N-p50-20_49	18640	64218.4	2	47985.7	42450*	51.28	TimLim
N-p50-20_50	11555	66378	2	48155.7	45560*	45.69	TimLim
N-pal19-5	1	10	11	45	10	0	AvgViol
N-pal19-10	3	45	16	90	36	25	AvgViol
N-pal19-15	7	105	45	114	72	45.83	AvgViol
N-pal19-18	39	114.17	100	114	98	16.5	MaxIter
N-pal19-19	11	114.11	58	114	107	6.64	AvgViol
N-pal43-10	5	45	15	210	40*	12.5	AvgViol
N-pal43-15	11	105	18	315	80*	31.25	AvgViol
N-pal43-20	591	190	17	420	139*	36.69	AvgViol
N-pal43-25	150	300	19	525	199*	50.75	AvgViol
N-pal43-30	150	435.01	100	602	276*	57.61	MaxIter
N-pal43-35	1343	595.03	100	602	361*	64.82	MaxIter
N-pal43-40	8744	604.7	100	602	459*	31.74	MaxIter
N-pal43-42	10872	610.31	88	602	500*	22.06	TimLim
N-pal43-43	8103	602.55	63	602	535*	12.62	AvgViol

Table 5.5: Numerical results of solving lagrangian decomposition model FCLOP-LD2

In the following, several diagrams are presented to analyse the behavior of different lagrangian relaxations on instances of our test-bed.

Every figure contains two bar charts corresponding to 1) the number of iterations per instance and 2) percentage of the relative gap resulted by the corresponding relaxation model.

The Figure 5.6 confirms that $N-atp24$ for any cardinality number, is relatively an easy instance. The diagram concerning the number of iterations show that all the instances are terminated before the time limit reached.

The second part of the Figure 5.6 concerns the relative dual gap and shows that except for the $LR-FCLOP-A$ relaxation, all the other relaxation strategies produce relatively high quality dual bound that result zero or very small deviation from the optimal solution. Among all the 5 relaxations that we have applied, the $LR-FCLOP-A$ relaxation strategy reached to the optimal solution for different p values in a small number of iterations.

In general one may conclude that $N-atp24$ is relatively an easy to solve instance for different p values.

It is interesting to note that even in such an easy instance, we still have a larger deviation from the optimal solution whenever the cardinality number p is an median value (is far from n and far from zero).

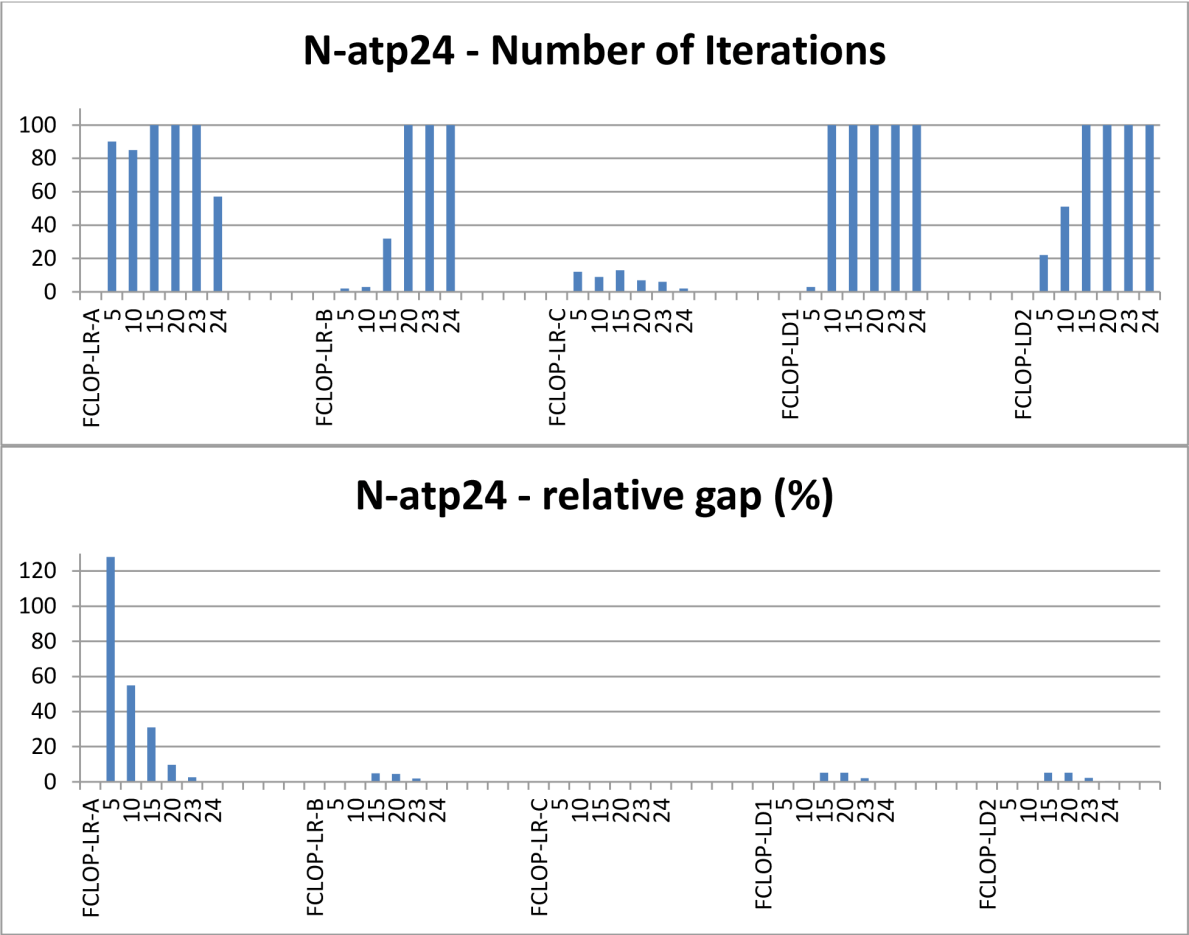


Figure 5.6: N-atp24 FCLOP instances when p varies

In Figure 5.7 there are some dashed bars (e.g. in *FCLOP-LR-A* when p is in $\{40, 45, 47, 48\}$), which precise that the procedure for these instances has been terminated by the TimLine stopping criteria(1800 seconds).

There are also instances having no bar associated to them (e.g. The *FCLOP-LR-C* when p is in $\{10, 20, 25, 30, 35, 47\}$).

Figure 5.7 shows that in contrary to the instance *N-atp24*, the *FCLOP-LR-C* relaxation strategy is not sufficiently strong to solve the *N-atp48*. Table (5.3) confirms that from among 10 different p values, *FCLOP-LR-C* could solve only four instances.

Another point to address in Figure 5.7 is that the instance *N-atp48* is more difficult comparing to the *N-atp24* since in almost all the cases, the algorithm terminated either by the time limit termination condition, or after reaching the maximum number of iterations or because of the memory issue. The cases when $15 \leq p \leq 40$, are quite difficult and the runs were mostly terminated in an unusual way. However we still were able to reach very high quality dual bounds by applying *FCLOP-LR-B* relaxation strategy and *FCLOP-LD2* Lagrangian decomposition.

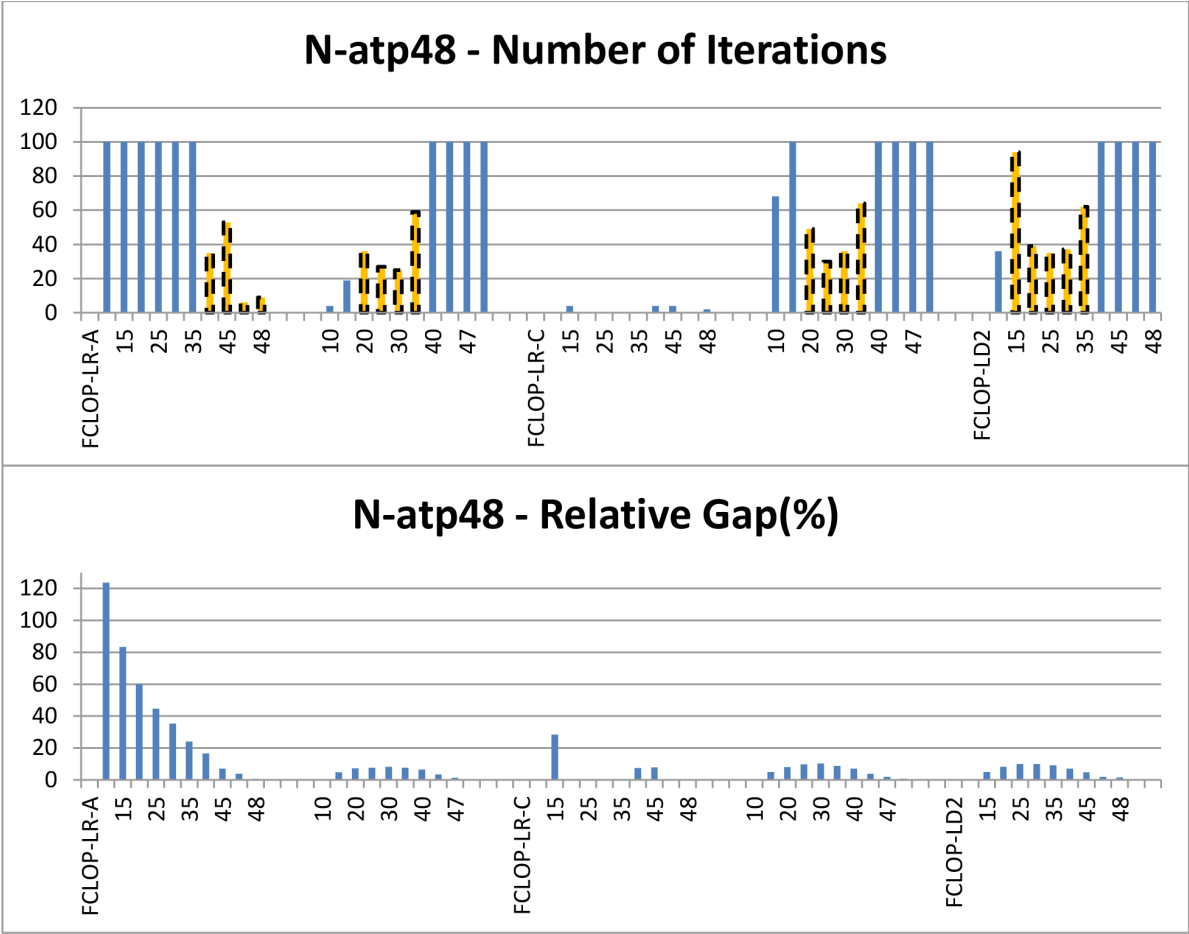


Figure 5.7: N-atp48 FCLOP instances when p varies

Figure 5.8 reminds that the instance *N-be75eec* is relatively easier to solve compared to the *N-atp48*. We have 50 vertices in this instance and cardinality number takes the following values: $\{20, 30, 40, 50\}$. The *FCLOP-LR-A* and *FCLOP-LD1* relaxation strategies have been always terminated by the MaxIter termination criteria. In fact, none of the instances has been terminated neither by the TimLim stopping criteria (10800 seconds) nor because of the memory issue.

As an other relatively easy to solve instance, the *FCLOP-LR-C* is able to reached to the optimal solution, for different p values.

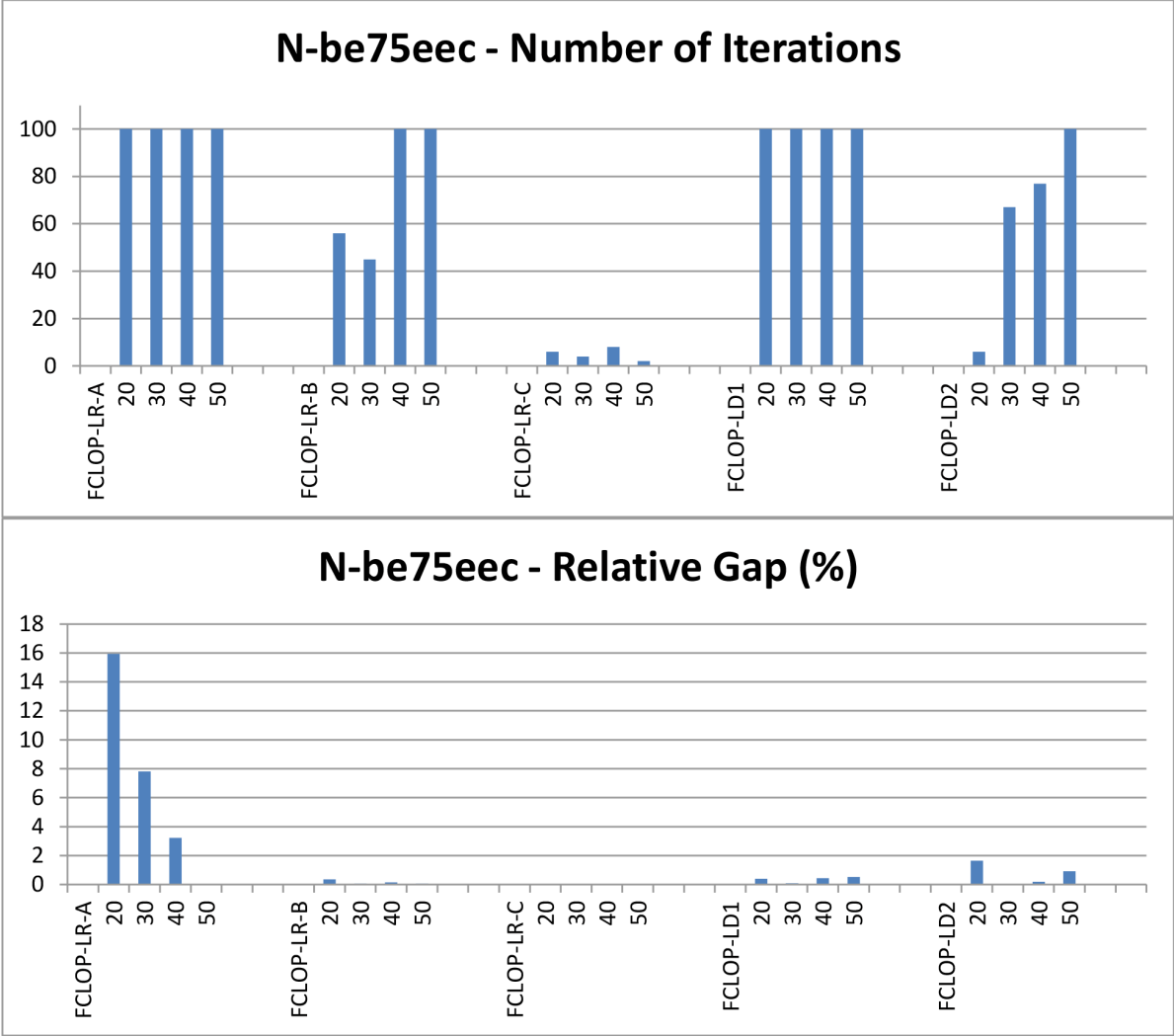


Figure 5.8: N-be75eec FCLOP instances when p varies

By analysing the Figure 5.9 together with Table 5.3, one observes that the *FCLOP-LR-C* relaxation strategy produce very high quality dual bounds for the FCLOP problem. Considering all the three criteria, number of iterations, dual gap quality and the CPU time, emphasize that the lagrangian decomposition scheme with three subproblems is the most qualified algorithm to solve this instance. In analysing this figure, it is important to keep in mind that the *N-econ36* is categorized as a relatively easy instance and as it is mentioned earlier, for easier instances, *FCLOP-LR-C* is a good choice.

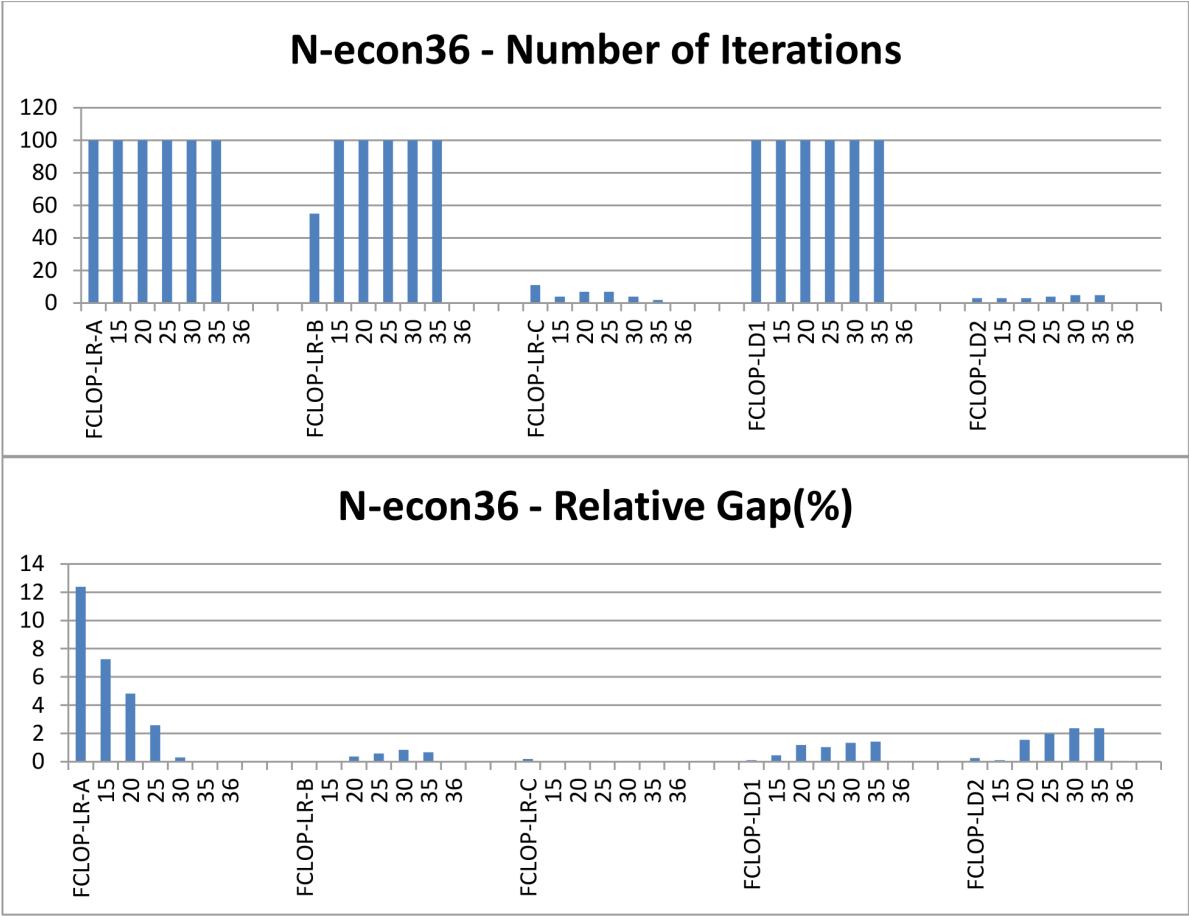


Figure 5.9: N-econ36 FCLOP instances when p varies

From Figure 5.10 together with the Tables Table 5.1 - Table 5.5, one observes that only solution algorithms, FCLOP-LR-B and FCLOP-LD2 were able to deal with these instances and all the other relaxation strategies were facing with the memory issue. Figure 5.10 confirms that several instances are terminated by TimLim termination status (after 10800 seconds). *N-p50-20* is one of the very difficult instances that its optimal value (in the case of the LOP) is not known in the literature.

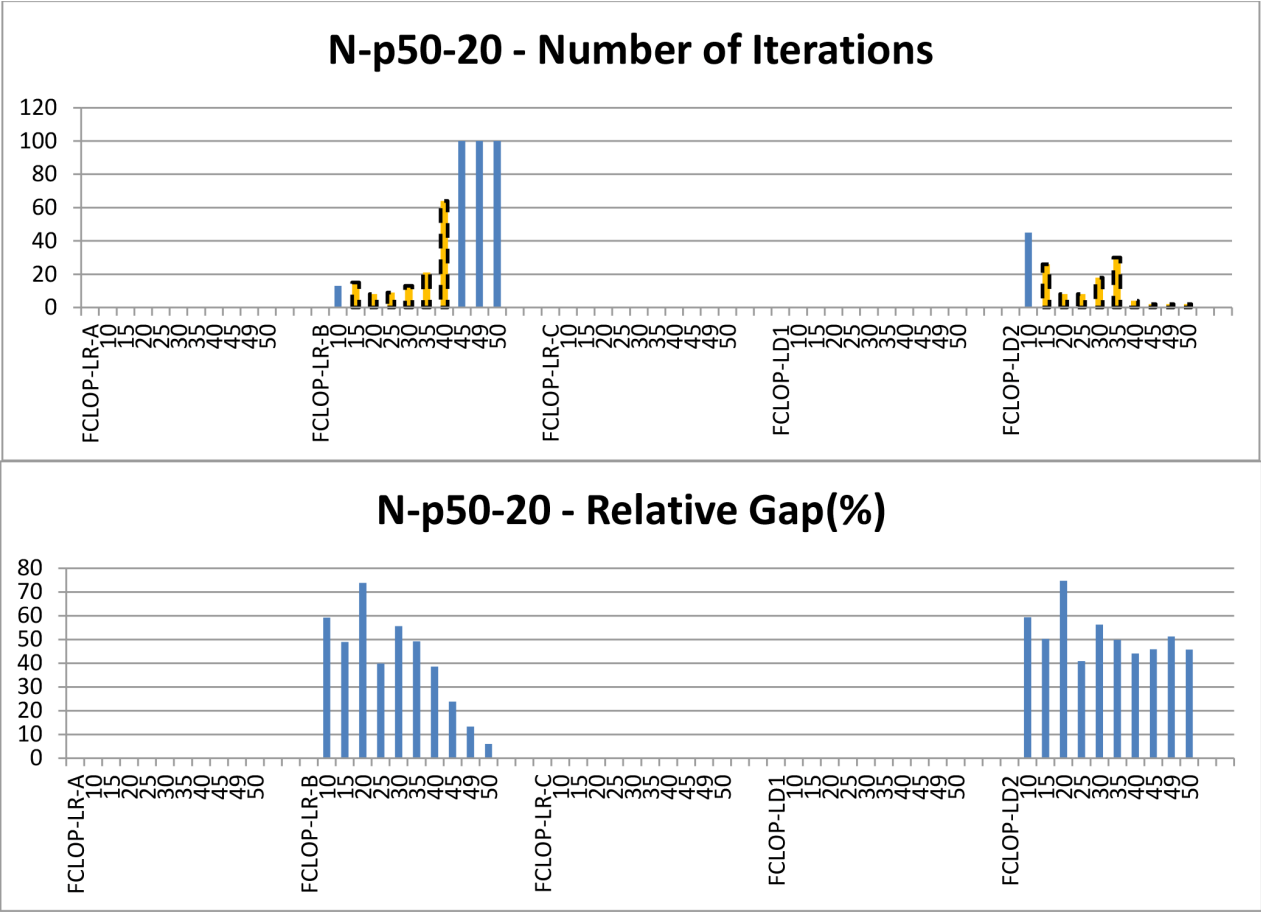


Figure 5.10: N-p50-20 FCLOP instances when p varies

The two instances **N-pal19** and **N-pal43** have more or less the same structure. The main difference between **N-pal19** and **N-pal43** is in their size (total number of nodes). The instances in this class become quickly very difficult as soon as the number of nodes increase due to the high level of symmetry inherent in them. As an example, **N-pal19** is a median instance containing 19 nodes while **N-pal43** is a very difficult instance containing 43 nodes. Comparing Figure 5.11 and Figure 5.12 reveals that for these difficult instances, one may prefer to apply the Lagrangian decomposition with three subproblems to obtain a good quality dual bound in a relatively short CPU time.

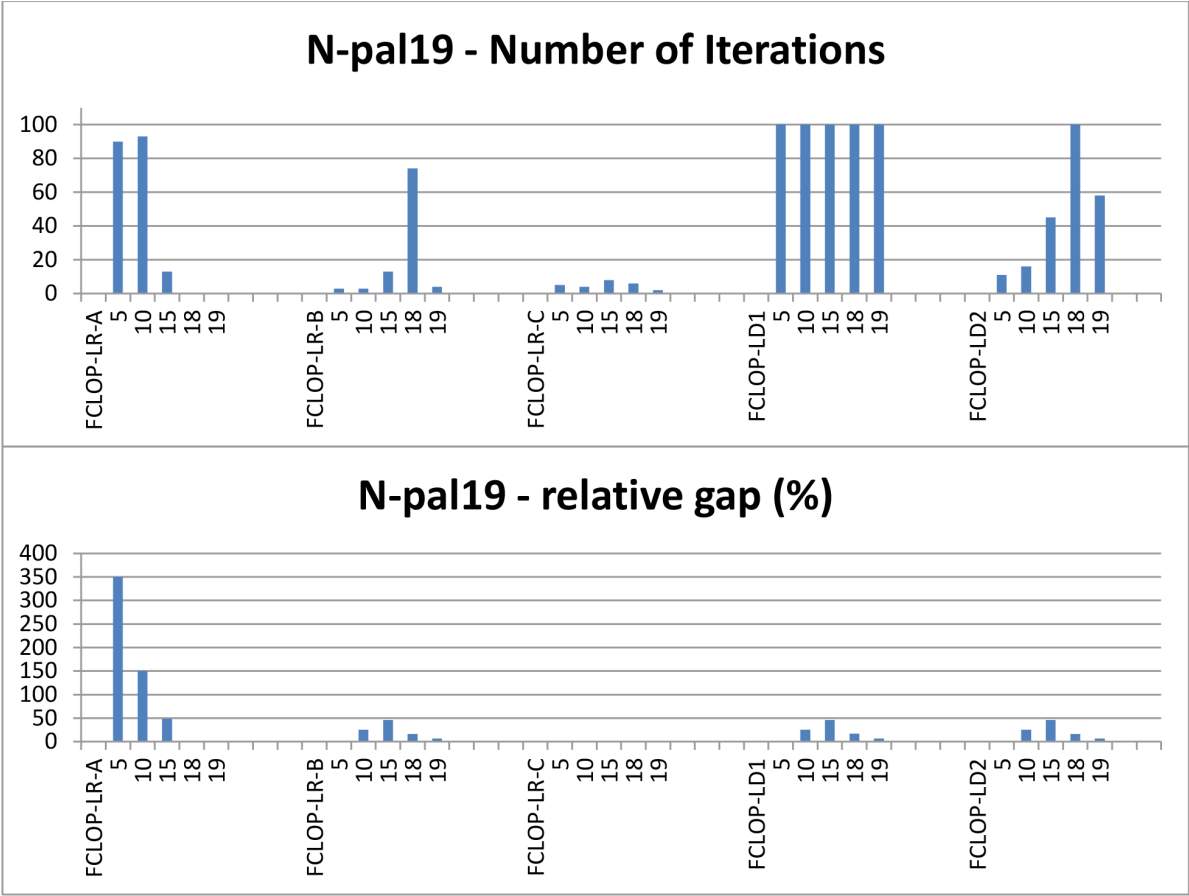


Figure 5.11: N-pal19 FCLOP instances when p varies

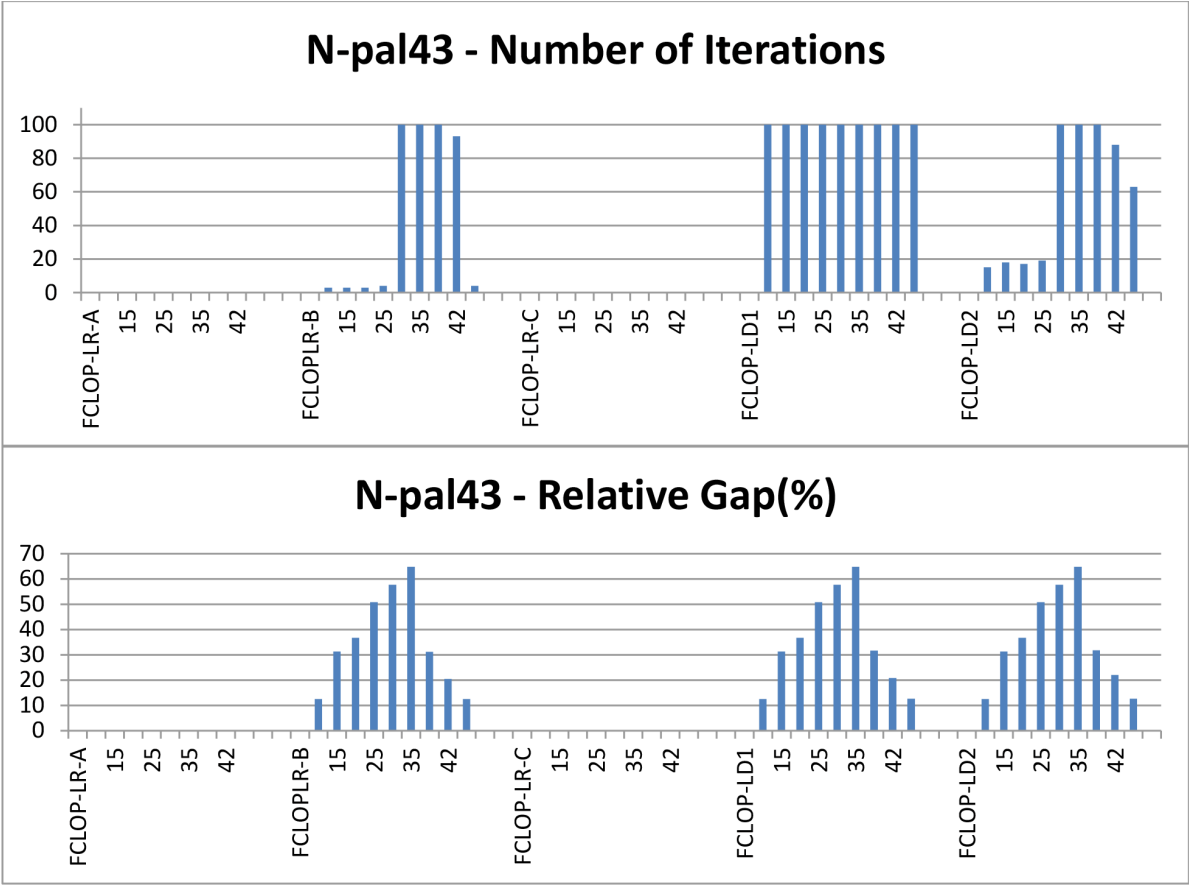


Figure 5.12: N-pal43 FCLOP instances when p varies

CHAPTER 6

Relaxation Based Exact Solution Method

In this chapter, we present a kind of cutting plane algorithm to reduce the number of constraints in the FCLOP integer programming formulation. Grötschel et al. (1985a) shows that triangle-free inequalities are defining facets for the LOP polytope. However, our studies show that this is not the case for the FCLOP polytope. We have shown that even after strengthening the triangle-free inequalities, the resulted inequalities,

$$x_{ij} + x_{jk} + x_{ki} \leq y_s + y_t \quad \forall \quad \{s, t\} \subset \{i, j, k\}, \quad \{i, j, k\} \subseteq V_n$$

do not define facets for the FCLOP polytope.

Numerical experiments show that this set of inequalities are very heavy ($\mathcal{O}(n^3)$) and negatively impact performance of general-purpose solvers. On the other hand, as p reduces, there is a huge number of these inequalities that will never become binding. In fact, they become redundant whenever at least one (or more) of the three vertices i, j and k is (are) not selected to be in the order—which is likely to happen when the gap between n and p is large enough.

The idea in here is to relax the set of triangle-free inequalities and add them to the model in the case of necessity. To do so we need to fix the p active vertices and then find the eventual violations and add the corresponding cuts to the relaxed model to eliminate the violations. For doing so, we have applied the idea of Combinatorial Benders Cuts to create the cuts in an iterative algorithm.

Benders decomposition (Benders, 1962, 2005) is a variable partitioning method known as an efficient decomposition method in MIPs and also in scenario-based stochastic programming (see e.g. Mulvey and Ruszczyński (1995);

Birge (1985); Birge and Louveaux (1988); Pereira and Pinto (1991)).

Benders decomposition was originally proposed for the mixed integer programming models aiming at partitioning the variables into two parts with respect to the types of variables and relaxing the set of *complicating variables*. Often, in MIPs the set of complicating variables are integer variables and the remaining ones are assumed as *easy-to-handle* ones (mainly because today's LP solvers are very efficient).

In the classical Benders decomposition method, the convexity of the sub-problem plays a very crucial role. The primal-dual relationship, which is exploited for generating Benders cuts (to be added to the Master Problem in an iterative matter) is resulted from the convexity of the sub-problem.

Later on, perhaps the stochastic programming community were the first place where the convexity of sub-problem was challenged — see Carøe and Tind (1998); van der Vlerk (1995); Schultz et al. (1998); Ahmed et al. (2002); Louveaux and van der Vlerk (1993). There were situations as in the case of integer recourse where the sub-problem was integer and no primal-dual relationship was available. Some authors has generic view at such problems while others often develop some algorithms tailored for certain applications. As an example of generic method, one can refer to Sherali and Fraticelli (2002).

In most of the aforementioned contributions, the particular interest in using Benders decomposition method was that either the sub-problem can be decomposed for every scenario or there was an efficient algorithm to solve the sub-problem while it was difficult to solve the whole at once.

However, there were certain cases where the sub-problem was not necessarily decomposable, no particular efficient algorithm (other than standard MIP solvers) is available or the original model was a Big-M model for a combinatorial optimization problem, which caused numerical instability during resolution.

The *Combinatorial Benders Cuts* first appeared in Codato and Fischetti (2006) for a Big-M model of Asymmetric Traveling Salesman Problem and very promising results were reported. In this approach the sub-problem was a feasibility problem rather than an optimization problem. The problem after removing some of the constraints was called the Master Problem while the sub-problem was composed of a set of constraints, which could be violated for the solutions produced by the master problem. The algorithm then seeks a *minimal infeasible subsystem* (more precisely Irreducible Infeasible Subsystem) to be used in generating cover-like constraints added to the master problem in order to cut off the current solution of the master

problem.

The method is later on extended to accommodate MIPs as well. Interested readers are referred to Codato and Fischetti (2006) for the relevant details.

Later Bai and Rubin (2009) proposed a combinatorial Benders cuts for the minimum tollbooth problem and Cao et al. (2010) for an integrated yard truck and yard crane scheduling problem.

This method is particularly proposed to solve the Big-M models. However, it led us to separate a part of constraints (particularly triangle-free inequalities) and solve the relaxed model, which becomes significantly easier.

In our algorithm we have relaxed the set of triangle-free inequalities. Then to avoid having the solutions which are infeasible for the initial problem, we control all the feasible solutions of the relaxed problem which are going to be installed as an incumbent in the Branch-and-Bound tree. If any triangle become recognized, then the corresponding triangle-free inequality(ies) will be added to the relaxed model to make the solution(s), infeasible. These separated cuts are adding globally and permanently and it ensures the algorithm to be finite in a polynomial time of iterations. It must be noted that in contrast with the classical branch-and-cut approaches where usually we emphasize on separating cuts at the root node and also for fractional solutions, we only separate such cuts when an incumbent is going to be installed.

In this chapter, after introducing the general concept, we will propose some algorithm for dealing with certain formulation of the FCLOP. A hybridized technique will be also presented that combines the Lagrangian relaxation method (solved by Bundle method Frangioni (2000)) with the relaxation based algorithm that we are presenting here. Computational results are presented at the end of each algorithm.

Benders decomposition

Let problem P be given as follows:

(P)

$$\min c'_1x + c'_2y \quad (6.1)$$

s.t.

$$Ax \geq a \quad (6.2)$$

$$By \geq b \quad (6.3)$$

$$Dx + Ey \geq d \quad (6.4)$$

$$x \in \mathbb{Z}^m \quad (6.5)$$

$$y \in \mathbb{R}^n \quad (6.6)$$

where $x \in \mathbb{Z}^m$ are assumed to be the *complicating variables*. Benders decomposition method tries to decompose the problem as a Master Problem (MP) and a sub-problem dual (SPD) as follows:

(MP)

$$\min c'_1 x + \eta \quad (6.7)$$

s.t.

$$Ax \geq a \quad (6.8)$$

$$\eta \geq ub + v(d - Dx) \quad \forall \text{ extreme points} \quad (6.9)$$

$$0 \geq ub + v(d - Dx) \quad \forall \text{ extreme rays} \quad (6.10)$$

$$x \in \mathbb{Z}^m, \eta \geq 0 \quad (6.11)$$

(SPD)

$$\max ub + v(d - D\bar{x}) \quad (6.12)$$

s.t.

$$uB + vE = c_2 \quad (6.13)$$

$$u_i, v_i \in \mathbb{R}^+ \quad (6.14)$$

In the first iteration, the set of optimality cuts (6.9) and feasibility cuts (6.10) are empty. A solution of MP is given to the SPD (note that the feasible region of SPD is independent of x). If for such a given x , SPD solves to optimality, then an optimality cut of form (6.9) and if otherwise a feasibility cut of form (6.10) will be added to the MP.

A solution to the MP is always a lower bound on the optimal solution of (P) (an upper bound in the case of maximization). Also, if for a given x , SPD solves to optimality, then an upper bound for the whole problem also becomes known.

The classical implementation of method exploits the decomposable structure of problems, identifies a set of complicating variables (or the constraints involving those variables) to put in a Master Problem (MP) and a Sub-problem (SP) which, is convex (often a linear program) such that the primal/dual relationship holds. The algorithm iterates between solving the master problem to optimality followed by solving the sub-problem where the complicating variables became parameters. The method makes use of exchange of information in terms of cuts between these two smaller problems to attain optimality of the original model.

However, when dealing with pure a integer programming sub-problem, i.e. not a convex one, the primal-dual relationship cannot be exploited to generate cuts in the sub-problem LP and add them to the master problem to improve the outer approximation. Rather, at every iteration, solution to the master problem is used to either finding the minimal infeasible subsystem or a set of constraints generated on the fly, to add to the MP in order to cut off the proposed infeasible solution (see *Combinatorial Benders (CB)* in Fischetti et al. (2009)).

This gave us the idea to generate an algorithm to separate useful relaxed cuts after relaxing the FCLOP model.

The separation algorithm can be described as in the sequel.

Separation algorithm

Let P be the following problem:

(P)

$$\min c_1'x \quad (6.15)$$

s.t.

$$Ax \geq a \quad (6.16)$$

$$Bx \geq b \quad (6.17)$$

$$x \in \mathbb{Z}^m \quad (6.18)$$

Definition 3 (Irreducible Infeasible Subsystem (IIS)) *A subsystem of inequalities and equations such that the subsystem is inconsistent (infeasible) and every proper subsystem is consistent (feasible).*

We form a master problem (MP) as follows:

(MP)

$$\min c_1'x \quad (6.19)$$

s.t.

$$Ax \geq a \quad (6.20)$$

$$x \in \mathbb{Z}^m \quad (6.21)$$

The sub-problem is defined as:

(SP)

$$\min 0 \quad (6.22)$$

s.t.

$$Bx \geq b \quad (6.23)$$

$$x \in \mathbb{Z}^m \quad (6.24)$$

which is a feasibility problem as the objective is void.
The separation algorithm is described in algorithm 1.

Algorithm 1: Separation algorithm.

```

1  MP'=MP;
2  optim :=false;
3  repeat
4       $\hat{x} := \text{solve (MP')}$ ;
5      status := solve (SP( $\hat{x}$ )) ;
6      if status = infeasible then
7          IISconst := IIS(SP( $\hat{x}$ ));
8          add IISconst to MP';
9      else
10          $x^* := \hat{x}$ ;
11         optim := true;
12     end
13 until optim=true;
```

6.1 Relaxation Based Exact Algorithm for the FCLOP

Our extensive computational experiments with the instances of the FCLOP have shown that the prohibitive number of classical triangle-free inequalities,

$$x_{ij} + x_{jk} + x_{ki} \leq 2, \quad \forall i, j, k \in V_n, i \neq j \neq k \quad (6.25)$$

are a major bottleneck in solving instances of the FCLOP.

In fact, we have observed that only a very small fraction (often below 1 percent) of the whole set of inequalities in (6.25) are binding at the optimal solution (in general for any feasible solution). Consequently, their existence in the set of constraints negatively affect the performance of general-purpose solvers mainly due to the degeneracy (cause by having too many similar rows in the simplex tableau) in addition to the symmetry in the structure of the FCLOP.

Let us recall the FCLOP model here:

FCLOP

$$\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \quad (6.26)$$

s.t.

$$x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V_n, i < j \quad (6.27)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V_n, i < j \quad (6.28)$$

$$x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall i, j \in V_n, i < j \quad (6.29)$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall i, j, k \in V_n \quad (6.30)$$

$$\sum_{i \in V_n} y_i = p \quad (6.31)$$

$$y_i, x_{ij} \in \{0, 1\} \quad \forall i, j \in V_n, i \neq j \quad (6.32)$$

Theorem 9 *The total number of effective constraints in the constraints set (6.30) in a FCLOP model is independent of n . In fact, it only depends on the cardinality, p , and is calculated as follows: $2 \binom{p}{3}$ while the total number of the set of (6.30) constraints is equal to $\frac{2}{3} \binom{n}{3}$.*

Figure 6.1 depicts the ratio of the effective constraints to the number of (6.30) constraints.

Example 2 *For $n = 35$, we have:*

1. *when $p = 5$ only 0.15 percent of constraints (6.30) are effective,*
2. *for $p = 10$ it was 1.88 percent, and,*
3. *for $p = 25$ only 35.14 percent of the whole constraints are necessary.*

Example 3 *For $n = 75$, we have:*

1. *when $p = 10$ only 0.17 percent are effective,*
2. *for $p = 20$, 6.01 percent, and,*
3. *for $p = 60$ only 50.67 percent of the whole constraints are necessary.*

Remark 6 *Obviously, when $p = n$, all the constraints are necessary as is the case in the LOP.*

For strengthening the formulation, yet we propose to replace the classical triangle-free inequalities(6.30) with:

$$x_{ij} + x_{jk} + x_{ki} \leq y_s + y_t \quad \forall \{s, t\} \subset \{i, j, k\} \subseteq V_n \quad (6.33)$$

It is shown in Table 3.4 reported in chapter 3 that replacing (6.33) with the classical triangle-free inequalities (6.30) will slightly improve the LP bound. However increasing the number of inequalities in (6.33) comparing to (6.30) had negative impact on the calculation time.

Thats why we have chosen to relax (6.33) and add them upon need in this relaxation based solution algorithm.

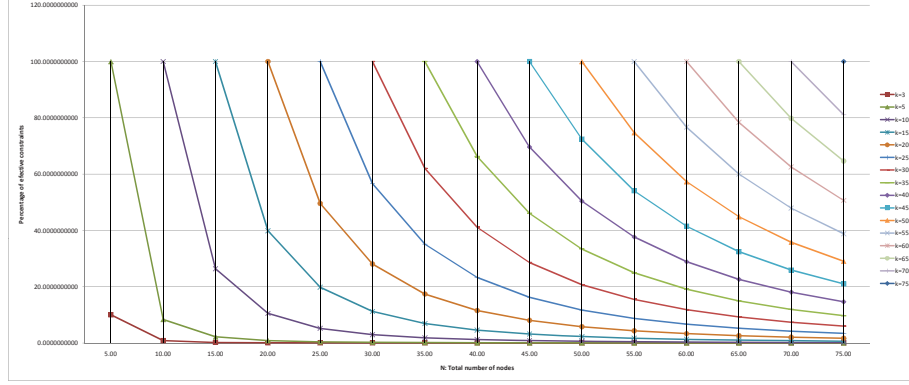


Figure 6.1: The percentage of efficient constraints of (6.30).

6.1.1 Algorithm

In the view of the aforementioned properties, a trivial decomposition for the FCLOP would be to decompose the problem where the master problem contains all the constraints of the FCLOP except (6.33). The relaxed cuts are then separated and added to the master problem upon violation.

Let $B^s \subset B$ be the set of active triangle-free inequalities at iteration s . The separation of triangle-free inequalities can be done in polynomial time as shown in Grötschel et al. (1985b). Let (\hat{x}, \hat{y}) be the current solution given by the relaxed Master. Of course $\sum_i \hat{y}_i = p$ and we denote $V(\hat{y})$ the set of active nodes. Now, if we let $w = 1 - \hat{x}$, a feasible acyclic solution will satisfy $w(C) \geq 1$ for any dicycle C on $V(\hat{y})$. So the separation subproblem will be to find a dicycle \hat{C} on $V(\hat{y})$ with minimal $w(C)$. If $w(\hat{C}) < 1$, the minimal dicycle violates the corresponding dicycle inequality i.e. $x(\hat{C}) \leq |\hat{C}| - 1$, which must be added in the Master to the set B^{s+1} .

PR Relaxed Master

$$\begin{aligned}
 & \max \sum_{i \in V} \sum_{j \in V, j \neq i} c_{ij} x_{ij} \\
 & s.t. \\
 & \quad x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V, i < j \\
 & \quad x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V, i < j \\
 & \quad x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall i, j \in V, i < j \quad (6.34) \\
 & \quad x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall C = (ijk) \in B^s \quad (6.35) \\
 & \quad \sum_{i \in V} y_i = p
 \end{aligned}$$

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

PR Subproblem

$$\begin{aligned} \min \quad & w(C) \\ \text{s.t.} \quad & \\ & C \text{ is a triangle-free in } V(\hat{y}) \end{aligned}$$

Observe that we can restrict the separation to triangle-frees and that we can generate as many cuts as there are dicycles with $w(C) < 1$. The process is guaranteed to be finite as the total number of violated inequalities is bounded.

In practice, Grötschel et al. (1985b) suggested to adapt a shortest-path algorithm (like Dijkstra) to find a shortest dicycle in the graph. Another strategy is that for a given set of nodes in an order (in a solution to the relaxed MP), we solve the system of triangle-free inequalities with fixed RHS. If infeasible, we can determine a set of Irreducible Infeasible Subsystem which correspond to those cuts that must be added. In CPLEX, this can be carried out within a Cut/LazyConstraint callbacks and the identified cuts, if any, can be directly added:

$$\min \quad 0 \tag{6.36}$$

s.t.

$$x_{ij} + x_{jk} + x_{ki} \leq 2/3(y_i + y_j + y_k) \quad \forall i, j, k \in V_n, i \neq j \neq k \tag{6.37}$$

$$x_{ij} = \bar{x}_{ij}, \quad \forall i, j \neq i, \tag{6.38}$$

$$y_i = \bar{y}_i, \quad \forall i, \tag{6.39}$$

$$y_i \in \{0, 1\} \quad \forall i \in V_n \tag{6.40}$$

where (\bar{x}, \bar{y}) is the current solution of the master program.

We have implemented this algorithm within a modern branch-and-bound framework where the cuts are added in the course of branch-and-bound and do not need to iterate between solving a master problem to optimality and then separating cuts by invoking separation routing.

In the course of branch-and-bound algorithm as soon as an integer solution to the MP is found and before it is installed as an incumbent, we examine the feasibility of the solution. If the solution is infeasible we try to find an IIS of the set (6.37)-(6.40), which represent the minimal number of required cuts to eliminate the integer solution and avoid installing it as an incumbent if infeasible.

Remark 7 *CPLEX is in charge of identifying Irreducible Infeasible Subsystem (IIS). Finding IIS is NP-hard and here in this algorithm we use the concept of Inclusion-wise Minimum Infeasible Subsystem, instead.*

Upon finding violated constraints from the set of constraints in (6.33), the set is updated to not include those constraints which have been already added to the model.

The process is guaranteed to be finished in polynomial number of separation. Moreover, the total number of violated inequalities is bounded.

In addition to the IIS results, the following cover constraints are generated which is of the same class of combinatorial cuts as in the original version of the combinatorial benders decomposition cuts (Fischetti et al., 2009).

Let I be the set of indexes of variables appearing in the set of constraints in IIS. The following cut eliminates the current solution proposed by the master problem.

$$\sum_{\{i,j\} \in I} (x_{ij} - 1) \leq -1 \quad (6.41)$$

(6.41) ensures that at least one of the variables x_{ij} which is '1' in the current solution will cannot be there. This constraint does not include y variables because the variables x are mainly the cause of infeasibility.

The following algorithm is called at every node of branch-and-bound tree with integer solution:

Algorithm 2: Separation process.

Input: $y_i : i \in V_n$

Output: subset of constraints (6.37)

- 1 fix $y_i : i \in V_n$ in all constraints (6.37);
 - 2 determine set of violated constraints in (6.37);
 - 3 $C = \text{find IIS of (6.37)-(6.40)}$;
 - 4 add C to relaxed FCLOP as global cuts;
 - 5 eliminate C from (6.37) permanently.
-

Numerical results are presented in Table 6.1.

6.1.2 Extended pure relaxation algorithm

In *Extended Pure Relaxation Algorithm* (EPRA), we relax (totally remove) the set of constraints that ensures the existence of an edge between any pair of active vertices.

Relaxing these constraints will cause removal of all the edges having zero cost in any feasible (or optimal) solution (i.e. if there exist i, j of which $w_{ij} = c_{ji} = 0$, then either x_{ij} or x_{ji} will never appear in any solution). The idea behind it is to make an equilibrium between all the symmetric solutions

of which are made by inverting the zero cost value edges.

The relaxed model then is as follows:

Relaxed-FCLOP

$$\begin{aligned}
& \max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \\
& s.t. \\
& \quad x_{ij} + x_{ji} \leq y_i \quad \forall \ i, j \in V_n, j < i \\
& \quad x_{ij} + x_{ji} \leq y_j \quad \forall \ i, j \in V_n, j < i \\
& \quad \sum_{i \in V_n} y_i = p \\
& \quad x_{ij}, y_i \in \{0, 1\}, \quad \forall \ i, j \in V_n, j \neq i
\end{aligned}$$

In this model, the set of

$$\begin{aligned}
& x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall \ i, j \in V_n, i < j \\
& \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} = p(p-1)/2
\end{aligned}$$

constraints are relaxed. By relaxing them we only consider the edges that have non-zero cost value. This helps us to reduce the symmetry inherently exists in the FCLOP, depending on the input matrix. For example before relaxing these constraints, if there exists an active pair i immediately followed by j in a feasible linear order such that $w_{ij} = c_{ji} = 0$, then we will have the same objective value by inverting the edge existed from i to j . We gain nothing by having both of these solutions in the branch-and-bound tree. However after relaxing these constraints we will never have any edges with zero cost value appearing in any feasible solution. It means that we set a unique value to a set of several feasible linear order that are having the same objective value.

After all, we apply the cut separation algorithm to separate the violated triangle-free inequalities, if any, and solve the model.

It is shown in Table 6.2 that comparing to the Table 6.1, we can solve the problem in a very reduced elapsed time. Since the relaxed constraint was to ensure the existence of an arc between any pair of selected vertices and the resulted feasible solution expected to have all the non-zero arcs active in the solution as well as the real optimal solution, one may expect to exploit the optimal solution after a simple post-processing procedure by adding a zero weight arc (x_{ij} or x_{ji}) between any pair of active i and j .

Unfortunately this is not the case. The issue, which cause to have non-zero

gap is that after removing the zero-weight arcs, we might have cycles made by more than 3 directed arcs.

It is known in the literature that all the k -dicycle inequalities define facets for the "Acyclic Subgraph polytope" (Grötschel et al., 1985b), but the triangle-free inequalities are the only ones of this class that define facets for "Linear Ordering polytope" (Grötschel et al., 1985a).

The main difference between these two models comes from the fact that we are dealing with a complete graph in linear ordering problem (a complete subgraph in the FCLOP).

Consequently, after removing the constraints ensuring the existence of an edge between any active pair of vertices we probably have an incomplete subgraph (made on a set of p active vertices). Then in this case the cycles of which containing 4 edges or more, might appear.

Note that this behavior is highly dependant on the structure of input data. That is the reason of having zero duality gap for some instances, i.e. for N-pal19 when cardinality is equal to 5, 10 15 18 or 19, also for N-atp48 when the cardinality is 10 and 15 and also N-atp24 when the cardinality is equal to 5 and 10.

The most interesting point in this observation is the value of relative gap that is quite small.

Remind that to find the relative gap, we calculate the difference between estimated value (upper bound U) and optimal value O if it is known or the best known feasible solution when the optimal value is not known (the absolute gap) and divide it by the optimal value (or the best known feasible solution).

$$relative\ gap = \frac{UpperBound - Optimal^*}{Optimal^*} \times 100$$

Numerical results reported in Table 6.2 confirm that such situations happen rarely (depending on the cost matrix) and we always have very tight upper bound even for difficult instances e.g. N-atp45-20, N-atp45-25, N-atp45-30 and N-atp45-35.

6.1.3 Relaxation Based Algorithm hybrid with Lagrangian Relaxation

As another idea, we have examined the relaxation based algorithm hybrid with Lagrangian relaxation algorithm. In this algorithm, we first relaxed the triangle-free inequalities as it is explained above, then ask CPLEX to identify the IIS. In the sequel, if the IIS is not empty, the separated cuts will be multiplied by lagrangian dual multipliers and added to the objective

function of the master problem. The dual problem then is solved applying Bundle method. Numerical results are reported in Table 6.3.

Table 6.4 is dedicated to report the comparison between the calculation time of the three algorithms, Relaxation based exact solution algorithm, Extended relaxation base algorithm and Relaxation based algorithm hybrid with Lagrangian relaxation, and the dual bound obtained from any of them.

6.2 Numerical results

The same testbed as before is used to test the new algorithms. CPLEX 12.5.1 has been used as a modern branch-and-bound framework in the sense that user can interfere the standard operation by implementing callbacks. All the instances are solved using the same machine as before but only two threads of CPU.

We let CPLEX solve the master problem and in the course of the branch-and-bound algorithm we examine the incumbents before being accepted by CPLEX as a feasible solution. If at certain integer node –where the incumbent is proposed– the IIS is not empty, then we add two kind of constraints: 1) constraints obtained from the IIS, 2) the cover inequality (also known as the combinatorial Benders (6.41)). This will let CPLEX continue without accepting that incumbent. In the case IIS is empty then the optimality has been obtained.

In this framework, the user is not obliged to create and examine all the cuts in any iteration to prevent the memory issues. The solver provides the possibility to generate the cuts on the fly, examined them and keep them only in the case of necessity.

In general, numerical results show that for different instances having small cardinality number, we managed to significantly improve the computational time.

The term O.M. in last column of the tables indicate that the solver (CPLEX) terminates by out of memory status.

It is interesting to note that unless in the cases when the machine ran out of memory, we always have optimal solution once CPLEX terminates normally.

6.2.1 Pure Relaxation Based Cutting Plane algorithm

Table 6.1 shows the numerical results concerning the relaxation based cutting plane algorithm.

One observes that when $p \ll n$, most of the instances are solved in optimality after reasonable time. The column %Eff.Cuts shows the ratio of number of user cuts over the total number of triangle-free inequalities existed in the

FCLOP model. N-atp48-48-25 is an instance that we have never solved it before in optimality, which is solved to optimality for the first time.

Fname	Obj.Val.	Time	#U.Cuts	Eff.Cuts(%)	#Nodes	Status
N-atp24-5	18	0.71	0	0	0	Optimal
N-atp24-10	53	20.93	83	2.05	130	Optimal
N-atp24-15	94	117.06	388	9.58	734	Optimal
N-atp24-20	140	212.82	563	13.90	243	Optimal
N-atp24-23	165	314.60	753	18.60	156	Optimal
N-atp24-24	172	294.31	616	15.21	31	Optimal
N-atp48-10	53	4622.58	393	1.13	3855	Optimal
N-atp48-15	97	11895.9	930	2.68	20414	Optimal
N-atp48-20	-	-	-	-	-	O.M.
N-atp48-25	205	36645.7	2536	7.33	54094	Optimal
N-atp48-30	264	48681.6	3464	10.01	35790	Optimal
N-atp48-35	324	76206.5	4933	14.26	60224	Optimal
N-atp48-40	387	65660.2	5069	14.65	12003	Optimal
N-atp48-45	-	-	-	-	-	O.M.
N-atp48-47	-	-	-	-	-	O.M.
N-atp48-48	-	-	-	-	-	O.M.
N-be75eec-20	130506	3898.72	232	0.59	412	Optimal
N-be75eec-30	191061	3505.25	242	0.61	3	Optimal
N-be75eec-40	223672	17249.6	826	2.10	46	Optimal
N-be75eec-50	236464	21468.7	1036	2.64	7	Optimal
N-econ36-15	400547	185.60	85	0.59	7	Optimal
N-econ36-20	469584	333.18	128	0.89	0	Optimal
N-econ36-25	506232	1624.58	478	3.34	40	Optimal
N-econ36-30	534406	2608.97	794	5.56	12	Optimal
N-econ36-35	547501	2594.73	1131	7.92	20	Optimal
N-econ36-36	548574	4177.09	1220	8.54	17	Optimal
N-p50-20-10	-	-	-	-	-	O.M.
N-p50-20-15	-	-	-	-	-	O.M.
N-p50-20-20	-	-	-	-	-	O.M.
N-p50-20-25	-	-	-	-	-	O.M.
N-p50-20-30	-	-	-	-	-	O.M.
N-p50-20-35	-	-	-	-	-	O.M.
N-p50-20-40	-	-	-	-	-	O.M.
N-p50-20-45	-	-	-	-	-	O.M.
N-p50-20-49	-	-	-	-	-	O.M.
N-p50-20-50	-	-	-	-	-	O.M.
N-pal19-5	10	0.73	12	0.61	0	Optimal
N-pal19-10	36	299.78	450	23.21	89420	Optimal
N-pal19-15	72	1763.73	562	28.99	157997	Optimal
N-pal19-18	98	3216.31	647	33.38	966470	Optimal
N-pal19-19	107	210.89	598	30.85	72960	Optimal
N-pal43-10	-	-	-	-	-	O.M.
N-pal43-15	-	-	-	-	-	O.M.
N-pal43-20	-	-	-	-	-	O.M.
N-pal43-25	-	-	-	-	-	O.M.

N-pal43-30	-	-	-	-	-	O.M.
N-pal43-35	-	-	-	-	-	O.M.
N-pal43-40	-	-	-	-	-	O.M.
N-pal43-42	-	-	-	-	-	O.M.
N-pal43-43	-	-	-	-	-	O.M.

Table 6.1: Numerical experiment for the pure relaxation based cutting plane algorithm

6.2.2 Extended pure Relaxation Based Cutting Plane Algorithm

Table 6.2 shows the numerical results concerning Relaxed Cutting Plane algorithm. The first column *Bound* reports the best upper bound that algorithm has found for the instance. In several instances i.e. N-atp48-10, N-atp48-15, N-atp48-20 and ..., the upper bound reported in this column is equal to the optimal solution value. In such cases, the gap reported in the second column *Gap(%)* is equal to zero. In other cases, the second column reports percentage of the relative gap. One observes that unless the two sets of instances N-p50-20 and N-pal43 that the solution procedure has been terminated due to the memory issue, in all the other cases, the relative gap is quite promising (less than or equal to 0.02%).

The next column, *Time*, reports the computational time required to solve the model. One observes that comparing to the computational time required for solving the cutting plane algorithm (reported in Table 6.1) in almost all the cases, solver needs relatively less time to solve the problem.

The column *U.Cuts(#)* reports the number of cuts that are generated and added to the model in the course of cutting plane algorithm.

The column *%Eff.Cuts* shows the ratio of number of user cuts over the number of all the cuts available in the FCLOP model.

Finally, the last column *B-B Nodes* reports the number of Branch-and-Bound tree nodes that have been checked until proving the optimality.

Remind that unless for two sets of instances N-p50-20 and N-pal43 that the solver has ran out of memory, in all the other instances the solution process has been terminated in optimality (Best dual bound reported in this table is the optimal solution of the relaxed problem).

Fname	Best Bound	Gap (%)	Time (Sec.)	U.Cuts (#)	Eff.Cuts (%)	B-B Nodes
N-atp24-5	18	0	5.65	3	0.07	539
N-atp24-10	53	0	11.25	21	0.51	579
N-atp24-15	95	0.01	13.84	29	0.71	402

N-atp24-20	142	0.01	16.25	38	0.93	164
N-atp24-23	167	0.01	19.71	39	0.96	17
N-atp24-24	175	0.01	9.84	36	0.88	0
N-atp48-10	53	0	2309.81	50	0.14	81369
N-atp48-15	97	0	2915.95	70	0.2	163056
N-atp48-20	148	0	2374.17	84	0.24	117803
N-atp48-25	208	0.01	2568.31	113	0.32	49887
N-atp48-30	267	0.01	2792.83	139	0.4	27050
N-atp48-35	329	0.01	3124.75	162	0.46	10953
N-atp48-40	393	0.01	2553.77	152	0.43	950
N-atp48-45	454	0.01	3306.56	164	0.47	339
N-atp48-47	480	0.01	2554.61	158	0.45	46
N-atp48-48	493	0.02	877.59	143	0.41	0
N-be75eec-20	130532	0.0001	969.37	42	0.1	14
N-be75eec-30	191160	0.0005	1373.13	104	0.26	0
N-be75eec-40	224642	0.004	2824.3	201	0.51	0
N-be75eec-50	237588	0.004	3091.73	269	0.68	0
N-econ36-15	400602	0.0001	105.9	38	0.26	5
N-econ36-20	470184	0.001	190.03	61	0.42	0
N-econ36-25	507301	0.002	498.59	132	0.92	19
N-econ36-30	536588	0.004	535.48	215	1.5	0
N-econ36-35	551347	0.007	540.37	311	2.17	0
N-econ36-36	552379	0.006	536.79	336	2.35	0
N-p50-20-10	-	-	-	-	-	-
N-p50-20-15	-	-	-	-	-	-
N-p50-20-20	-	-	-	-	-	-
N-p50-20-25	-	-	-	-	-	-
N-p50-20-30	-	-	-	-	-	-
N-p50-20-35	-	-	-	-	-	-
N-p50-20-40	-	-	-	-	-	-
N-p50-20-45	-	-	-	-	-	-
N-p50-20-49	-	-	-	-	-	-
N-p50-20-50	-	-	-	-	-	-
N-pal19-5	10	0	11.95	22	1.13	14303
N-pal19-10	36	0	120.17	257	13.26	63236
N-pal19-15	72	0	161.53	285	14.7	48844
N-pal19-18	98	0	4543.44	281	14.49	1173336
N-pal19-19	107	0	449.9	283	14.6	234469
N-pal43-10	-	-	-	-	-	-
N-pal43-15	-	-	-	-	-	-
N-pal43-20	-	-	-	-	-	-
N-pal43-25	-	-	-	-	-	-
N-pal43-30	-	-	-	-	-	-
N-pal43-35	-	-	-	-	-	-
N-pal43-40	-	-	-	-	-	-
N-pal43-42	-	-	-	-	-	-
N-pal43-43	-	-	-	-	-	-

Table 6.2: Numerical results for extended pure relaxation algorithm

6.2.3 Cutting Plane Algorithm Hybrid with Lagrangian Relaxation

Table 6.3 reports the numerical results concerning the cutting plane algorithm hybrid with Lagrangian relaxation method.

In this model, the cutting plane algorithm will call CPLEX to identify the irreducible infeasible subsystem (IIS) of the violated triangle-free inequalities. Once the IIS is known, the violated cuts in it will be added to the objective function of the master problem in a lagrangian form. Bundle method is applied to solve the dual subproblem.

In this algorithm, we have two levels of iterations: First is the cutting plane algorithm that is to identify the violate cuts. Second level is the iterations to solve the sub-gradient problem that is placed inside each iteration of the cutting plane algorithm.

As always, the first column in Table 6.3 contains name of the examined instances. The second column *elapsed Time* reports the computational time required to solve the instance.

The third column *Rel. Gap(%)* reports the relative gap of the solution founded in this algorithm. The next column *nr. Iter.* reports the number of iterations to solve the sub-gradient algorithm in the last iteration of the cutting plane algorithm right before pruning the optimality (finding an empty IIS set).

The next column, *LP Rel. Gap(%)* reports the LP relative gap to make it easy to be compared with the relative gap resulted by this algorithm and the last column reports the termination condition of the solution process.

One observes that in several very difficult instances e.g. *N-p50-20* and *N-pal43* for different cardinality number, cutting plane algorithm hybrid with lagrangian relaxation has provided a relatively high quality dual bound. Time limit to solve the bundle algorithm is 10800 seconds.

Fname	elapsed Time	best Bound	Rel.Gap(%)	nr. Iter.	LP Rel.Gap(%)	CPLEX Status
N-atp24-5	1	18	0	2	128	Optimal
N-atp24-10	7	53	0	3	54.86	Optimal
N-atp24-15	55	98.61	4.90	25	30.97	Optimal
N-atp24-20	77	146.37	4.55	85	9.4	Optimal
N-atp24-23	-	-	-	-	1.96	O.M.
N-atp24-24	-	-	-	-	0	O.M.
N-atp48-10	45	53	0	4	123.66	Optimal
N-atp48-15	742	101.67	4.81	16	83.30	Optimal
N-atp48-20	11277	158.87	7.34	36	60.18	TimeLimit
N-atp48-25	11398	220.75	7.68	27	44.56	TimeLimit
N-atp48-30	11270	285.48	8.13	26	34.70	TimeLimit
N-atp48-35	11077	349.03	7.72	60	23.65	TimeLimit
N-atp48-40	-	-	-	-	12.70	O.M.
N-atp48-45	-	-	-	-	4.38	O.M.
N-atp48-47	-	-	-	-	1.20	O.M.

N-atp48-48	-	-	-	-	0	O.M.
N-be75eec-20	15	131910	1.07	12	1.88	Optimal
N-be75eec-30	22	191201	0.07	27	0.05	Optimal
N-be75eec-40	86	224351	0.30	41	0.14	Optimal
N-be75eec-50	282	237081	0.26	57	0	Optimal
N-econ36-15	3	401546	0.24	6	0.66	Optimal
N-econ36-20	4	472666	0.65	12	0.09	Optimal
N-econ36-25	6	512316	1.20	17	0.26	Optimal
N-econ36-30	15	540244	1.09	27	0.11	Optimal
N-econ36-35	31	554661	1.30	37	0.08	Optimal
N-econ36-36	30	555394	1.24	37	0.002	Optimal
N-p50-20-10	421	3680	59.23	13	474.75	Optimal
N-p50-20-15	8142	7607.44	48.99	13	290.20	Optimal
N-p50-20-20	13226	12860.9	73.86	8	259.13	TimeLimit
N-p50-20-25	13736	19155.8	39.85	9	142.43	TimeLimit
N-p50-20-30	11580	26532.1	55.59	13	133.68	TimeLimit
N-p50-20-35	11373	34745.4	49.13	21	91.74	TimeLimit
N-p50-20-40	10876	43426.3	38.57	64	46.35	TimeLimit
N-p50-20-45	-	-	-	-	23.50	O.M.
N-p50-20-40	-	-	-	-	13.04	O.M.
N-p50-20-40	-	-	-	-	5.69	O.M.
N-pal19-5	1	10	0	3	350	Optimal
N-pal19-10	0	45	25	3	150	Optimal
N-pal19-15	3	105	45.83	11	58.33	Optimal
N-pal19-18	7	114.09	16.41	42	16.32	Optimal
N-pal19-19	0	114	6.54	4	6.54	Optimal
N-pal43-10	2	45	12.5	3	425	Optimal
N-pal43-15	1	105	31.25	3	293.75	Optimal
N-pal43-20	1	190	36.69	3	202.15	Optimal
N-pal43-25	25	300	50.75	4	163.81	Optimal
N-pal43-30	39	435	57.60	10	118.11	Optimal
N-pal43-35	-	-	-	-	66.75	O.M.
N-pal43-40	86	602	31.15	51	31.15	Optimal
N-pal43-42	95	602	20.4	47	20.4	Optimal
N-pal43-43	8	602	12.52	4	12.52	Optimal

Table 6.3: Numerical experiment of relaxation based algorithm hybrid with Lagrangian relaxation

6.2.4 Comparing the three algorithms

Table Table 6.4 is served to compare the numerical results of relaxation based cutting plane algorithm, extended relaxation based cutting plane algorithm and the relaxation based cutting plane algorithm hybrid with the Lagrangian relaxation algorithm.

The first column as always contains the name of instances.

The second column *RBA Time* reports the calculation time that pure re-

laxation based algorithm(RBA) required for solving the instances of the FCLOP.

The third column *ERBA Bound* reports the best bound provided by the extended pure relaxation based algorithm (ERBA) while the next column contains the LP bound of the instances for making it easy to compare the bounds.

The next column *RBA+LR Time* reports the calculation time required for the relaxation based algorithm hybrid with Lagrangian relaxation algorithm and the last column *RBA+LR Relative Gap(%)* shows the percentage of the relative gap of solutions resulted from the relaxation based algorithm hybrid with Lagrangian relaxation algorithm.

One observes that relaxation based separation algorithm is valuable mostly whenever the cardinality number is far from the number of node. The separation algorithm becomes heavy as soon as the cardinality number get close in to the number of nodes (e.g. *N-atp48* and *N-p50-20-50*).

Fname	RBA Time	ERBA Time	ERBA Rel.Gap(%)	RBA+LR Time	RBA+LR Rel.Gap(%)	LP Rel.Gap(%)
N-atp24-24-5	0.71	5.65	0	1	0	128
N-atp24-24-10	20.93	11.25	0	7	0	54.86
N-atp24-24-15	117.06	13.84	0.01	55	4.90	30.97
N-atp24-24-20	212.82	16.25	0.01	77	4.55	9.4
N-atp24-24-23	314.6	19.71	0.01	-	-	1.96
N-atp24-24-24	294.31	9.84	0.01	-	-	0
N-atp48-48-10	4622.58	2309.81	0	45	0	123.66
N-atp48-48-15	11895.9	2915.95	0	742	4.81	83.30
N-atp48-48-20	-	2374.17	0	11277	7.34	60.18
N-atp48-48-25	36645.7	2568.31	0.01	11398	7.68	44.56
N-atp48-48-30	48681.6	2792.83	0.01	11270	8.13	34.70
N-atp48-48-35	76206.5	3124.75	0.01	11077	7.72	23.65
N-atp48-48-40	65660.2	2553.77	0.01	-	-	12.70
N-atp48-48-45	-	3306.56	0.01	-	-	4.38
N-atp48-48-47	-	2554.61	0.01	-	-	1.20
N-atp48-48-48	-	877.59	0.02	-	-	0
N-be75eec-50-20	3898.72	969.37	0.0001	15	1.07	1.88
N-be75eec-50-30	3505.25	1373.13	0.0005	22	0.07	0.05
N-be75eec-50-40	17249.6	2824.3	0.004	86	0.30	0.14
N-be75eec-50-50	21468.7	3091.73	0.004	282	0.26	0
N-econ36-36-15	185.6	105.9	0.0001	3	0.24	0.66
N-econ36-36-20	333.18	190.03	0.001	4	0.65	0.09
N-econ36-36-25	1624.58	498.59	0.002	6	1.20	0.26
N-econ36-36-30	2608.97	535.48	0.004	15	1.09	0.11
N-econ36-36-35	2594.73	540.37	0.007	31	1.30	0.08
N-econ36-36-36	4177.09	536.79	0.006	30	1.24	0.002
N-p50-20-50-10	-	-	-	421	59.23	474.75
N-p50-20-50-15	-	-	-	8142	48.99	290.20

N-p50-20-50-20	-	-	-	13226	73.86	259.13
N-p50-20-50-25	-	-	-	13736	39.85	142.43
N-p50-20-50-30	-	-	-	11580	55.59	133.68
N-p50-20-50-35	-	-	-	11373	49.13	91.74
N-p50-20-50-40	-	-	-	10876	38.57	46.35
N-p50-20-50-45	-	-	-	-	-	23.50
N-p50-20-50-49	-	-	-	-	-	13.04
N-p50-20-50-50	-	-	-	-	-	5.69
N-pal19-19-5	0.73	11.95	0	1	0	350
N-pal19-19-10	299.78	120.17	0	0	25	150
N-pal19-19-15	1763.73	161.53	0	3	45.83	58.33
N-pal19-19-18	3216.31	4543.44	0	7	16.41	16.32
N-pal19-19-19	210.89	449.9	0	0	6.54	6.54
N-pal43-43-10	-	-	-	2	12.5	425
N-pal43-43-15	-	-	-	1	31.25	293.75
N-pal43-43-20	-	-	-	1	36.69	202.15
N-pal43-43-25	-	-	-	25	50.75	163.81
N-pal43-43-30	-	-	-	39	57.60	118.11
N-pal43-43-35	-	-	-	-	-	66.75
N-pal43-43-40	-	-	-	86	31.23	31.15
N-pal43-43-42	-	-	-	95	20.492	20.4
N-pal43-43-43	-	-	-	8	12.52	12.52

Table 6.4: Comparison table

CHAPTER 7

Facets and Numerical Efficiency

Our investigation in polyhedral studies presented in chapter 3, shows that the inequalities

$$x_{ij} + x_{jk} - x_{ik} - y_j \leq 0, \quad \forall i, j \in V_n, i \neq j$$

are defining facets for the cases when $p = n$ and $p = n - 1$. We have also shown that in all the other cases, they improve the LP bound.

The set of (??) facet defining valid inequalities are new and can be replaced with the classical triangle-free inequalities. Adding these new facet defining valid inequalities to the FCLOP model dramatically improve the integrality gap. Numerical results of comparing the LP relaxation of the instances before and after adding (??) to the model are presented in Table 3.7.

In chapter 7, we are proposing two Lagrangian relaxations as well as a relax-and-cut algorithm to improve the dual bound and compare the results with the old tables presented before, in chapter 3.

7.1 Lagrangian relaxation

Consider the FCLOP model as follows:

FCLOP

$$\max \quad \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \tag{7.1}$$

s.t.

$$x_{ij} + x_{ji} \leq y_i \quad \forall \ i, j \in V_n, i < j \quad (7.2)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall \ i, j \in V_n, i < j \quad (7.3)$$

$$x_{ij} + x_{ji} - y_i - y_j + 1 \geq 0 \quad \forall \ i, j \in V_n \ i < j \quad (7.4)$$

$$x_{ij} + x_{jk} - x_{ik} - y_j \leq 0 \quad \forall \ i, j, k \in V_n, i \neq j \neq k \quad (7.5)$$

$$\sum_{i \in V_n} y_i = p \quad (7.6)$$

$$\sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} = p(p-1)/2 \quad (7.7)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall \ i, j \in V_n, i \neq j \quad (7.8)$$

The following two Lagrangian relaxations has been examined.

LR1

$$\begin{aligned} \max \quad & \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \\ & - \sum_{k \in V_n} \sum_{j \in V_n, j \neq k} \sum_{i \in V_n, i \neq j \neq k} v_{ijk} (x_{ij} + x_{jk} - x_{ik} - y_j) \end{aligned} \quad (7.9)$$

s.t.

$$x_{ij} + x_{ji} \leq y_i \quad \forall \ i, j \in V_n, i < j \quad (7.10)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall \ i, j \in V_n, i < j \quad (7.11)$$

$$x_{ij} + x_{ji} - y_i - y_j + 1 \geq 0 \quad \forall \ i, j \in V_n \ i < j \quad (7.12)$$

$$\sum_{i \in V_n} y_i = p \quad (7.13)$$

$$\sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} = p(p-1)/2 \quad (7.14)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall \ i, j \in V_n, i \neq j \quad (7.15)$$

LR2

$$\begin{aligned} \max \quad & \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \\ & - \sum_{j \in V_n} \sum_{i \in V_n, i \neq j} u_{ij} (-x_{ij} - x_{ji} + y_i + y_j - 1) \end{aligned} \quad (7.16)$$

s.t.

$$x_{ij} + x_{ji} \leq y_i \quad \forall \ i, j \in V_n, i < j \quad (7.17)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V_n, i < j \quad (7.18)$$

$$x_{ij} + x_{jk} - x_{ik} - y_j \leq 0 \quad \forall i, j, k \in V_n \ i \neq j \neq k \quad (7.19)$$

$$\sum_{i \in V_n} y_i = p \quad (7.20)$$

$$\sum_{i \in V_n} \sum_{j \in V_n, j \neq i} x_{ij} = p(p-1)/2 \quad (7.21)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n, i \neq j \quad (7.22)$$

7.1.1 Numerical results

Subgradient method and bundle algorithm have been used to solve the two Lagrangian relaxation **LR1** and **LR2**. Numerical results are presented in two following Tables 7.1 and 7.2.

LR1

Table 7.1 reports the numerical results of applying bundle method and the code implemented by Antonio Frangioni (2000) to solve LR1 Lagrangian dual problem. In this model, we have dualized new triangle-free inequalities. Numerical results show that for the instances which are usually very difficult to solve, i.e. **N-atp48** and **N-p50-20**, we were able to dramatically improve the dual bound. On the other side, it seems that this Lagrangian dual problem is not the best model to find a good upper bound for easier instances when CPLEX itself can solve the instances easily. As mentioned earlier, computational results of different solution method for the FCLOP are always highly depending on the input instances. Moreover, our experiments show that different instances have different behavior facing with different solution methods that we have examined.

For example, here we could not solve **N-atp24** for three cases when $p = 20$, $p = 23$ and $p = 24$ while these instances were of the easiest ones so far. However we had relatively good dual bounds for **N-atp48** when p gets medium values (e.g. $p = 20$) which were quite challenging for almost all the solution methods we have applied so far.

A very interesting point in Table 7.1 is the presence of very high quality dual bound for some of the very difficult instances of which are found for the first time, e.g. **N-p50-20** for various p values and **N-pal43** as well.

LR2

Table 7.2 shows numerical results applying Bundle algorithm to solve Lagrangian dual problem LR2. In this model, the set of (7.4) inequalities are dualized. Numerical experiments show that except only two instances **N-econ36** when $p = 25$ and the second one when $p = 35$, the other instances of which were not running out of memory, all are solved in optimality in a

Fname	elapsedTime	bestBound	nrIteration	LP bound	Optimal*	BundleStatus
N-atp24-5	1	18	2	41.04	18	Optimal
N-atp24-10	11	53	5	82.08	53	Optimal
N-atp24-15	189	94.67	73	123.12	94	Optimal
N-atp24-20	-	-	-	153.16	140	O.M.
N-atp24-23	-	-	-	168.25	165	O.M.
N-atp24-24	-	-	-	172	172	O.M.
N-atp48-10	226	53	12	118.54	53	Optimal
N-atp48-15	10915	98.14	80	177.81	97	TimeLimit
N-atp48-20	11064	152.73	24	237.08	148	TimeLimit
N-atp48-25	11180	211.35	20	296.35	205	TimeLimit
N-atp48-30	10927	269.13	28	355.62	264	TimeLimit
N-atp48-35	10873	325.60	69	400.65	324	TimeLimit
N-atp48-40	-	-	-	436.16	387	O.M.
N-atp48-45	-	-	-	466.62	447	O.M.
N-atp48-47	-	-	-	477.7	472	O.M.
N-atp48-48	-	-	-	483	483	O.M.
N-be75eec-20	23	130904	18	132966	130506	Optimal
N-be75eec-30	15	191115	27	191168	191061	Optimal
N-be75eec-40	36	223988	45	224006	223672	Optimal
N-be75eec-50	9	237081	57	236464	236464	Optimal
N-econ36-15	3	401531	6	403222	400547	Optimal
N-econ36-20	2	472666	12	470033	469584	Optimal
N-econ36-25	4	512039	18	507568	506232	Optimal
N-econ36-30	7	539997	29	535017	534406	Optimal
N-econ36-35	7	554661	37	547984	547501	Optimal
N-econ36-36	2	555394	37	548588	548574	Optimal
N-p50-20-10	10887	3528.09	45	13282.6	2311*	TimeLimit
N-p50-20-15	12056	7528.93	12	19923.9	5106*	TimeLimit
N-p50-20-20	15816	12690.7	6	26565.2	7397*	TimeLimit
N-p50-20-25	21686	18738.7	6	33206.5	13697*	TimeLimit
N-p50-20-30	13645	25314.4	8	39847.8	17052*	TimeLimit
N-p50-20-35	11642	30719.8	16	44672.5	23298*	TimeLimit
N-p50-20-40	-	-	-	45864.2	31337*	O.M.
N-p50-20-40	10986	35431.7	68	47055.8	38099*	TimeLimit
N-p50-20-49	37	46974.7	94	47985.7	42450*	Optimal
N-p50-20-50	-	-	-	48155.7	45560*	O.M.
N-pal19-5	0	10	6	45	10	Optimal
N-pal19-10	9	45	40	90	36	Optimal
N-pal19-15	-	-	-	114	72	O.M.
N-pal19-18	1	105.08	26	114	98	Optimal
N-pal19-19	1	114	4	114	107	Optimal
N-pal43-10	44	45	16	210	40*	Optimal
N-pal43-15	252	105	42	315	80*	Optimal
N-pal43-20	858	190	64	420	139*	Optimal
N-pal43-25	12365	299.98	34	525	199*	TimeLimit
N-pal43-30	11362	377.21	40	602	276*	TimeLimit
N-pal43-30	-	-	-	602	361*	O.M.
N-pal43-30	-	-	-	602	459*	O.M.
N-pal43-42	53	581.36	39	602	500*	Optimal
N-pal43-43	1	602	4	602	535*	Optimal

Table 7.1: numerical experiment-Lagrangian relaxation-LR1-Bundle method

very reasonable time and after only two iterations.

Another very interesting result show that for the instance N-p50-20 when $p = n = 50$ we could again improve the best dual bound known in the literature to 47504. This LOP problem is one of the very difficult instances available in the literature which its optimal solution is not known yet. The best known dual bound in the literature for this problem is 48155.

In Table 7.2, despite difficulties to tackle very hard to solve instances e.g. N-p50-20 and N-pal43, still we had very good success to solve almost all the other of the testbed in only 2 iterations. Moreover, except only 2 instances N-econ-36 when $p = 25$ and also when $p = 30$, in all the other instances we have **zero** duality gap.

Fname	elapsedTime	bestBound	nrIteration	LP bound	Optimal*	BundleStatus
N-atp24-5	4	18	2	41.04	18	Optimal
N-atp24-10	16	53	2	82.08	53	Optimal
N-atp24-15	17	94	2	123.12	94	Optimal
N-atp24-20	3	140	2	153.16	140	Optimal
N-atp24-23	1	165	2	168.25	165	Optimal
N-atp24-24	1	172	2	172	172	Optimal
N-atp48-10	2041	53	2	118.54	53	Optimal
N-atp48-15	8363	97	2	177.81	97	Optimal
N-atp48-20	12971	148	2	237.08	148	Optimal
N-atp48-25	10443	205	2	296.35	205	Optimal
N-atp48-30	2864	264	2	355.62	264	Optimal
N-atp48-35	3761	324	2	400.65	324	Optimal
N-atp48-40	68	387	2	436.16	387	Optimal
N-atp48-45	85	447	2	466.62	447	Optimal
N-atp48-47	55	472	2	477.7	472	Optimal
N-atp48-48	9	483	2	483	483	Optimal
N-be75eec-20	22	130506	2	132966	130506	Optimal
N-be75eec-30	16	191061	2	191168	191061	Optimal
N-be75eec-40	49	223672	2	224006	223672	Optimal
N-be75eec-50	5	236464	2	236464	236464	Optimal
N-econ36-15	12	400547	2	403222	400547	Optimal
N-econ36-20	4	469584	2	470033	469584	Optimal
N-econ36-25	7	506268	2	507568	506232	Optimal
N-econ36-30	10	534406	2	535017	534406	Optimal
N-econ36-35	9	547542	2	547984	547501	Optimal
N-econ36-36	3	548574	2	548588	548574	Optimal
N-p50-20-10	-	-	-	13282.6	2311*	O.M.
N-p50-20-15	-	-	-	19923.9	5106*	O.M.
N-p50-20-20	-	-	-	26565.2	7397*	O.M.
N-p50-20-25	-	-	-	33206.5	13697*	O.M.
N-p50-20-30	-	-	-	39847.8	17052*	O.M.
N-p50-20-35	-	-	-	44672.5	23298*	O.M.
N-p50-20-40	-	-	-	45864.2	31337*	O.M.
N-p50-20-45	-	-	-	47055.8	38099*	O.M.
N-p50-20-49	-	-	-	47985.7	42450*	O.M.
N-p50-20-50	-	-	-	48155.7	45560*	O.M.
N-pal19-5	1	10	2	45	10	Optimal
N-pal19-10	132	36	2	90	36	Optimal
N-pal19-15	3330	72	2	114	72	Optimal
N-pal19-18	1200	98	2	114	98	Optimal
N-pal19-19	254	107	2	114	107	Optimal
N-pal43-10	-	-	-	210	40*	O.M.
N-pal43-15	-	-	-	315	80*	O.M.
N-pal43-20	-	-	-	420	139*	O.M.
N-pal43-25	-	-	-	525	199*	O.M.
N-pal43-30	-	-	-	602	276*	O.M.
N-pal43-35	-	-	-	602	361*	O.M.
N-pal43-40	-	-	-	602	459*	O.M.
N-pal43-42	-	-	-	602	500*	O.M.
N-pal43-43	-	-	-	602	535*	O.M.

Table 7.2: numerical experiment-lagrangian relaxation-LR2-Bundle method

7.2 Relax-and-Cut Algorithm

7.2.1 Introduction

The more the cardinality of set of dualized constraints, the more time is spent on evaluation of subgradient. If the cardinality is exponential in the size of instance, then evaluation of subgradient becomes a bottleneck. Moreover, often a very small fraction of all the cuts are actually violated. However, those that are actually satisfied still contribute in calculation of $g = b - Ax$ in the denominator of step-size calculation formula. In other words, the number of strictly positive subgradients tends to be huge and therefore the calculated step size will be extremely small, leaving Lagrangian multiplier values virtually unchanged from iteration to iteration. This would hinder the convergence and deteriorates the efficiency of subgradient method.

Beasley (1993) suggested that $g_i = 0$ for strictly positive subgradient i . However, still the issue of exponential number of relaxed constraints persists.

Relax-and-Cut (first mentioned in Escudero et al. (1994)) tried to iteratively add or dualize the violated constraints. Gavish (1985) proposed *Augmented Lagrangian Approach* where the cuts are added after (5.4) has been solved (Delayed Relax-and-Cut). Balas and Christofides (1981) proposed the so called *Restricted Lagrangian Approach* for Traveling Salesman Problem. Lucena (1992, 1993) proposed a new class of relax-and-cut, called Non-Delayed Relax-and-Cut, where the additional constraints to be dualized are identified after every iteration of subgradient and not only after the convergence of subgradient.

Guignard (1998) characterized the kind of valid inequalities which can be efficient in the relax-and-cut process.

7.2.2 Algorithm

In this section, a relax-and-cut algorithm is proposed to solve the FCLOP. This algorithm is based on the results concerning polyhedral study and facet defining inequalities presented in chapter two.

Fischetti and Salvagnin (2011) proposed a relax-and-cut framework for efficiently using gomory mixed-integer cuts. They relax the new generated gomory mixed-integer cuts and add them to the objective function multiplied by the Lagrangian multipliers. Belloni and Lucena (2004) also presents a relax-and-cut algorithm to solve linear ordering problem.

Given the FCLOP as follows:

$$\max \sum_{i \in V_n} \sum_{j \in V_n, j \neq i} w_{ij} x_{ij} \quad (7.23)$$

$$s.t. \quad x_{ij} + x_{ji} \leq y_i \quad \forall i, j \in V_n, i < j \quad (7.24)$$

$$x_{ij} + x_{ji} \leq y_j \quad \forall i, j \in V_n, i < j \quad (7.25)$$

$$x_{ij} + x_{ji} \geq y_i + y_j - 1 \quad \forall i, j \in V_n, i < j \quad (7.26)$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \forall i, j, k \in V_n \quad i \neq j \neq k \quad (7.27)$$

$$\sum_{i \in V_n} y_i = p \quad (7.28)$$

$$x_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in V_n \quad (7.29)$$

It is shown in chapter 2 that the set of following strengthening valid inequalities are facet-defining for the FCLOP polytope:

$$x_{ij} + x_{jk} - x_{ik} - y_j \leq 0 \quad \forall i, j, k \in V_n, i \neq j \neq k \quad (7.30)$$

The idea is to solve the LP relaxation of the FCLOP. Once we find the solution, we check whether there exist any violated facet defining inequality of (7.30) type or not. If there is any, then we add them to the model and directly relax them in a Lagrangian fashion. Indeed, if in an iteration an inequality which is already added to the model is found as a violated inequality, then we need to increase its Lagrangian multiplier to augment the penalty to avoid violating an inequality.

Algorithm 3: Separation process of the relax-and-cut algorithm.

Input: $x_{ij}, y_i : i, j \in V_n$

Output: subset of constraints (7.30)

- 1 fix $x_{ij}, y_i : i, j \in V_n$ in all constraints (7.30);
 - 2 determine set of violated constraints in (7.30);
 - 3 C =find minimum infeasible subsystem of (7.30);
 - 4 add C to FCLOP as relaxed constraints to the objective function;
-

Numerical experiments are presented in Table 7.3.

7.2.3 Numerical results

Numerical experiments in Table 7.3 shows that the behavior of our relax-and-cut algorithm is highly depending on input data. In Table 7.3, we have compared the numerical experiences of relax-and cut algorithm with LR1. In LR1, all the facet defining triangle-free inequalities are directly dualized and then the dual problem is solved using bundle algorithm. As it was expected, we did not change the best bound obtained in relax and bound method comparing the LR1, they are all similar. However the elapsed computational time in these two experiments are not similar for all the instances, i.e. the elapsed computational time of relax-and-cut algorithm for N-atp48 when $p = 10$ is less than LR1, when $p = 15$ they are not too much deferent and for the case when $p = 20$, it is a little bit greater than LR1.

Fname	Relax-and-Cut			LR1	
	elapsedTime	bestBound	nrIteration	elapsedTime	BundleStatus
N-atp24-5	1.28	18	2	0.49	Optimal
N-atp24-10	12.01	53	5	14.59	Optimal
N-atp24-15	189.81	94.67	73	156.3	Optimal
N-atp24-20	-	-	-	-	O.M.
N-atp24-23	-	-	-	-	O.M.
N-atp24-24	-	-	-	-	O.M.
N-atp48-10	226.24	53	12	383.42	Optimal
N-atp48-15	10915.22	98.14	80	10910.02	Optimal
N-atp48-20	11064.69	152.73	24	10401.11	Optimal
N-atp48-25	11181.01	211.35	20	19120.48	Optimal
N-atp48-30	10928.01	269.13	28	7977.48	Optimal
N-atp48-35	10873.08	325.6	69	16419.51	Optimal
N-atp48-40	-	-	-	-	O.M.
N-atp48-45	-	-	-	-	O.M.
N-atp48-47	-	-	-	-	O.M.
N-atp48-48	-	-	-	-	O.M.
N-be75eec-20	23.27	130904	18	29.31	Optimal
N-be75eec-30	15.51	191115	27	13.53	Optimal
N-be75eec-40	36.8	223988	45	48.99	Optimal
N-be75eec-50	9.29	237081	57	15.98	Optimal
N-econ36-15	3.65	401531	6	4.53	Optimal
N-econ36-20	2.82	472666	12	6.18	Optimal
N-econ36-25	4.02	512039	18	1.13	Optimal
N-econ36-30	7.9	539997	29	10.71	Optimal
N-econ36-35	7.69	554661	37	15.17	Optimal
N-econ36-36	2.03	555394	37	0.24	Optimal
N-p50-20-10	10887.88	3528.09	45	5553.01	TimeLimit
N-p50-20-15	12056.42	7528.93	12	21461.74	TimeLimit
N-p50-20-20	15816.3	12690.7	6	15816.39	TimeLimit
N-p50-20-25	12686.07	18738.7	6	2602.37	TimeLimit
N-p50-20-30	13645.56	25314.4	8	23062.31	TimeLimit
N-p50-20-35	11642.65	30719.8	16	116.43	TimeLimit
N-p50-20-40	10986.18	35431.7	68	18566.93	TimeLimit
N-p50-20-49	37.28	48749.7	100	4.12	MaxIter
N-p50-20-50	-	-	-	-	O.M.
N-pal19-5	0.62	10	6	0.8	Optimal
N-pal19-10	9.27	4519	40	9.3	Optimal
N-pal19-15	-	-	-	-	Optimal
N-pal19-18	1.89	105.08	26	2.48	Optimal
N-pal19-19	1.94	114	4	3.8	Optimal
N-pal43-10	44.9	45	16	54.84	Optimal
N-pal43-15	252.55	1056	42	235.5	Optimal
N-pal43-20	858.53	1909	64	17.17	Optimal
N-pal43-25	12365.92	299.98	34	14963.77	Optimal
N-pal43-30	11362.09	377.21	40	13635.25	Optimal
N-pal43-35	-	-	-	-	O.M.
N-pal43-40	-	-	-	-	O.M.
N-pal43-42	53.55	581.36	39	90.63	Optimal
N-pal43-43	1.98	602	4	0.6	Optimal

Table 7.3: numerical experiment of Relax-and-Cut algorithm comparing to LR1

Summary, conclusions and outlook to future work

8.1 Summary and conclusions

In this thesis we introduced a new variant of the classical Linear Ordering Problem (LOP) where the set of nodes to be linearly ordered are exogenously fixed to p . We referred to this new variant as *Fixed Cardinality Linear Ordering Problem (FCLOP)*. A particular case of the FCLOP where $p = n$ becomes LOP and it is well-known that LOP belongs to the class of NP-Hard problems in combinatorial optimization problems (COP).

The problem itself generalizes the LOP and of course would share the same application that are imagined for the LOP. Yet, as also explained in the introduction section, the FCLOP (perhaps with some minor side constraints) is an inherent part of many network location problems (in particular those p -center and p -median problems) in transportation and telecommunications. When developing primal decomposition algorithms, such as Benders decomposition, for such kind of network design problems, often the master problem is composed of the constraints of the FCLOP (perhaps plus some other side constraints) while the flow transfer part moves to the sub-problem.

As the major concern in such primal decomposition algorithms —Benders in this case— is the efficiency of MP resolution, characterizing polyhedral structure and developing efficient solution methods —exploiting such polyhedral information— for the FCLOP becomes even more important.

In this thesis we have started by introducing the problem and proposing an integer programming formulation for the problem at hand. The numerical

results proved the intractability of the model and the impact of symmetry on the performance of general-purpose MIP solvers such as CPLEX. We have used standard instances from different categories of instances from LOLIB. The computational experiments revealed that the quality of relaxation and several iterations of non-improving bounds during the branch-and-bound(-and-cut) process in CPLEX demands for better polyhedral studies of the IP formulation and tightening LP polytope.

We have proposed several equivalent IP formulation and several classes of valid inequalities. Those valid inequalities came from different sources including some optimization over the first Chvátal-Gomory closure. Some classes of valid inequalities have shown to contribute a lot in the performance of general-purpose solvers by tightening the LP relaxation polytope. An effort has been made to also show that some of the proposed valid inequalities were also facet-defining ones.

After studying the formulation of the problem, we have proposed several Lagrangian (decomposition) based relaxations making use of the tighter formulations and the improving inequalities which could also be used in a relax-and-cut fashion. We also proposed a simple method for dynamically generating symmetry-breaking inequalities in the course of branch-and-bound upon emergence of every incumbent.

In addition to the Lagrangian-based dual decomposition approaches, a primal decomposition algorithm which relied on the idea of Combinatorial Benders cuts has been developed which proven to be very efficient and obtain optimal/feasible solution for instances which has not been solved until then.

We can conclude that polyhedral analysis and the improving inequalities led to very promising results. That means, even a black-box general-purpose solver such as CPLEX, not even a tailored algorithm, could show a better performance. In terms of bound, Lagrangian relaxation and in particular, Lagrangian decomposition shown very promising and is a better alternative to the LP relaxation bound as the sub-problems do not show any integrality property.

We strongly believe that the idea of pure relaxation accompanying with a cut separation algorithm which came from Benders decomposition shown to be very promising for some classes of instances. However, still the issues related to a non-convex sub-problem hinders a successful application of the method.

However, still the quality of bound used in the branch-and-bound algorithm as well as symmetry play very important roles.

8.2 Future work

The efficiency of resolution of the FCLOP is very negatively impacted by the number of triangle-free constraints rather than the number of variables. Such a huge number of constraint with very similar and sparse non-zero rows in the matrix is also a serious source of degeneracy and numerical instabilities. Often the solvers are interrupted with exception while solving the root node, even before starting to branch on variables.

In such a situation, a trivial way of handling such problems is to apply cut generation methods such as branch-and-cut and relax-and-cut which generates the necessary cuts on the fly instead of adding them directly to the model. We believe that, branch-and-cut methods deserve more attention. At the same time, an efficient cut generation method also owes to the quality of cuts which in turn relies on the solid knowledge of the polyhedral structure.

Even though the intractability of the model mostly owes to the number of rows in the model, yet perhaps a Dantzig-Wolfe reformulation of the problem could significantly reduce the existing symmetry issue in the model provided that the pricing sub-problem is efficiently solvable (perhaps by some kind of branch-and-cut or a variant of Benders). A possible bin packing-like modeling would also deserve attention as often such models are known for being quite tight.

Our experiments on optimizing over the first Chvátal-Gomory closure showed very promising. Investigating the other closures (knapsack, elementary, MIR and/or disjunctive closures etc.) also deserves attention. As the number of variables remains still manageable, the concept of lift-and-project and extended formulation combined with the column generation would also be another research direction.

Symmetry is also another issues in the model which deserves attention. Our investigation in this area shows that there is no symmetry group for the FCLOP when $p \geq 3$, however a possible research avenue would be to investigate the advanced concepts in symmetry groups of polyhedra. Again such kind of symmetry-breaking inequalities must be generated on the fly rather than being directly added to the model.

Our emphasis in this thesis was on the polyhedral studies and exact solution methods. However, in most cases, for a real-life size instance of problem, the decision makers ask for a good quality solution, which can be obtained in a moderate computational time. In such cases, heuristics procedures —in particular those exploiting e.g. the information about the dual – such as

Lagrangian-based heuristics deserve attention.

Bibliography

- Ahmed, S., Shapiro, A., and Shapiro, E. (2002). The sample average approximation method for stochastic programs with integer recourse. *Submitted for publication*.
- Bai, L. and Rubin, P. A. (2009). Combinatorial benders cuts for the minimum tollbooth problem. *Operations research*, 57(6):1510–1522.
- Balas, E. and Christofides, N. (1981). A restricted lagrangean approach to the traveling salesman problem. *Mathematical Programming*, 21(1):19–46.
- Barahona, F. and Anbil, R. (2000). The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399.
- Beasley, J. E. (1993). Lagrangean heuristics for location problems. *European Journal of Operational Research*, 65(3):383–399.
- Belloni, A. (2001). Lagrangian Heuristics to Linear Ordering. pages 1–19.
- Belloni, A. and Lucena, A. (2004). Lagrangian heuristics for the linear ordering problem. *Metaheuristics: Computer Decision-Making eds. MGC Resende and JP de Sousa, Kluwer, Dordrecht*, pages 37–63.
- Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.
- Benders, J. (2005). Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science*, 2(1):3–19.
- Bertacco, L., Brunetta, L., and Fischetti, M. (2008). The Linear Ordering Problem with cumulative costs. *European Journal of Operational Research*, 189(3):1345–1357.

- Birge, J. R. (1985). Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007.
- Birge, J. R. and Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392.
- Bolotashvili, G. (1986). On the facets of the permutation polytope, communic. *Georgien Academy of Sciences*, 121:N2.
- Bolotashvili, G., Kovalev, M., and Girlich, E. (1999). New facets of the linear ordering polytope. *SIAM Journal on Discrete Mathematics*, 12(3):326–336.
- Bruglieri, M., Ehrgott, M., Hamacher, H. W., and Maffioli, F. (2006). An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints. *Discrete Applied Mathematics*, 154(9):1344–1357.
- Buchheim, C., Wiegele, A., and Zheng, L. (2010). Exact algorithms for the quadratic linear ordering problem. *INFORMS Journal on Computing*, 22(1):168–177.
- Campos, V., Laguna, M., and Martí, R. (1999). Scatter search for the linear ordering problem. *New ideas in optimization*, pages 331–339.
- Cao, J. X., Lee, D.-H., Chen, J. H., and Shi, Q. (2010). The integrated yard truck and yard crane scheduling problem: Bendersdecomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):344–353.
- Carøe, C. C. and Tind, J. (1998). L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83(1-3):451–464.
- Charbit, P., Thomassé, S., Yeo, A., et al. (2007). The minimum feedback arc set problem is np-hard for tournaments. *Combinatorics, Probability and Computing*, 16:01–04.
- Charon, I. and Hudry, O. (2007). A survey on the linear ordering problem for weighted or unweighted tournaments. *4OR*, 5(1):5–60.
- Codato, G. and Fischetti, M. (2006). Combinatorial benders’ cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766.
- Eades, P., Lin, X., and Smyth, W. F. (1993). A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47(6):319–323.

- Escudero, L., Guignard, M., and Malik, K. (1994). A lagrangian relax-and-cut approach for the sequential ordering problem with precedence relationships. *Annals of Operations Research*, 50(1):219–237.
- Fischetti, M. and Salvagnin, D. (2011). A relax-and-cut framework for gomory mixed-integer cuts. *Mathematical Programming Computation*, 3(2):79–102.
- Fischetti, M., Salvagnin, D., and Zanette, A. (2009). Minimal infeasible subsystems and benders cuts. *Mathematical Programming to appear*.
- Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18.
- Frangioni, A. (2000). Generalized Bundle Methods. pages 1–40.
- Garcia, C. G., Pérez-Brito, D., Campos, V., and Martí, R. (2006). Variable neighborhood search for the linear ordering problem. *Computers & operations research*, 33(12):3549–3565.
- Garey, M. R. and Johnson, D. S. (1979). Computers and intractability: A guide to the theory of np-completeness.
- Gavish, B. (1985). Augmented lagrangean based algorithms for centralized network design. *Communications, IEEE Transactions on*, 33(12):1247–1257.
- Geoffrion, A. M. (1974). *Lagrangean relaxation for integer programming*. Springer.
- Grötschel, M., Jünger, M., and Reinelt, G. (1983). Optimal triangulation of large real world input-output matrices. *Statistische Hefte*, 25(1):261–295.
- Grötschel, M., Jünger, M., and Reinelt, G. (1984). A cutting plane algorithm for the linear ordering problem. *Operations research*, 32(6):1195–1220.
- Grötschel, M., Jünger, M., and Reinelt, G. (1985a). Facets of the linear ordering polytope. *Mathematical Programming*, 33(1):43–60.
- Grötschel, M., Jünger, M., and Reinelt, G. (1985b). On the acyclic subgraph polytope. *Mathematical Programming*, 33(1):28–42.
- Guignard, M. (1998). Efficient cuts in lagrangean relax-and-cut schemes. *European Journal of Operational Research*, 105(1):216–223.
- Guignard, M. and Kim, S. (1987). Lagrangean decomposition: A model yielding stronger lagrangean bounds. *Mathematical Programming*, 39:215–228.

- Held, M. and Karp, R. (1971). The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1(1):6–25.
- Held, M. and Karp, R. M. (1970). The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6):1138–1162.
- Held, M., Wolfe, P., and Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical programming*, 6(1):62–88.
- Hoogeveen, J. and Velde, S. (1995). Stronger lagrangian bounds by use of slack variables: Applications to machine scheduling problems. *Mathematical Programming*, 70(1-3):173–190.
- Hungerländer, P. and Rendl, F. (2011). Semidefinite relaxations of ordering problems. *Mathematical Programming*, pages 1–21.
- ILOG, I. (2012). *Inc. CPLEX 12.5 User Manual*. Somers, NY, USA.
- Laguna, M., Marti, R., and Campos, V. (1999). Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computers & Operations Research*, 26(12):1217–1230.
- Lemarechal, C. (1975). An extension of Davidon methods to non differentiable problems. *Nondifferentiable Optimization*, pages 95–109.
- Lemarechal, C. (1978). *Nonsmooth optimization and descent methods*. International Institute for Applied Systems Analysis.
- Lemaréchal, C. (2001). Lagrangian relaxation. In *Computational Combinatorial Optimization*, pages 112–156. Springer.
- Lenzen, M. (2007). Structural path analysis of ecosystem networks. *Ecological Modelling*, 200:334–342.
- Leung, J. and Lee, J. (1994). More facets from fences for linear ordering and acyclic subgraph polytopes. *Discrete Applied Mathematics*, 50(2):185–200.
- Louveaux, F. V. and van der Vlerk, M. H. (1993). Stochastic programming with simple integer recourse. *Mathematical programming*, 61(1-3):301–325.
- Lucena, A. (1992). Steiner problem in graphs: Lagrangean relaxation and cutting planes. volume 21, pages 2–8.
- Lucena, A. (1993). Steiner problem in graphs: Lagrangean relaxation and cutting planes. volume 21, pages 147–154.
- Margot, F. (2010). Symmetry in integer linear programming. In *50 Years of Integer Programming 1958-2008*, pages 647–686. Springer.

- Martí, R. and Reinelt, G. (2011). *The linear ordering problem: exact and heuristic methods in combinatorial optimization*, volume 175. Springer.
- McLennan, A. (1990). Binary stochastic choice. *Preferences, Uncertainty and Optimality (Westview Press, Boulder, 1990)*, pages 187–202.
- Mitchell, J. E. and Borchers, B. (2000). Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In *High performance optimization*, pages 349–366. Springer.
- Moon, J. (1966). On subtournaments of a tournament. *Canad. Math. Bull*, 9(3):297–301.
- Mulvey, J. M. and Ruszczyński, A. (1995). A new scenario decomposition method for large-scale stochastic optimization. *Operations research*, 43(3):477–490.
- Pereira, M. and Pinto, L. M. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1-3):359–375.
- Polyak, B. T. (1969). Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9(3):14–29.
- Reinelt, G. (1985). *The linear ordering problem: algorithms and applications*, volume 8. Heldermann Berlin.
- Reinelt, G. (1993). A note on small linear-ordering polytopes. *Discrete & Computational Geometry*, 10(1):67–78.
- Schiavinotto, T. and Stützle, T. (2004). The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, 3(4):367–402.
- Schultz, R., Stougie, L., and Van Der Vlerk, M. H. (1998). Solving stochastic programs with integer recourse by enumeration: A framework using gröbner basis. *Mathematical Programming*, 83(1-3):229–252.
- Sherali, H. and Fraticelli, B. (2002). A modification of benders’ decomposition algorithm for discrete sub-problems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342.
- Sirdey, R. and Kerivin, H. (2007). Polyhedral combinatorics of a resource-constrained ordering problem part i: on the partial linear ordering polytope. *Rapport technique PE/BSC/INF/017912 V, 1*.
- van der Vlerk, M. H. (1995). *Stochastic programming with integer recourse*. PhD thesis, PhD thesis, University of Groningen, The Netherlands.