On the Directed Cut Polyhedra and Open Pit Mining

Conor Meagher

Doctor of Philosophy

School of Computer Science

McGill University

Montreal,Quebec

2010-08-31

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

STATEMENT OF ORIGINALITY

I certify that the research presented in this thesis only comprises of my original contribution to knowledge except where duly acknowledged and referenced. It contains no material that has been accepted in any form for another degree.

All of the original work presented in this thesis is at this point unpublished with the exception of the results from Chapter 9 which previously appeared in [61]. Certain results have also previously appeared in the series of COSMO Stochastic Mine Planning Laboratory annual research reports.

Much of the work in Chapters 3 through 6 on the directed cut polytope was done in collaboration with my supervisor David Avis.

The critical review of existing mining methods in Chapter 2 and the application of a developed optimization algorithm to a real world mining problem presented Chapter 9 was done in collaboration with another thesis supervisor Roussos Dimitrakopoulos.

In Chapter 5, the results of Theorem 27 were previously proved by Bondarenko and Uryvaev [17]. The statement of their theorem and proof is given in terms of the correlation polytope. We present an alternate proof of the same theorem in terms of the cut polytope and switching. This leads to an original polynomial time algorithm for solving optimization problems on the cut polytope under given criteria.

ACKNOWLEDGEMENTS

It is said that after your parents a Ph.D. supervisor has the greatest influence on your life. I had the privilege of having three supervisors during my time at McGill. My three dads were David Avis, Roussos Dimitrakopoulos and Bruce Reed. The combination of discrete optimization, mining and graph theory appearing in this thesis is not an accident and are a direct reflection of their influence and respective areas of expertise. David taught me much of what I know on discrete optimization and that you can take any problem and relate it to the cut polytope. I am greatly indebted to Roussos for introducing me to optimization problems and opportunities in the mining industry. Without Bruce I doubt I would have ever pursued this degree, while my focus drifted away from my original intent to focus on graph colouring Bruce taught me a lot of graph theory and probability before I changed my focus. Much of what I learnt from my supervisors falls into the category of life lessons. I feel wiser in many areas outside of academics for having known them.

During my time at McGill I was also blessed with a great set of peers. In particular, my 232 office mates Bohdan, Perouz, Victor and Teressa. The 109 office group Sean, Jamie, Andrew, Louigi and Nicolas, the Burnsider's Neil and Nic. My Ph.D. advisory committee and friends Adrian and Bruce Shep. The lunch group of Luc, Godfried and everyone that attended.

I would also like to thank the students and postdocs in the COSMO mine planning lab. The industry partners, their generous contributions of time and financial assistance were of great help. I would also like to thank BHP Billiton, in particular

Peter Stone and Mark Zuckerberg for letting me visit Melbourne and spending time with them.

Places and groups that meant a great deal to me during the past few years are Rue Violoncelle, Robichaud Office, the Claremont cougars, MCFC, the Lemmys and the Ghostryders.

Lastly, but primarily, I would like to thank my family and Sara for everything.

ABSTRACT

Many aspects of open pit mine planning can be modelled as a combinatorial optimization problem. This thesis reviews some existing mine scheduling methods and some of their short comings. Many of the problems are related to the partially ordered knapsack problem with multiple knapsack constraints. This is a special case of a maximum directed cut problem with multiple knapsack constraints on the arcs in the cut.

The major contribution of this thesis is the study of the directed cut polytope and cone, which are the convex hull and positive hull of all directed cut vectors of a complete directed graph, respectively. Many results are presented on the polyhedral structure of these polyhedra. A relation between the directed cut polyhedra and undirected cut polyhedra is established that provides families of facet defining inequalities for the directed cut polyhedra from the undirected cut polyhedra.

A polynomial time algorithm for optimizing over the undirected cut polytope is given for the special case ch an objective function has the same optimal value on two relaxations, the rooted metric polytope and the metric polytope. Projections of the directed cut polytope onto the arc set of an arbitrary directed graph are researched. A method known as triangular elimination is extended from the undirected cut context to a directed cut context.

A complexity result proving that the problem of selecting a physically connected maximum value set of blocks from a 2D grid is NP-hard is given. In the mining literature such a grid would be called a bench.

An implementation of a LP rounding algorithm known as pipage rounding is applied to a pushback design problem. This simple and efficient technique produces results within 6.4% for a real data set.

ABRÉGÉ

De nombreux aspects de la planification d'une mine à ciel ouvert peuvent être modélisés comme des problèmes d'optimisation combinatoire. La première partie de cette thèse passe en revue quelques méthodes de planification existantes dans la littérature et certaines de leurs lacunes. Plusieurs problèmes sont liés au "partially ordered knapsack" (POK) problème avec contraintes de type sac à dos. Il s'agit d'un cas particulier du problème de coupe maximale dans un graphe dirigé avec des contraintes de type sac-à-dos sur les arcs de la coupe.

La contribution majeure de cette thèse est l'étude du cône et du polytope des coupes dirigées, lesquels sont respectivement l'enveloppe convexe et l'enveloppe positive de toutes les coupes d'un graphe dirigé complet. Plusieurs résultats sur la structure polyèdrale des ces polyèdres sont présentés.

Une relation entre les polyèdres de coupes dirigées et les polyèdres de coupes non-dirigées est établie. Cette relation permet d'obtenir des familles de facettes définissant des inégalités valides pour les polyèdres de coupes dirigées à partir des inégalités valides et des facettes du polyèdre de coupes non-dirigées. Un algorithme polynomial pour le polytope des coupes non-dirigées est proposé dans le cas particulier d'une fonction objectif ayant la même valeur optimale pour deux relaxations, le polytope métrique enraciné et le polytope métrique. Les projections du polytope de coupes dirigées sur les arcs d'un graphe dirigé sont également étudiées. Une méthode de projection intitulée élimination triangulaire est généralisée du cas non-dirigé au cas dirigé.

Le problème qui consiste à sélectionner d'une grille 2D un ensemble de sommets connectés de valeur maximale est également étudié. Dans le contexte des mines, les sommets sont les blocs et la grille 2D est un banc. Un résultat de complexité établissant la NP-complétude de ce problème est présenté.

Un algorithme "page rounding" arrondit la solution de la relaxation linéaire a été implémenté pour résoudre le problème de conception de "pushbacks". Cet algorithme simple et efficace a été testé sur des données réelles et a permis d'obtenir des solutions très proches de la solution optimale (écart de 6.4% par rapport à des données réelles).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Outline

Scheduling the extraction of an open pit mine can be viewed as a combinatorial optimization problem with tens to hundreds of thousands of variables. What one would like to solve can easily be modelled as an integer program but due to the number of variables and constraints involved, current commercial integer program solvers will take too much time to solve the optimization problem to be practically useful. This thesis looks at the long term scheduling of an open pit mine and optimization problems related to it.

Chapter 2 is an introduction and review of work previously done on long term open pit mine planning and the related partially ordered knapsack problem. In particular, many results of this thesis are polyhedral in nature, we therefore focus much of the survey on valid inequalities and facets of the knapsack and partially ordered knapsack polytopes.

The major contribution of this thesis is the theory that is developed on the directed cut polyhedra. Chapter 3 reviews some previous work on the undirected cut polyhedra and semidefinite programming. Section 3.4 defines the directed cut polytope and cone. The rooted directed semimetric and directed semimetric polytope (resp. cone) are defined and shown to be relaxations of the directed cut polytope (resp. cone). Chapter 4 introduces some families of facet defining inequalities for the directed cut cone and polytope. A theorem relating facets of the undirected cut polytope and facets of the directed cut polytope is also presented. Operations on facet defining and valid inequalities of the directed cut cone and polytope such as zero-lifting, collapsing, permutations and switching are proved.

The problem of deciding whether there exists an integer optimum solution of the same value as that of a given fractional solution to an optimization problem is of fundamental importance. Very few useful results are known of this type, and Chapter 5 describes one of them that can be efficiently implemented. A result previously proved and stated in terms of the correlation polytope is proved in terms of the cut polytope and the switching operation. This alternate proof leads to a polynomial time algorithm for optimizing over the cut polytope in the case when optimizing over the rooted metric and metric polytope have the same value for a given objective function.

Chapter 6.1 investigates the projection of the directed cut polyhedra. The directed cut polyhedra are initially defined in terms of an underlying complete directed graph, but it is of interest to get a description of the polyhedra when projected onto different support graphs. An operation defined previously for the cut polytope known as triangular elimination that is a form of Fourier-Motzkin elimination combined with lifting is generalized to the directed cut polytope.

Future possible work on the directed cut polyhedra are described in Chapter 7. In particular, a conjecture on the characterization of DMET(G) is presented. The class of graph for which DMET(G) =DCUT(G) is also discussed in this chapter.

Chapter 8 returns to the computational complexity of the pushbacks design problem which was discussed in Chapter 2. Specifically, it is proved that if one wants to find maximum weight physically connected set of blocks in an orebody block model, the problem becomes NP-hard, even without a knapsack constraint. This result deals specifically with the geometric layout of an orebody block model, in contrast with other reductions which deal with a partially ordered knapsack problem involving a graph that can not be embedded into the orebody block model setting.

Some experimental results on a heuristic rounding procedure are presented in Chapter 9. The goal of this algorithm is to produce a pushback that strictly adheres to a knapsack constraint and chooses the cut-off grade dynamically. The algorithm described is implemented and run on an actual copper deposit.

A brief set of conclusions are presented in the final chapter. Followed by an appendix listing the vertices of $\text{RDMET}_3^\square$, $\text{DMET}_3^\square$, $\text{RDMET}_4^\square$ and $\text{DMET}_4^\square$.

While this thesis is laid out and prepared to be read from start to finish, the dissertation can be read in many different ways. If one is only interested in the theory of directed cut polyhedra, Chapters 3 through 7 will be of interest and can be read independently of the rest of the thesis.

Chapter 8 and 9 are largely independent of each other and the rest of the thesis. It is recommended that the reader read Chapter 2 until Section 2.3 before reading either Chapter 8 or Chapter 9.

If the reader is interested in valid inequalities for the knapsack polytope, the partially ordered knapsack polytope and the directed cut polytope, one could read Section 2.3, Chapter 4 and Chapter 6.

Lastly, if the reader is interested in our proof of the result of Bondarenko and Uryvaev [17] and how to optimize over the cut polytope in polynomial time when optimizing over the rooted semimetric and semimetric polytope have the same objective value, then reading Chapter 3 up to Section 3.4 and Chapter 5 will suffice.

CHAPTER 2
Survey of open pit optimization and the partially ordered knapsack problem

Open pit mine design and long-term production scheduling are a critically important parts of mining ventures and deal with the efficient management of cash flows in the order of hundreds of millions of dollars. Mine design and production scheduling determines both the economic outcome of a project and the technical plan to be followed from mine development to mine closure. It is an intricate and complex problem to address due to its large scale, the unavailability of a truly optimal net present value (NPV) solution, and the uncertainty in the key parameters involved (geological, mining, financial, and environmental).

The optimization of open pit mine design consists primarily of defining the "ultimate pit limits" which define what will eventually be removed from the ground, and dividing up the pit into manageable volumes of materials often referred to as pushbacks, cutbacks, or phases. Pushbacks, as they are referred to herein, allow for the mine designer to develop short term schedules for a smaller more manageable data set. They also contribute to the yearly production schedules so one can apply an economic discount rate when calculating the NPV of the mine. Typically, an orebody model of what is predicted to be in the ground is produced through one of various techniques ([31], [32], [39]). The resulting orebody model is typically represented as a block model, where the physical area of the deposit is broken up into rectangular blocks of a given size and each block has a predicted ore content. From this orebody model, optimization techniques are used to produce the ultimate pit. The ultimate pit is the maximum valued pit possible that obeys slope and physical constraints. Pushbacks are produced from the sections of the orebody model that remain within the ultimate pit limits.

Traditional production scheduling methods are performed using pushbacks designed to maximize the economic value, or metal content within each incremental pushback in a greedy fashion. There are major issues with the existing pushback design methods that lead to sub-optimal production schedules including: (a) not considering requirements in grade and ore quality parameters; (b) ignoring the in-situ grade uncertainty; (c) large variations in size of the pushbacks, or so-termed "gap" leading to impractical results; (d) not considering discounting during the optimization and assuming that a greedy approach will maximize discounted value.

It should be stressed that the total NPV that can be generated from a mining operation strongly depends on the pushback design that guides the extraction sequence of ore and periodical metal production. It is impossible to generate a truly optimal production schedule using sub-optimally designed pushbacks. Production schedules based on sub-optimal pushback designs fail to produce the maximum and optimal NPV of a mining project (eg. [39], [40], [71], [74]).

A popular technique for producing pushbacks is to take an algorithm that produces an ultimate pit and run it multiple times over the orebody model where the economic block values are scaled down by a series of decreasing factors, values of $\lambda_1, ..., \lambda_t$ where $0 \leq \lambda_i \leq 1$ are used to replace the value v of a block with $v - \lambda_i v$.

The result is a series of nested pits, small pits are produced when the orebody model block values are scaled down by a large factor, as the factor gets smaller, larger and larger pits are produced until the ultimate pit is produced when the factor is 0. The series of nested pits produced gives the mine designer possible pushback options. This is the approach used by the Lerchs-Grossman algorithm [59] implementation of Whittle [82], a series of heuristically discounted pits is produced in a greedy fashion until it is no longer profitable to consider any further pits. The final pit is used as the ultimate pit limits.

The scaling approach suffers from the problem of having the pushback sizes produced differing erratically. A series of small pit increments followed by a very large pit may be generated. A simple example when this would happen is if there was a large section of ore beneath a large amount of waste. It would not be feasible to mine anything until the scaling factor reaches some threshold value and then a large pit with no incremental smaller pushbacks would be produced; such a situation is depicted in Figure 2–1. Large size differences between consecutive pushbacks that may render them impractical are often referred to as *gap problems* and are a common problem in developing designs that are feasible in an engineering sense, without manual "re-designing" which then has unknown effects on the optimization of the design.

Producing a series of pits in the fashion described above also suffers from the problem that the pit produced for a given factor may be disconnected and not one single pit. If one chooses the pushbacks strictly in this fashion, single pushbacks may have multiple sections that are physically far from each other, making them

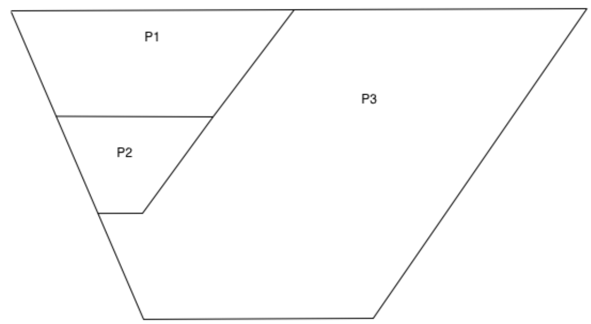


Figure 2–1: Schematic representation of an open pit design showing three pushbacks with gap problems

impractical. Ideally, a pushback should be one connected piece and not fragmented. A further problem with the technique described is that other geometric limitations open pit pushbacks must adhere to are not considered. This can include requiring the pit base be a convex shape and of a minimum width.

Existing algorithms for pushback and open pit optimization are typically designed to only consider one fixed orebody model. The traditional single estimate assessment for pertinent parameters, including project net present value (NPV), expected cash flows, metal quantities, and expected production costs. Two major flaws of traditional optimization in mine design and planning are: (i) inputs are assumed certain while they are not, thus uncertainty from geological, mining and market sources is not accounted; and (ii) conventional mathematical models cannot handle input uncertainty models, viz a viz stochastically described inputs. Consequences of these flaws are demonstrated in an example [32] where mine design optimization in an open-pit gold mine shows that the consideration of geological uncertainty predicts a NPV that is considerably less than that forecasted via conventional modelling. The difference arises from significant departures in expected cash flows between the traditional single-block orebody estimates and stochastic models, and demonstrates potentially misleading results from combining traditional orebody models with complex non-linear optimization algorithms. Furthermore, this example highlights the conceptual and computational inadequacy, and technological limits of mine design and production scheduling technologies currently used, when optimizing under uncertainty. With advances in stochastic simulation techniques, new algorithms are needed to handle multiple equally likely orebody model realizations. The techniques should provide a robust optimization over all orebody models and not just perform well in expectation.

In Section 2.1 some popular traditional methods for pushback design are reviewed and how they address the issues introduced is discussed.

Throughout this thesis many references will be made to both directed and undirected graphs. To avoid confusion, we will adopt the notation ij to represent a directed arc from node i to node j in a directed graph. When necessary the notation $(i, i+1)$ may be used to represent a directed arc from node i to node $i+1$. The notation i, j will represent an edge in an undirected graph between nodes i and j.

2.1 A review of existing methods

2.1.1 The Lerchs-Grossman algorithm

The most well established procedure in practice for producing ultimate pit limits is the Lerchs-Grossman (L-G) algorithm [59]. This algorithm constructs a directed graph $G = (V, A)$ where each node $v \in V$ represents a block in the orebody model. A weight w_v equal to the cost of removing a waste block v or equal to the profit of processing ore block v is associated with each node $v \in V(G)$. The arcs of G represent the slope constraints of the open pit mining problem. For a given deposit, a set of slope angles will be determined by engineers so the pit walls do not collapse from being too steep or mine extra waste from being too shallow. These are determined based on characteristics of the geology and engineering guidelines. If a block j must be physically removed prior to block i then G will contain a directed path from node i to node j, we denote an arc from node i to node j in $A(G)$ as ij.

The L-G algorithm finds a maximum weight *graph closure*, a subset of nodes $S \subseteq V(G)$ such that no arc $a \in A(G)$ has a tail in S and head in $V(G) \setminus S$ and $\sum_{v \in S} w_v$ is maximized over all such S. The graph closure represents the traditional ultimate pit limits. By the construction of the arc set $A(G)$ and the definition of a graph closure it is straight forward to see that every graph closure can be physically mined and will adhere to the engineering slope constraints.

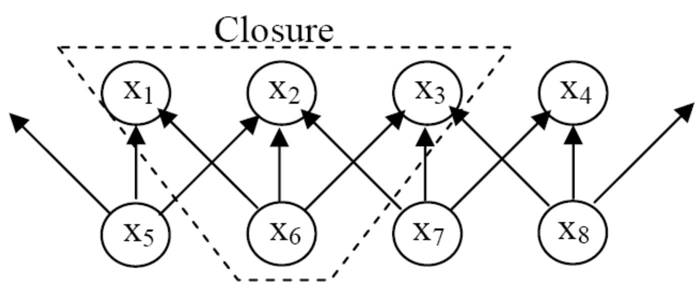


Figure 2–2: A depiction of a graph closure, the x_i labelled nodes of the graph represent blocks in our block model

The algorithm begins by adding a dummy root node s to the graph G with arcs directed from s to every node in G. When referring to a tree or spanning tree of our graph, edges are considered undirected. The mass of a branch is the sum of the weights of the nodes in the branch. An arc is termed *strong* if it is a downward arc (towards s in the tree) that supports a mass that is strictly positive, or an upward arc that supports a mass that is non-positive. Otherwise, the arc is termed weak. A spanning tree rooted at s is *normalized* if the only strong arcs it contains are adjacent to the root s.

The L-G algorithm produces a series of normalized trees until one of the trees corresponds to a graph closure. It can be shown that this graph closure is in fact the maximum graph closure.

2.1.2 Seymour's parameterized pit limit algorithm

Fred Seymour [74] modified the Lerchs-Grossman algorithm to incorporate what is known as parameterization. Open pit parameterization produces maximum valued pits as a function of another parameter (where this parameter is defined for each block in our orebody model). Seymour chooses pit volume as the parameter to parameterize in his paper. If one was to plot the economic pit value vs. chosen parameter value as shown in Figure 2–3, Seymour's algorithm can return precisely those pit designs that lie on the upper convex hull of this point set. If the upper convex hull is well defined and feasible pits exists at or around the desired parameter values (pit sizes in Seymour's paper) then one can use such pits to develop pushbacks that don't suffer from non-uniform sizes.

The algorithm follows the approach of the L-G method. But instead of producing one final tree, representing the maximum graph closure-ultimate pit, it produces a set of branches, where a branch's strength is its value divided by the sum of the volume of the blocks in the branch. A threshold value is used to determine if a branch is “strong” or “weak”, by altering the threshold, a series of nested pits can be produced. All strong branches together form the normalized tree that L-G’s algorithm returns when the threshold is set to its minimum value.

While this approach can provide some useful results in practice [74], if the pits that lie on upper convex hull are far apart in terms of size then gap problems will continue to persist. The algorithm will not return pits of the desired size, this situation is depicted in Figure 2–3. Seymour’s algorithm would return the two nested pits on the upper convex hull, but will not return any of the potentially useful designs that lie between these two sizes that lie below the convex hull.

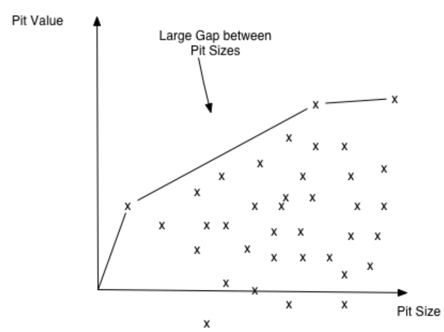


Figure 2–3: A plot of the upper convex hull of the pit value vs. pit size shows a large gap between possible pit sizes

2.1.3 Network flow approaches

Following the success of modelling the ultimate pit problem as a maximum graph closure problem, Picard [69] showed how to find the maximum closure of a graph by using a network flow algorithm. This allows one to use known efficient algorithms for finding a maximum flow and use it to find the ultimate pit limits.

The maximum flow problem can be stated as: given a directed graph G, with capacities on the edges, a source node s, and a sink node t; one wants to know the maximum amount of flow that can travel from the source node s to the sink t without violating the capacity constraints on the edges. An arc ij with a capacity of c_{ij} can send at most c_{ij} units of flow from node i to node j. The flow must also obey the conservation of flow constraint at each node in $V(G) - \{s, t\}$, which states that the flow into a node is equal to the flow out of the node, ie.

$$\sum_{i:ij \in A(G)} x_{ij} \quad = \quad \sum_{k:jk \in A(G)} x_{jk} \qquad \forall j \in V(G) \setminus \{s, t\}. \tag{2.1}$$

A minimum cut is the set of arcs with their tail in a subset of nodes $S \subseteq V(G) - \{t\}$ containing s and head in $V(G) - S$ such that the sum of the capacities in the cut is the minimum over all such cuts. Since any flow going from s to t is constrained to be at most the capacity of a minimum cut, it follows that the maximum s-t flow is at most the size of a minimum cut. It can be shown that these two quantities are in fact equal. Given a maximum $s-t$ flow one can find a minimum cut by starting at t and doing a depth first search of the edges that are saturated (the flow over the arc equals the arc capacity) in the reverse direction from the sink.

Picard [69] showed that given a graph G, on which one wishes to find a maximum closure, one can construct an auxiliary graph G' where the minimum cut in G' corresponds to the maximum closure of G. Construct G' by taking a copy of G and adding two new nodes, a source s and a sink t. Add arcs from s to every node that has positive weight in G and add arcs from every negative weight node to t. Give the edges of the form sv a capacity c_{sv} equal to the weight of v in G and give arcs of the form vt a capacity c_{vt} equal to the absolute value of the weight of v in G. Give all other arcs, the arcs that correspond to slope constraints, infinite capacity.

Consider the small example of a vertical cross-section of an orebody model in Figure 2–4. Figure 2–5 depicts the construction of the network from the orebody model in Figure 2–4 the unlabelled arcs have infinite capacity. A minimum cut in G' will have only arcs directed from s or to t, since all other arcs have infinite capacity. In the context of an orebody model, one can think of a minimum cut consisting of arcs directed to the ore that is left in the ground and arcs from the waste that is left in the pit. The infinite capacity arcs ensure that slope constraints are maintained. Since the orebody model is finite, minimizing the value of ore left outside the pit plus the cost of the waste left inside the pit is equivalent to maximizing the ore inside the pit minus the waste inside the pit. Figure 2–6 shows the minimum cut in our example, the dashed arcs correspond to the arcs in the minimum cut.

1	-2	-2	-2	-2
	5	6	-3	
		4		

Figure 2–4: Vertical cross section of an orebody model

One can formulate the minimum cut problem as a linear program (LP) in such a way that the constraint matrix is totally unimodular. This implies that one can get an integral solution by solving the LP, which can be done in polynomial time.

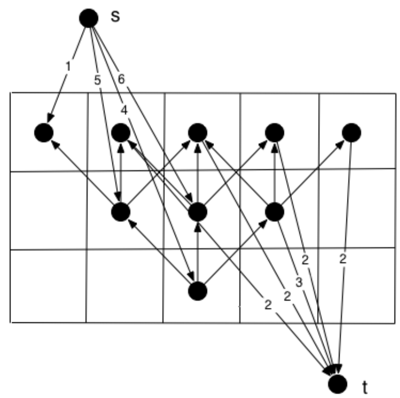


Figure 2–5: Network constructed from the orebody model in Figure 2–4, s is the source and t is the sink

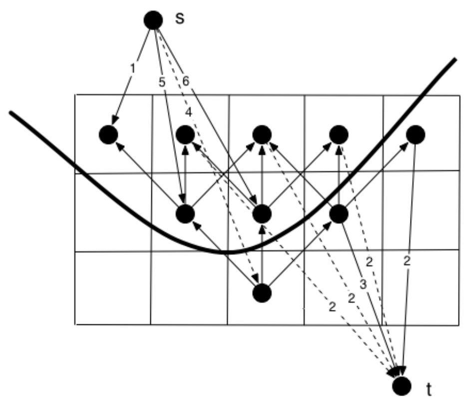


Figure 2–6: A minimum cut of the network show in Figure 2–5

Hochbaum and Chen [53],[52] showed that the L-G algorithm can be used as a network flow algorithm. From the series of normalized trees they showed how one could obtain an optimal network flow. They also analyzed the runtime of the L-G algorithm and improved it by scaling techniques (different from those used to generate pushback designs) to show that L-G can be implemented to run in $O(mn \log n)$ time, where m and n are the number of arcs and nodes in the constructed graph respectively. The network flow algorithm they developed is known as the pseudoflow algorithm. Muir [63] implemented the pseudo-flow algorithm and found it more efficient than the L-G algorithm in practice.

Gallo, Grigoriadis, and Tarjan [37] developed a way to use a network flow algorithm to produce a series of parameterized minimum cuts. This process returns the series of pits that are on the upper convex hull of economic pit value vs. the chosen parameter, the same set of pits that Seymour's algorithm can return. This process can be used to generate all the pits on the upper convex hull with very little additional computation. These possible pit designs, however, will suffer from the same gap issues as those returned by Seymour's algorithm.

2.1.4 Dagdelen-Johnson Lagrangian parameterization

In [22], Dagdelen and Johnson formalized the process of parameterization in the context of Lagrangian relaxation. Lagrangian relaxation is a process where a troublesome constraint is removed from the LP and placed in the objective. In the context of an open pit optimization problem, the technique applied to the problem of finding a pit of a fixed tonnage is shown. This can be done by modelling the ultimate pit as a LP, with the added constraint that the number of blocks in the pit is a fixed amount, say b:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} c_i x_i \\
s.t. \quad & x_j - x_i \leq 0 \qquad for\ (v_i, v_j) \in A(G) \\
& \sum_{i=1}^{n} x_i = b \\
& x_i \in \{0,1\} \qquad for\ i = 1, .., n
\end{aligned}$$

If the constraint $\sum_{i=1}^{n} x_i = b$ is removed this system is totally unimodular and the LP relaxation gives an integral solution and can be solved efficiently by the simplex method. However, the constraint $\sum_{i=1}^{n} x_i = b$ ruins the total unimodularity of the constraint matrix and it is unlikely that the LP relaxation will give an integral solution, making the IP much more difficult to solve efficiently. The Lagrangian relaxation of this problem would be to place this constraint in the objective along with a penalty factor $\lambda \geq 0$ for violating it. The new IP would be:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} (c_i - \lambda) x_i - \lambda b \\ s.t. \quad & x_j - x_i \leq 0 \qquad for\ (v_i, v_j) \in A(G) \\ & x_i \in \{0, 1\} \qquad for\ i = 1, .., n \end{aligned}$$

This IP is totally unimodular once again thus by relaxing the integrality on the x_i's one can solve it efficiently. Since one fixes the penalty λ and b is fixed λb is a constant and can be removed from the objective function. It is straight forward to see that the problem being solved is the ultimate pit limit problem where the economic value of the orebody model is scaled down by a constant factor λ, since each block i has economic value $(c_i - \lambda)$ in the LP. Choosing λ to be zero this is equivalent to finding the ultimate pit limits. As λ gets larger one can expect to get smaller and smaller pits. One can therefore view the procedure of finding nested pits by Dagdelen and Johnson's Lagrangian Parametrization as an equivalent procedure to that of scaling the orebody model value and running the L-G algorithm to get a series of nested pits. It therefore suffers from the same gap problems as those discussed previously. Choosing appropriate values of λ is not always straight forward either, it may take quite a bit of time to try to find a value of λ to produce pits close to the desired tonnage, and it might not even be possible to produce a pit of the desired size with this technique.

2.1.5 IP formulations

Due to technical and engineering limits there are many constraints that should be considered that intrinsically limit the size of a pushback based on its period of extraction [70]. Two such constraints are milling constraints and extraction capacity constraints. The mill should typically be fed a certain minimum and maximum quantity of ore. Also, constraints on the number of trucks can limit the amount of ore/waste that can be mined in a given period. These constraints can often be modelled as *knapsack constraints* which have the form:

$$\sum_{i \in V(G)} a_i x_i \quad \leq \quad b \tag{2.2}$$

where $b > 0$ and $a_i \geq 0$ for all $i \in V(G)$.

Since efficient algorithms exist to find optimal pits without knapsack constraints, we would like to know if an efficient algorithm exists with these restrictions. The problem of finding a graph closure with a knapsack constraint will be discussed further in Section 2.3.1.

If one considers the problem of finding an optimal pit with only the restriction that the pit must be connected (one single entity) it can be shown that this problem becomes NP-hard. We present this result in Chapter 8.

One approach [77] to solve these large IPs is to aggregate blocks together to decrease the number of variables in the IP. Doing this in a naive fashion can alter the shape of the ultimate pit that is produced. Taking the average of a set of blocks tends to increase the small values and decrease the large values of the blocks in the orebody model which leads to dilution. This can have a dramatic effect on the feasibility study of a mine, and has the same effect as what is known in mining literature as selectivity [72].

2.1.6 Fundamental tree algorithm

An approach for combining blocks together known as the fundamental tree algorithm was introduced by Ramazan [71], [70]. The fundamental tree method combines blocks in such a way that the ultimate pit produced on the combined blocks is the same as that produced if the blocks were not combined together. The approach decreases the number of blocks, which in some cases makes solving integer programs feasible for larger volume mines.

Since the number of decision variables has decreased in the IP formulation, one can put more constraints into the IP and still have efficient run times. The fundamental tree algorithm combines together blocks into a set of the so-calleds *fundamental trees*, which is a set of blocks where:

1. the blocks can be profitably mined,
2. the blocks obey the slope constraints and,
3. there is no proper subset of the chosen blocks that meets 1 and 2.

The fundamental tree method is very similar to the pit parameterization method of Seymour. One could consider the "branches" produced by Seymour's algorithm as fundamental trees, then use these branches as the "ore" variables in an LP formulation. Ramazan chooses to try to minimize the size of the trees by requiring that no proper subset of a tree can be profitably minded, the branches produced by Seymour's method will combine branches together as long as the ratio of value over tonnes increases, resulting in larger trees.

The problem with the technique of treating large branches or trees as single binary variables in an IP formulations is that one often wishes to have constraints due to milling, blending and transportation requirements that aren't considered in the process of producing the combined decision variable. Larger fundamental trees allow the IP formulation to be solved more efficiently but will affect selectivity in terms of constraints such as blending. There is no clear way of limiting the size of the fundamental tress produced. Often too many fundamental trees are produced and the IP formulation is still too large to be solved in practice.

2.2 Further limitations and more advanced algorithms

In most common practice, economic discounting is only heuristically used at the time of pushback design optimization. Nested pits are created in a greedy fashion so that one tries to produce a series of pits where the value of a pushback divided by its volume is always greater then a future pushbacks economic value divided by its volume [82]. Tolwinski and Underwood [81] developed an algorithm that explicitly uses discounting in schedule design but provides only heuristic solutions due to the long runtime required to reach optimality on a large mine. If one wishes to apply a discount rate of d to the constrained pushback design problem over p periods and have constrained pushbacks of size at most b, the problem can be formulated as the following integer program:

$$
\begin{aligned}
\max \quad & \sum_{k=0}^{p-1}\sum_{i=1}^{n}(1+d)^k c_i x_{i,p} \\
\textit{subject to} \quad & x_{i,l} - \sum_{k=1}^{l} x_{j,k} \leq 0 \qquad \forall i, l = 0..p-1 \textit{ and } ij \in A(G) \\
& \sum_{k=0}^{p-1} x_{i,k} \leq 1 \qquad \forall i \\
& \sum_{i=1}^{n} x_{i,k} \leq b \qquad \textit{for } k = 1..p \\
& x_{i,k} \in \{0,1\} \qquad \forall i = 1..n,\ k = 0..p-1
\end{aligned}
$$

This IP formulation would take too long to solve but it does define the pushback design objective that would optimize the pits NPV. An algorithm that solves the constrained pushback design problem for one pit could be used multiple times in a greedy fashion to obtain a series of pushbacks, however, it is easy to construct examples where it is not always optimal in terms of NPV to apply this greedy technique. To optimize the NPV one needs to consider the design of all pushbacks and discounting at the same time.

Further limitations of existing methods include using pre-determined cut-off grades. A cut-off grade defines what is determined to be waste or ore (and in more complex models, sent to the stockpile). Cut-off grades will often vary from pushback to pushback depending on the period of extraction. Often a pushback design algorithm is run for a specific cut-off grade and the process is iteratively repeated with alternate cut-off grades heuristically until a given set is determined as the best.

Tachefine and Soumis [78], [79] formulated a multi-period pushback design optimization problem as an integer program, where each period corresponding to a pushback had a knapsack constraint. Under this model, the economic value of a pushback could be appropriately economically discounted for the period of extraction. They used a set of Lagrangian multipliers, one for each period/pushback and then used a search optimization algorithm like steepest descent to try to find the best Lagrangian multipliers. The solutions produced would violate the knapsack constraints by a small amount, they then made the produced schedules feasible by employing a either a tabu search heuristic or one of two greedy discarding methods they developed.

Akaike [1] developed a similar algorithm that used an extension of the Lagrangian relaxation approach with a network flow formulation in which the ability to have a dynamic cutoff grade was incorporated. Choosing appropriate values of the Lagrangian multipliers to adhere to specific constraints is again done by a steepest-descent or equivalent algorithm.

More recent work of Bienstock and Zuckerberg [15] investigate more advanced techniques of using Lagrangian relaxation methods to solve optimization problems that can be formulated as linear programs where the removal of a small number of constraints produces integral optimal solutions. They have applied their techniques specifically to mine scheduling problems, as the removal of the knapsack type constraints in a typical IP formulation of POK problem leaves a totally unimodular system.

With advances in orebody modelling through stochastic simulations, one would like a set of tools that could optimize an open pit design over a set of multiple realizations of the orebody model simultaneously. If one wanted to optimize the over the averaged values of each block to obtain a single orebody model and optimize the design over that model, much of the information contained in the multiple realizations would be lost. Maximizing the expected value of a design over all realizations is not necessarily the best technique either, situations occur when such an approach can yield designs with a high NPV on a few realizations but a very poor NPV on most. One would like a robust procedure that would perform well over almost all realizations. To this end, the information from each realization must be maintained and techniques to handle the large amount of data and produce designs that perform well for most realizations must be developed.

Godoy and Dimitrakopoulos [40] developed an optimization algorithm based on producing a schedule for each simulation through traditional techniques. They then use a simulated annealing algorithm to produce schedule that optimizes NPV and penalizes deviation from production targets. While their algorithm has been shown to produce positive results, this approach suffers from having to produce many different schedules, one for each simulation, prior to running the simulated annealing algorithm. It also doesn't address many of the additional problems with traditional techniques, like dynamic cut-off grade optimization.

Dimitrakopoulos and Ramazan [33] used a stochastic integer program model to solve the problem of having to produce the initial set of schedules one at a time. Their method constructed a large integer program from the multiple simulations and attempts to find a schedule that minimizes the sum of deviations from production targets in each simulation. This method produced positive results as well, however the integer program formulations produced tend to get too large to be practically useful for a large deposit.

A further limitation of existing methods not yet discussed is connectivity. Ideally a pushback design should be physically connected. A pushback that has parts that are physically disconnected is not practical from a mining perspective as moving the mining equipment can be a costly and time consuming operation. In Section 8 we present a complexity result related to finding a connected set of blocks in a geometry that arises in the open pit mining problem. In the work of [21], [3] the geometric problem of finding an optimal specific shape on a 2D grid is investigated. These results can be thought of as applying to one level or bench of a mining problem.

Many of the advanced techniques cited in this section have produced positive results. They all, however, do not necessarily solve the optimization problems to optimality in a reasonable amount of time. Further, they often use heuristics and assumptions that don't reflect the exact problem to produce feasible schedules. None of the advanced methods described solve all the issues addressed simultaneously. They tend to optimize different aspects of the problem separately and find ways to iteratively try and improve solutions until some sort of convergence is reached. Traditional methods of optimizing also presume that the ultimate pit limits are known prior to the pushback design. These piecewise approaches can lead to suboptimal solutions. Recently, the mining industry has become interested in the area termed global optimization [83]. This global optimization can be viewed as removing the piecewise optimization steps from the mine planning process.

With the many different types of open pit mine scheduling and constraints involved, it would be nice if an algorithm could solve a generic IP formulated for the POK problem with multiple knapsack constraints. While the POK formulation can't encompass all of the problems it can be used to model most with the exception of mining width and connectivity.

2.3 Knapsack and partially ordered knapsack polytopes

The IPs that have been described thus far can be thought of as a set of precedence constraints in combination with a set of knapsack type inequalities. The mining methods reviewed up to this point have mainly used two different techniques, either Lagrangian relaxation (scaling) or solving IPs through block amalgamation. While these are the traditional approaches of the mining community, the problem of solving knapsack type problems with precedence constraints has been studied outside the realm of mining optimization.

The combination of precedence constraints with a knapsack inequality is known as either a *partially ordered knapsack problem* (POK) or a *precedence constrained knapsack problem* (PCKP) in scheduling optimization. It is known as the maximum weight ideal (MWI) problem in poset theory [34]. It can also be viewed as a subcase of the densest k-sub-hypergraph problem from graph theory.

2.3.1 Knapsack problems

One of the classic NP-complete problems from Karp's [55] 21 NP-complete problems is the subset sum problem. The subset sum problem is: Given a set of integers, does the sum of some nonempty subset equal zero? This can be generalized to the $0-1$ knapsack problem; given a set of weights w_i and a set of values v_i for $i = 1, ..., n$ can you choose a subset S of $N = \{1, ..., n\}$ such that:

$$\sum_{j \in S} v_j \geq K \tag{2.3}$$

$$\sum_{j \in S} w_j \leq b \tag{2.4}$$

$$\tag{2.5}$$

for a given value of b and K. The optimization $0 - 1$ knapsack problem is given by replacing constraint (2.3) with:

$$\max \quad \sum_{j \in S} v_j. \tag{2.6}$$

The $0 - 1$ knapsack problem has been studied extensively. If the weight W is polynomial in the size of n there exists a dynamic programming algorithm that can solve the problem in polynomial time, $O(nW)$. For larger W, a fully polynomial time approximation scheme (FPTAS) is known.

While there are many algorithms for and variations of the $0 - 1$ knapsack problem, our focus will be on polyhedral properties and results. In the rest of this section and in Section 2.4 we will describe past polyhedral work on the knapsack and POK polytopes. The study of the knapsack polytope is a vast area of research as the knapsack inequalities play a pivotal role in commercial integer program solvers, we focus primarily on results on the knapsack polytope that have been extended to the POK knapsack.

The knapsack polytope is the convex hull of all $0-1$ points $x \in \{0,1\}^{|N|}$ satisfying a given linear inequality of the form:

$$\sum_{j \in N} a_j x_j \leq b \tag{2.7}$$

where the a_j's and b are non-negative.

A subset $S \subseteq N$ is called a *cover* of inequality (2.7) if:

$$\sum_{j \in S} a_j \quad > \quad b.$$

Clearly, not all x_j's can be 1 for $j \in S$ as this would violate (2.7), which implies that for any x in the knapsack polytope and cover S:

$$\sum_{j \in S} x_j \quad \leq \quad |S| - 1. \tag{2.8}$$

Inequality (2.8) is known as a *cover inequality.* If a cover S has the property that for any proper subset $T \subset S$ the following inequality is satisfied:

$$\sum_{j \in T} a_j \quad \leq \quad b$$

then S is a *minimal cover.* Minimal cover inequalities are used extensively in cutting plane algorithms for solving general integer programs with knapsack type inequalities.

These inequalities can be strengthened by a technique known as *lifting.* Lifting, which was originally introduced by Gomory [42], takes a valid inequality and constructs a valid inequality in a higher dimensional space. There are many variations and types of lifting, lifting for the knapsack and POK problems will be discussed here, in Chapter 4 lifting will be applied to the cut and directed cut polyhedra.

For a subset $S \subset N$ of the possible knapsack items, a lower dimensional knapsack polytope can be defined as:

$$Knap^S = \{x \in \{0,1\}^{|S|} : \sum_{i \in S} a_i x_i \leq b\}$$

Lifting takes a valid (or facet) inequality of such a lower dimensional knapsack and constructs an inequality valid for the original knapsack polytope.

For knapsack cover inequalities, the *lifted cover inequality* for a minimal cover $C \subseteq N$ has the form:

$$\sum_{i \in C} x_i + \sum_{i \in N \setminus C} \alpha_i x_i \quad \leq \quad |C| - 1. \tag{2.9}$$

Choosing the values of the α_i's, which are non-negative integers, is known as the lifting problem. A generalized lifting method for 0-1 knapsack problems was developed by Wolsey [87]. This method generalized earlier work on lifting inequalities for 0-1 knapsack problems (see [9], [49], [66], [64] and [86]).

Given a minimal cover C, let $X = \{i \notin C : a_i > a_j \ \forall j \in C\}$, the inequality:

$$\sum_{j \in C \cup X} x_j \quad \leq \quad |C| - 1 \tag{2.10}$$

is known as an *extended cover* inequality and is a simple example of a lifted cover inequality that can be computed quickly.

Lifting cover inequalities can defined more generally. Given an ordering $\pi(1), ..., \pi(|N| - |C|)$ of the elements of $N \setminus C$ the lifting coefficients $\alpha_{\pi(j)}$ can be computed by solving the problems:

$$\begin{aligned}
\alpha_{\pi(j)} \quad \leq \quad & |C| - 1 - \max\{\sum_{i \in C} x_i + \sum_{i=1}^{j-1} \alpha_{\pi(i)} x_{\pi(i)}\} \qquad (2.11) \\
s.t. \qquad & \sum_{j \in C} a_j x_j + \sum_{j=1}^{i-1} a_{\pi(j)} x_{\pi(j)} \leq b - a_{\pi(j)} \\
& x_i \in \{0,1\} \quad \forall j \in C \\
& x_{\pi(i)} \in \{0,1\} \quad \forall i = 1, ..., j-1
\end{aligned}$$

Choosing the $\alpha_{\pi(j)}$ such that the inequality (2.11) is satisfied produces a valid inequality for the knapsack polytope. If inequality (2.11) is satisfied with equality then inequality (2.9) defines a facet of the knapsack polytope. This gives a way of constructing valid (and facet-inducing) inequalities from the cover inequalities. As the inequality that arises from the lifting operation is dependent on the ordering π, the choice of π can greatly affect how useful the lifted inequality is for solving an optimization problem on the knapsack polytope. Not only are there $2^{|N \setminus C|}$ ways of choosing π but solving the lifting maximization is equivalent to solving a knapsack problem on a slightly smaller dimensional polyhedra [50].

There have been many experiments testing how effective using cutting plane techniques based on the lifted cover inequalities are. They have been found to work well in practice. There are many papers investigating methods of generating them efficiently. See [46] for a thorough review.

2.4 POK polytope

The cover inequalities mentioned above have been generalized to the POK polytope, the convex hull of all $0-1$ points satisfying a given knapsack constraint and satisfying all precedence constraints. A node i of the precedence graph is called a *predecessor* of a node j if there exists a directed path from i to j in $A(G)$ and j is referred to as a *successor* of i. Define the set P_i to contain i and all predecessors of node i. For a set of nodes $C \subseteq V(G)$, let $P(C) = \{i : i \in P_j \; for \; some \; j \in C\}$, so $P(C)$ is the set of all predecessors of nodes in C unioned with the set of nodes C. Note that the POK literature uses the opposite direction to imply precedence from that used thus far.

In the POK literature ([16], [68], [18], [58]), two nodes i, j are called *incomparable* if $i \notin P_j$ and $j \notin P_i$. In the same vein, a set S is called incomparable, if for all $i, j \in S$, i and j are incomparable. There are two different definitions of a *cover* $C \subset V(G)$ for the POK problem. Park and Park [68] define C to be an *induced cover* if C is incomparable and $\sum_{j \in P(C)} a_j > b$ and C is a *minimal induced cover* if $\sum_{j \in P(C \setminus \{i\})} a_j \leq b$ for each i.

An alternate definition of an induced cover is proposed by Boyd [18], in which $C \subset V(G)$ be a minimal induced cover if $P' = P(C) \setminus j$ satisfies $\sum_{i \in P'} a_i \leq b$ for all $j \in C$. Using both of the defined induced covers the inequality:

$$\sum_{i \in C} x_i \quad \leq \quad |C| - 1 \tag{2.12}$$

is valid for the POK polytope.

If $Q \subseteq \mathbb{R}^{n+m}$ is a polyhedron where $(x, y) \in Q$ means $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ then the *projection* of Q onto the space $\mathbb{R}^n$ corresponding to x is:

$$Proj_x(Q) \quad = \quad \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m such \; that \; (x, y) \in Q\}.$$

Boyd goes on to prove structural polyhedral results on the projection of the POK polytope. Let POK(C) $= Proj_{P(C)}(POK)$ be the projection of the POK polytope onto the space indexed by the nodes of $P(C)$.

Let $H(C)$ be the set of nodes $i \in C$ such that i doesn't have any successors in C. The following theorem appears in [18]:

Theorem 1 ([18]) *Given any minimal cover C the constraint:*

$$\sum_{i \in C} x_i \leq K - 1 \tag{2.13}$$

is a facet of POK(C) if and only if $\cap_{\{S \subseteq H(C) : |S| = K-1\}} P(S) = \emptyset$.

A further class of valid inequalities is obtained from $(1, k)$-configurations. A set $C \cup \{t\} \subseteq V(G)$, $t \notin C$ is a *(1,k)-configuration* if:

- the items of $C \cup \{t\} \subseteq V(G)$ are incomparable
- $C \cup \{t\}$ is a cover with $\sum_{i \in P(C \cup \{t\}) \setminus \{t\}} a_i \leq b$

  and
- $Q \cup \{t\}$ is a minimal cover, for all $Q \subseteq C$ with $2 \leq |Q| = k \leq |C|$.

The $(1, k)$-configurations for precedence constraints are a straight forward generalization of the $(1, k)$-configurations of the $0 - 1$ knapsack polytope introduced by Padberg [67]. Boyd proves the following theorem:

Theorem 2 *Let $C \cup \{t\}$ be a $(1,k)$-configuration and let Q be a subset of C of cardinality k. For any $r \leq k$ the constraint:*

$$(k-r+1)x_t + \sum_{i \in Q} x_i \quad \leq \quad k \tag{2.14}$$

is a facet of $POK(C \cup \{t\})$ if and only if $P_t \cap P(Q) = \emptyset$ and $\cap_{\{S \subseteq Q : |S| = r-1\}} P(S) = \emptyset$.

Van de Leensel et al. [58] uses Boyd's definition of a minimal cover to investigate algorithmic consequences of lifting and lifting order sequences. We use $T(C)$ to denote $P(C) \setminus C$ and $R(C)$ to denote $V(G) \setminus P(C)$. In [58], a lifting order π is called a *predecessors first, remaining variables second* (PFRS) order for a subset of items $W \subseteq V(G)$ if π is a one-to-one mapping, $\pi : T(W) \cup R(W) \rightarrow \{1, ..., |T(W) \cup R(W)|\}$ satisfying:

- (i) $\pi(i) < \pi(j)$ if $i \in T(W), j \in R(W)$
- (ii) $\pi(i) < \pi(j)$ if $i, j \in T(W)$ and $j \in T(i)$
- (iii) $\pi(i) < \pi(j)$ if $i, j \in R(W)$ and $i \in T(j)$.

Given a minimal induced cover C and a PFRS order π, define predecessor and successor sets for a node j as:

$$p^\pi(j) \quad = \quad \{i \in P(C) \cup R(C) : \pi(i) < pi(j)\} \tag{2.15}$$

$$and$$

$$s^\pi(j) \quad = \quad \{i \in P(C) \cup R(C) : \pi(i) > \pi(j)\}. \tag{2.16}$$

For a lifted cover inequality following a PFRS order π at stage j where $j \leq |T(C)|$, the lifting process has already constructed the inequality:

$$\sum_{i \in C} x_i + \sum_{i \in T(C) \cap p^\pi(j)} \alpha_i (1 - x_i) \quad \leq \quad |C| - 1.$$

The lifting process then computes the value of α_j by solving:

$$\begin{aligned} \alpha_j \quad &= \quad |C| - 1 - \max\{\sum_{i \in c} x_i + \sum_{i \in T(C) \cap p^\pi(j)} \alpha_i (1 - x_i)\} \qquad (2.17) \\ s.t. \quad & \quad x_i = 1 \quad for\ i \in T(C) \cap s^\pi(j) \\ & \quad x_i = 0 \quad for\ i \in R(C) \\ & \quad x_j = 1 \\ & \quad x \in POK. \end{aligned}$$

At a stage $j > |T(C)|$, (ie. j corresponds to a an item in $R(C)$) the process has already constructed the lifted cover inequality:

$$\sum_{i \in C} x_i + \sum_{i \in T(C)} \alpha_i (1 - x_i) + \sum_{i \in R(C) \cap p^\pi(j)} \alpha_i x_i \quad \leq \quad |C| - 1.$$

The process then computes the value of α_j by solving:

$$\begin{aligned} \alpha_j \quad &= \quad |C| - 1 - \max\{\sum_{i \in C} x_i + \sum_{i \in T(C)} \alpha_i (1 - x_i) + \sum_{i \in R(C) \cap p^\pi(j)} \alpha_i x_i\} \qquad (2.18) \\ s.t. \quad & \quad x_i = 0 \quad for\ i \in R(C) \cap s^\pi(j) \\ & \quad x_j = 0 \\ & \quad x \in POK. \end{aligned}$$

This ordering ensures that in each step of the lifting process the precedence constraints are not violated and the variables set to 1 do not violated the knapsack inequality. Furthermore, Van de Leensel et al. [58] prove:

Theorem 3 *[58] Let C is a minimal induced cover and π be a PFRS order for C then if the coefficients α_j of x_j are lifted according to (2.17) for $j \in T(C)$ and according to (2.18) for $j \in R(C)$ then the inequality:*

$$\sum_{i \in C} x_i + \sum_{i \in T(C)} \alpha_i (1 - x_i) + \sum_{i \in R(C)} \alpha_i x_i \quad \leq \quad |C| - 1$$

defines a facet of the POK polytope.

It is stated that this lifting sequence can be used to lift $(1, k)$-configuration inequalities so that the final inequality obtained is facet inducing for the POK polytope.

Computing the values of the α_j's can be in general as difficult as solving a knapsack (or POK) problem. However, for the case $j \in P(C)$ Van de Leensel et al. provide a combinatorial interpretation that yields a polynomial time algorithm for computing the α_j's. This is accomplished by computing the number of components in the graph induced by C and subsets of $T(C)$. Let f be a function on subsets W of $T(C)$ that counts the number of components in the graph induced on node set $W \cup C$. Define a sequential lifting order π to be a *reverse topological order* if it satisfies $\pi(i) < \pi(j)$ when $j \in P_i$.

Given a reverse topological order π for the items in $T(C)$, define

$$\gamma_i \quad = \quad f(\{\pi^{-1}(1), ..., \pi^{-1}(i-1)\}) - f(\{\pi^{-1}(1), ..., \pi^{-1}(i)\}).$$

This γ function counts the difference in the number of components in the graph H induced by $C \cup \{\pi^-1(1), ..., \pi^-1(i-1)\}$ and the graph induced by adding node $\pi^{-1}(i)$ and relevant arcs to H. The resulting theorem is:

Theorem 4 ([58]) *Let $C \subseteq V(G)$ be a minimal induced cover and let π be a reverse topological order on $T(C)$. If the values of γ_i are computed according to π then for each $j = \pi^{-1}(1), ..., \pi^{-1}(|T(C)|)$:*

$$\sum_{i \in C} x_i + \sum_{i=\pi^{-1}(1)}^{\pi^{-1}(j)} \gamma_i(1 - x_i) \quad \leq \quad |C| - 1 \tag{2.19}$$

is a facet inducing inequality for the polytope Q where Q is the convex hull of valid POK solutions x where $x_i = 1$ for $i \in \{\pi^{-1}(j+1), ..., \pi^{-1}(|T(C)|\}$ and $x_i = 0$ for $i \in R(C)$.

There are initially at most $|C|$ graph components and every node that is added from $T(C)$ can only decrease the number of components. If a node $i \in T(C)$ has its corresponding x_i set to 0 this means that at least $\gamma_i + 1$ elements of C must be set to zero and the right hand side of the inequality can be decreased by γ_i. Computing the γ_i's can be done easily by counting the number of graph components in polynomial time.

In contrast, it is proved that computing the α_j's for $j \in R(C)$ to be a NP-hard problem by reducing from max clique. This reduction relies on a specific precedence graph G on which the lifting problem will be solved. The graph arising in the specific instances of POK problems that one is interested maybe be easier to handle. Van de Leensel et al. prove that this is the case when the precedence graph is a tree and the coefficients are polynomially bounded in the size of the tree. In this case computing the value of the α_j's for $j \in R(C)$ is a straight forward application of Johnson and Niemi's [54] algorithm for solving POK problems on trees via dynamic programming.

Much of the work of Van de Leensel is based on the work of Park and Park [68]. However, Park and Park use their alternative definition in which $C \subset V(G)$ is a *minimal induced cover* if:

$$\sum_{i \in P(C \setminus \{i\})} a_i \quad \leq \quad b$$

for all $i \in C$.

With regards to open pit mining. Boland et al. [36], [16] discovered a set of inequalities termed *clique based inequalities* that are valid for the POK polytope.

For two nodes i and j, if:

$$\sum_{l \in P_i \cup P_j} a_l \quad > \quad b$$

then i and j are said to *conflict*. A *conflict graph* is the graph $CG = (V(G), E)$ where $E = \{i, j : i \ and \ j \ conflict\}$. If $C \subset V(G)$ is a clique in the conflict graph then the inequality:

$$\sum_{j \in C} x_j \quad \leq \quad 1 \tag{2.20}$$

is valid for the POK polytope. Conditions on when (2.20) is facet defining are also presented.

In terms of finding an optimal solution to a POK problem not a great deal is known. The problem was shown to be hard to approximate within a factor of $2^{\log n^{\delta}}$ for some $\delta > 0$ under the assumption that 3-SAT $\notin DTIME(2^{n^{3/4+\epsilon}})$ by

Hajiaghayi et al. [47]. They actually show that this inapproximability result is true for the densest k-sub-hypergraph problem, a problem which POK generalizes. The POK problem has been related to a machine scheduling problem known as $1|prec|\sum w_jC_j$, minimizing average completion times of precedence constrained jobs on a single machine by Woeginger [84]. A constant factor approximation algorithm for the POK problem would yield a $2-\epsilon$ approximation algorithm for $1|prec|\sum w_jC_j$ an open problem that has been researched extensively [56] [73].

In terms of positive results, efficient approximation algorithms have been found for certain classes of precedence constraint graphs. As the problem contains the $0-1$ knapsack problem when no precedence constraints are given the problem is NP-hard on every class of graphs. As mentioned earlier, Niemi and Johnson [54] developed a FPTAS for the case where the precedence graph is a directed out-tree. This algorithm is based on dynamic-programming and the item weights being polynomial in the size of the problem.

Kolliopoulos and Steiner [56], give a PTAS algorithm for a class of 2-dimensional graphs generalizing series-parallel graphs. As with the Niemi and Johnson algorithm it is pseudo-polynomial time based on the weights.

One way to solve a POK problem is to formulate it as a directed cut problem with a knapsack constraint on the arcs in the cut. Let G be the directed graph representing the precedence constraints where an arc i to j implies that if node i is chosen then node j must be chosen as well. Let v_i be the value of node i. Let $a_i \geq 0$ be the weight associated with node i and $\sum_{i\in V(G)} a_ix_i \leq b$ be the knapsack constraint associated with the POK problem.

Construct a new graph G' from G by adding a new node s and arcs $si\ \forall i \in V(G)$, ie. $V(G') = V(G) \cup \{s\}$ and $A(G') = A(G) \cup \{si : i \in V(G') \setminus \{s\}\}$. Assign an arc weight of v_i to arcs si for all $i \in V(G') \setminus \{s\}$ and a weight of $-M$ to each arc ij that appeared in the precedence graph G where $M = \sum_{i:v_i>0} v_i$. The POK problem can now be stated as a maximum weight directed cut problem on the graph G' with the knapsack constraint on the arcs si: $\sum_{i:i\in V(G')\setminus\{s\}} a_i x_{si} \leq b$. We want node s to be inside the cut, therefore, we also add arcs from every node in $i \in V(G) \setminus \{s\}$ to s and give them a weight of $-M$.

Clearly, any maximum weight directed cut will not violate the precedence constraints as doing so would give a weight of at most $-M + \sum_{si:si\ in\ cut} v_{si} \leq 0$ and choosing the empty set achieves a cut at least as good. If node s is outside the cut and the nodes on one side of directed cut is not the empty set, then the weight of the directed cut is again at most $-M < 0$ which is worse than choosing the empty set.

We have now shown that the POK problem can be expressed as a maximum directed cut problem on the graph G'. In the following chapters we investigate the directed cuts in further detail. While our original motivation for looking at directed cuts came from the POK problem and mining applications, it is a combinatorial optimization problem of interest in its own right.

CHAPTER 3
The directed and undirected cut polyhedra

The cut polytope, $CUT_n^\square$, (resp., cut cone, CUT_n) is the convex hull (resp., positive hull) of the edge incidence vectors of the cuts in the complete graph, K_n. The cut cone and polytope arise in many fields [28, 29, 30], and the structure of facets of the cut polytope has been intensively studied. The book by Deza and Laurent [30] entitled "Geometry of Cuts and Metrics" is largely devoted to the study of the cut polyhedra and related metrics, it is nearly 600 pages and give a host of references to further work in the area. Cut polyhedra are too broad a topic for us to cover the majority of past work. We will instead focus on the results that we relate in some way to the directed cut polyhedra with explanations as to why these results are of interest to us.

Before reviewing existing work on the cut polyhedra, we will explain why the directed cut polytope is in one sense more complicated than the cut polytope. For the cut polytope an operation known as *switching* exists which is a face preserving automorphism. For a set of nodes $S \subset V(K_n)$, let $\delta(S) \in \{0,1\}^{|E(K_n)|}$ denote the incidence vector of the edges in the cut with S on one side and $V(K_n) \setminus S$ on the other. If $\delta(A_1), ..., \delta(A_{2^{n-1}})$ is an ordering of the set of all cuts of K_n and $\delta(S)$ is an arbitrary cut where $S \subseteq V(G)$, then the *switching operation* with respect to $\delta(S)$ maps the set of cuts $\delta(A_1), ..., \delta(A_{2^{n-1}})$ onto itself.

The switching operation is in fact the symmetric difference operation Δ. For a cut $\delta(A_i)$, $\delta(A_i \Delta S) = \delta(A_j)$ for some $1 \leq j \leq 2^{n-1}$. Applying the operation to the family of cuts $(\delta(A_1), ..., \delta(A_{2^{n-1}}))$ yields $(\delta(A_1'), ..., \delta(A_{2^{n-1}}'))$ where $\delta(A_i') = \delta(A_i \Delta S)$. As stated, it can be shown that $\{\delta(A_1), ..., \delta(A_{2^{n-1}})\} = \{\delta(A_1'), ..., \delta(A_{2^{n-1}}')\}$, ie. the switching operation is an automorphism.

In terms of the cut polytope, the operation is a face preserving automorphism. This implies that any vector, $\delta(A_i)$, which is a vertex of the cut polytope, can be mapped to the origin by setting $S = A_i$. By using the switching operation, if one knows the facial structure at the origin then one can obtain the facial structure at any vertex $\delta(A_i)$ of the cut polytope. This allows one to study the facial structure of the cut cone and extend the results to the structure of the cut polytope.

In contrast, the directed cut polytope does not have such a set of automorphisms. In Section 4.5 we show that even for $\vec{K}_4$, the complete directed graph with four nodes, two vertices of the directed cut polytope can have a different number of incident faces. This implies that studying the directed cut cone can not provide a complete understanding of the facial structure of the directed cut polytope. In this sense, the directed cut polytope is more complex than the cut polytope. An operation relating switching to the directed cut polyhedra will be presented in Section 4.6, however it will not be as useful as switching for the cut polytope.

Many of our results will still focus on the study of the directed cut cone. Early research on the cut polyhedra focused on the cone as well, before an understanding of the switching operation existed. These are complex objects and an understanding of the directed cut cone is still fundamental to understanding the polytope.

The current chapter and the chapters that follow on the directed cut polyhedra are the main contributions of this thesis. We envision that the theory developed here could to be useful to many fields beyond the focus of our work. In the same way that research on the cut polyhedra has touched many different fields of research from Ising spin glass models (see for example: [43], [13]), network design [60] and more recently quantum computing (see, [7]), just to name a few.

3.1 Semidefinite programming and cuts

Optimization over the cut polytope is known as the maximum cut problem, and is NP-hard. A LP-relaxation for this problem is provided by the metric polytope, and performance bounds are available, eg., see [8]. A celebrated result of Goemans and Williamson [41] uses semidefinite programming to provide tighter performance bounds. The max cut problem can be modelled as the quadratic program:

$$\begin{aligned} \max \quad & \frac{1}{2}\sum_{i<j} c_{i,j}(1 - y_i y_j) \\ s.t. \quad & y_i \in \{-1, 1\} \qquad \forall\ i \in V(G). \end{aligned} \tag{3.1}$$

Where nodes i with equal values of y_i lie on the same side of the optimal cut. Relaxing the constraint that the y_i's are $1-$dimensional, one can instead require y_i to be an $n-$dimensional vector on the unit sphere and replace the objective function multiplication of y_i's with a dot product. The resulting relaxation has the form:

$$\begin{aligned} \max \quad & \frac{1}{2}\sum_{i<j} c_{i,j}(1 - y_i \cdot y_j) \\ s.t. \quad & y_i \in S_n \qquad \forall\ i \in V(G) \end{aligned} \tag{3.2}$$

where S_n is the unit sphere of dimension n. This relaxation is a semidefinite program that can be solved in polynomial time by the interior point [65] or ellipsoid methods [45]. Goemans and Williamson then use the solution of the semidefinite program and round the solution vectors to values of either 1 or -1 based on which side of a randomly chosen hyperplane through the origin they appear. The valid cut they obtain is within an expected value of at least 0.87856 of optimal. This analysis is based on the weights $c_{i,j}$ on the edges being non-negative.

In general any $\{-1,1\}$ quadratic program of the form (3.1) can be expressed as the binary quadratic program:

$$\begin{aligned} \max \quad & \sum_{i \neq j} c_{i,j} x_i (1 - x_j) \\ s.t. \quad & x_i \in \{0,1\} \qquad \forall\, i \in V(G) \end{aligned}$$

The two forms of quadratic programs are linearly transformations of each other. The product xx^T can be modelled by a symmetric matrix Y. Let y be the diagonal of matrix Y, the set of feasible matrices must satisfy the property that $Y - yy^T$ is positive semidefinite.

The techniques used in the semidefinite programming rounding algorithm assume that the weights on the arcs are non-negative. Alon and Naor [2] developed an alternate method of rounding a semidefinite program and analysis of the method that provides a way to allow negative terms to be considered on the edges. These methods do not trivially extend to allow knapsack type inequalities on the edges to be included. We discuss some existing work on formulating a knapsack type constraint in a semidefinite program briefly here, for a more thorough presentation see [51].

Given a quadratic program with a knapsack constraint $a^T x \leq b$ the most straight forward method of modelling it is by restricting the values of the diagonal of Y, ie.:

$$\begin{aligned} \max \quad & trace(CY) \\ s.t. \quad & trace(Diag(a)Y) \leq b \\ & Y - yy^T \succeq 0 \end{aligned} \tag{3.3}$$

where y is the diagonal of symmetric matrix Y and C is the constraint matrix $c_{i,j}$. In [51], this and other methods of modelling knapsack constraints are compared. Another method they investigate is squaring both sides of the knapsack constraint to obtain the inequality: $a^T x x^T a \leq b^2$. Modelling this as a semidefinite program gives:

$$\begin{aligned} \max \quad & trace(CY) \\ s.t. \quad & trace(aa^T Y) \leq b^2 \\ & Y - yy^T \succeq 0. \end{aligned}$$

They show that this modelling is stronger than (3.3). A further strengthening is obtained by multiplying inequality $a^T y \leq b$ by either y_i or $(1 - y_i)$. Using all possible n inequalities obtained by multiplying the knapsack inequality by y_i and using the inequality obtained by multiplying the knapsack inequality by $(1 - y_1)$ and summing it with the knapsack inequality multiplied by y_1 the following semidefinite program is obtained:

$$\begin{array}{ll} \max & trace(CY) \\ s.t. & \sum_{j=1}^{n} a_j y_{i,j} \leq b y_i \qquad i = 1, ..., n \\ & \sum_{j=1}^{n} a_j (y_{j,j} - y_{1,j}) \leq b(1 - y_1) \\ & Y - yy^T \succeq 0. \end{array}$$

It can be shown that this is a further strengthening. Further strengthening inequalities will be discussed in Section 3.2 and how binary quadratic programming relates to optimization over the cut polyhedra.

Goemans and Feige [35] extended the techniques of Goemans and Williamson [41] to obtain a 0.859 approximation to the maximum directed cut problem. They formulated the maximum directed cut problem as a binary quadratic program and considered the semidefinite relaxation:

$$\max \quad \sum_{ij \in A(G)} w_{ij} (t_i \cdot f_j) \tag{3.4}$$

$$s.t. \quad t_i \cdot f_i = 0 \qquad \forall i \in V(G) \tag{3.5}$$

$$t_i + f_i = v_0 \qquad \forall i \in V(G) \tag{3.6}$$

$$t_i, f_i \in \mathbb{R}^k \qquad \forall i \in V(G) \tag{3.7}$$

The binary quadratic program uses variable t_i to represent a truth assignment of node i being in the set S. If $f_i = 1$ then node i is not in S. If the arc ij is in the cut then $t_i = 1$ and $f_j = 1$. The constraint $t_i f_i = 0$ ensures that at most one of t_i or f_i can be set to 1.

The semidefinite relaxation takes t_i and f_i to be vectors of dimension k. The objective function (3.4) is expressed as the dot product of t_i and f_j, for arc ij being in the cut, multiplied by the arc weight w_{ij}. The vector v_0 is any unit vector. They strengthen the relaxation further by adding the triangle inequalities to the relaxed formulation. Triangle inequalities for cut and directed cut problems will be discussed in further detail in Section 3.2.

While the semidefinite relaxations have proved to be effective and tighter relaxations of the cut and knapsack polyhedra it is not immediately clear how one could use the knapsack semidefinite relaxations in conjunction with the techniques used in the semidefinite programming rounding algorithm of [41] to produce cuts that adhere to the knapsack constraint.

To this end our focus has been on integer program formulations and linear program relaxations of the POK problem. One way to model a POK problem is as a graph closure with a knapsack constraint or in the case of mine scheduling, multiple knapsack constraints. A problem that generalizes the POK problem is that of finding a directed cut with a knapsack constraint on the arcs that cross the cut. This problem can be used to model many of the optimization problems arising in mine planning. While much is known about the structure of the knapsack polytope, relatively little is known about the structure of the POK polytope and the directed cut polytope.

Directed cuts are of interest in their own right. One of the classical algorithms taught in most undergraduate computer science programs is on how to find a minimum weight directed cut with non-negative edge weights. While the undirected cut polytope has been extensively studied, little work has previously been done of the directed cut polytope. The focus of this chapter and Chapters 4 through 7 will be on furthering our understanding of the directed cut polyhedra.

3.2 The cut polytope and cone

A *distance function* d on a set S, $d : S \times S \to \mathbb{R}$, is a function where d is symmetric, $d(x,y) = d(y,x)\ \forall x,y \in S$, and $d(x,x) = 0\ \forall x \in S$. When d is a distance function on S then (S,d) is known as a distance space.

Since we often have to refer to both arcs and edges we will use a pair of indices without a comma to define a directed arc, ie. ij refers to the arc from node i to node j. Similarly, we will use a pair of subscripts separated by a comma to refer to an edge, ie. i,j refers to an edge between nodes i and j and $i,j = j,i$.

If the following are satisfied,

$$d_{i,k} \leq d_{i,j} + d_{j,k}, \tag{3.8}$$

$$d_{i,j} = 0 \to i = j, \tag{3.9}$$

then (S,d) is a *metric space*. Removing restriction (3.9) defines what is known as a *semimetric space*. One can easily check that the function $\delta : S \to \{0,1\}^{E(K_n)}, S \subseteq$

$V(K_n)$ mapping a subset S to the incidence vector of edges leaving the set S defines a semimetric.

The simplest facets of the cut polytope are those defined by the triangle inequalities:

$$\begin{aligned} x_{i,j} - x_{i,k} - x_{j,k} &\leq 0, \\ -x_{i,j} + x_{i,k} - x_{j,k} &\leq 0, \\ -x_{i,j} - x_{i,k} + x_{j,k} &\leq 0, \end{aligned} \tag{3.10}$$

and the perimeter triangle inequalities:

$$x_{i,j} + x_{j,k} + x_{i,k} \leq 2,, \tag{3.11}$$

for distinct $i, j, k \in V(K_n)$. The cone defined by inequalities (3.10) is known as the semimetric cone MET_n. The polytope defined by inequalities (3.10) and (3.11) is known as the semimetric polytope $\mathrm{MET}_n^\square$. It is well known that the semimetric cone and polytope are relaxations of the cut cone and polytope respectively,

$$CUT_n \subseteq MET_n \quad and \quad CUT_n^\square \subseteq MET_n^\square.$$

It is also well known that triangle inequalities (3.10) and the perimeter triangle inequalities (3.11) are facet defining inequalities for the cut polytope. This can be proved by showing that the triangle inequalities (3.10) are facets of for $\mathrm{CUT}_3^\square$ and then applying the operations known as switching and zero lifting. Switching was discussed earlier and will be discussed in further detail below. Zero lifting will be discussed later as well.

The triangle inequalities involving a specific node, say node $1 \in V(K_n)$, define a relaxation of the semimetric polyhedra. The rooted semimetric cone, RMET_n is defined by the inequalities:

$$\begin{aligned} x_{1,j} - x_{1,k} - x_{j,k} &\leq 0, \\ -x_{1,j} + x_{1,k} - x_{j,k} &\leq 0, \\ -x_{1,j} - x_{1,k} + x_{j,k} &\leq 0, \end{aligned} \tag{3.12}$$

for $2 \leq j < k \leq n$. The rooted semimetric polytope, $\mathrm{RMET}_n^\square$ is defined by the inequalities (3.12) along with the perimeter inequalities involving node 1:

$$x_{1,j} + x_{j,k} + x_{1,k} \leq 2,, \tag{3.13}$$

for $2 \leq j < k \leq n$.

Proposition 27.2.1 of [30] states:

Proposition 5 ([30]) *The only integral vectors of $RMET_n^\square$ are the cut vectors $\delta(S)$ for $S \subseteq V(K_n)$. Moreover, every cut vector is a vertex of $RMET_n^\square$.*

This implies that the cut, semimetric and rooted semimetric polyhedra satisfy the relation:

$$CUT_n \subseteq MET_n \subseteq RMET_n, \text{ and } CUT_n^\square \subseteq MET_n^\square \subseteq RMET_n^\square.$$

Another family of polyhedra related to the cut polyhedra are known as the correlation polytope and cone. Let $V = \{1, ..., n\}$ be a set of n elements, for a subset $S \subseteq V$ define $\pi(S) \in \mathbb{R}^{\binom{n}{2}+n}$ where $1 \leq i \leq j \leq n$ by:

$$\pi(S)_{i,j} = \begin{cases} 1 & if\ i,j \in S \\ 0 & otherwise. \end{cases}$$

The positive hull generated by the vectors $\pi(S)$ for all $S \subseteq V$ is the *correlation cone*, COR_n. Likewise, the convex hull of all vectors $\pi(S)$ is the *correlation polytope*, $\mathrm{COR}_n^\square$.

The correlation polytope $COR_n^\square$ is a linear transformation of the cut polytope $CUT_{n+1}^\square$. A mapping $\xi : \mathbb{R}^{E(K_{n+1})} \rightarrow \mathbb{R}^{\binom{n}{2}+n}$ can be defined that maps a vector $p \in COR_n^\square$ onto a vector $x \in CUT_{n+1}^\square$, ie. $p = \xi(x)$. Where:

$$p_{i,i} = x_{i,n+1} \qquad i = 1, ..., n \tag{3.14}$$

$$p_{i,j} = \frac{1}{2}(x_{i,n+1} + x_{j,n+1} - x_{i,j}) \qquad \forall 1 \leq i < j \leq n. \tag{3.15}$$

Since we can write out the inverse ξ^{-1} where:

$$x_{i,n+1} = p_{i,i} \qquad i = 1, ..., n \tag{3.16}$$

$$x_{i,j} = p_{i,i} + p_{j,j} - 2p_{i,j} \qquad \forall 1 \leq i < j \leq n \tag{3.17}$$

ξ is a bijection.

As vertex $n+1$ has a special role in this mapping, in the literature the mapping ξ is referred to as *pointing* at vertex $n+1$. The choice of $n+1$ was arbitrary and mappings could be defined pointing at any vertex of K_{n+1}. Note that this mapping is valid for transforming a vector in the cone CUT_{n+1} to a vector p in the cone COR_n. The relation between the cut and correlation polyhedra was discovered independently by many authors. The book by Deza and Laurent [30] lists Hammer [48], Deza [26], Barahona, Jünger and Reinelt [14], and De Simeone [23].

The bijection allows us to easily obtain families of facet defining inequalities for the correlation polytope from well known families of facet defining inequalities for the cut polytope and vice versa. Using the bijection, the triangle inequalities (3.10) and (3.11) become:

$$p_{i,j} \geq 0 \tag{3.18}$$

$$p_{i,j} \leq p_{i,i} \tag{3.19}$$

$$p_{i,j} \leq p_{j,j} \tag{3.20}$$

$$p_{i,i} + p_{j,j} \leq 1 + p_{i,j} \tag{3.21}$$

$$p_{i,k} + p_{j,k} \leq p_{k,k} + p_{i,j} \tag{3.22}$$

$$p_{i,i} + p_{j,j} + p_{k,k} \leq 1 + p_{i,j} + p_{i,k} + p_{j,k} \tag{3.23}$$

for the correlation cone and polytope.

Proposition 5.2.7 of [30] expresses a mapping between facets of the cut and correlation polyhedra:

Proposition 6 (Proposition 5.2.7 of [30]) *Let $a \in \mathbb{R}^{V(K_n)}$, $b \in \mathbb{R}^{E(K_n)}$, $c \in \mathbb{R}^{E(K_{n+1})}$ be linked by:*

$$\begin{aligned} c_{i,n+1} &= a_i + \frac{1}{2} \sum_{1 \leq j \leq n, j \neq i} b_{i,j} \quad i = 1, ..., n \\ c_{i,j} &= -\frac{1}{2} b_{i,j} \quad 1 \leq i < j \leq n \end{aligned}$$

For a given, $\alpha \in \mathbb{R}$, the inequality:

$$\sum_{1 \le i < j \le n} c_{i,j} x_{i,j} \le \alpha$$

is valid (facet inducing) for $CUT^{\square}_{n+1}$ if and only if the inequality:

$$\sum_{1 \le i \le n} a_i p_{i,i} + \sum_{1 \le i < j \le n} b_{i,j} p_{i,j} \le \alpha$$

is valid (face inducing) for $COR^{\square}_{n}$.

The binary quadratic programming problem can be viewed as an optimization problem on the correlation polytope. As:

$$\begin{array}{rll} \max & \displaystyle\sum_{1 \le i \le j \le n} c_{i,j} x_i x_j & \\ s.t. & x_i \in \{0,1\} & i = 1, ..., n \end{array}$$

is equivalent to:

$$\begin{array}{rll} \max & c^T \pi(S) & \\ s.t. & \pi(S) \in COR^{\square}_{n}, & S \subseteq V. \end{array}$$

This means that facets of the correlation polytope can be used to strengthen relaxations of a binary quadratic program. In particular, the semidefinite relaxations mentioned earlier can be strengthened with the triangle inequalities (3.18) - (3.23).

3.3 Operations on the cut polyhedra

When studying the polyheral structure of an object like the cut polytope, one would like a way of proving that inequalities are valid or facets. In this section we will review a few operations that can be performed on the cut polytope that either preserve valid inequalities or prove that inequalities are facet defining. Of particular interest to us are the operations: permuting, collapsing, switching and zero lifting. Other operations exist but they will either be left out or introduced later.

As stated in Deza and Laurent's book [30], since the cut polytope and cone deal with the complete graph K_n the faces are clearly preserved under any permutation of the nodes. Given a permutation σ of the nodes $\{1, ..., n\}$ and $v \in \mathbb{R}^{E(K_n)}$ define $\sigma(v)$ to be $\sigma(v)_{i,j} = v_{\sigma(i),\sigma(j)}$ for $i, j \in E(K_n)$. The following result trivially holds:

Lemma 7 (Lemma 26.2.1 of [30]) *Given* $v \in \mathbb{R}^{|E(K_n)|}$, $v_0 \in \mathbb{R}$ *and* σ *a permutation of* $\{1, ..., n\}$*, the following statements are equivalent:*

- *The inequality* $v^T x \leq v_0$ *is valid (resp. facet inducing for* $CUT_n^\square$*.*
- *The inequality* $\sigma(v)^T x \leq v_0$ *is valid (resp. facet inducing for* $CUT_n^\square$*.*

The *collapsing operation*, as it is referred to in the literature ([25], [27]), maps a vector $v \in \mathbb{R}^{E(K_n)}$ to a vector $v' \in \mathbb{R}^{E(K_m)}$ where $m < n$. It provides a way to construct a valid inequality for $\mathrm{CUT}_m^\square$ from a valid inequality for $\mathrm{CUT}_n^\square$ by identifying vertices and adding the weights of their incident edges. Formally, the collapsing operation of a vector $v \in \mathbb{R}^{E(K_n)}$ for a partition $\pi = \{M_1, ..., M_m\}$ of the vertices $\{1, ..., n\}$ is:

$$v_{i,j}^{\pi} = \sum_{s \in M_i, t \in M_j} v_{s,t}. \tag{3.24}$$

This notion generalizes to an arbitrary subgraph $G \subseteq K_n$, if we simply assume that $v_{i,j} = 0$ for edges not appearing in $E(G)$.

Two other useful operations for proving inequalities are facets of the cut polytope are switching and zero lifting. Conversely to collapsing, *lifting*, as described in Section

2.3.1, takes a valid or facet inducing inequality for a lower dimensional polyhedra, like $CUT_n^\square$, and constructs a valid or facet inducing inequality for a higher dimensional polyhedra, like $CUT_{n+1}^\square$. A useful type of lifting, known as *zero-lifting*, takes an inequality $v^T x \leq v_0$ that is valid for $\mathrm{CUT}_n^\square$ and constructs the vector v' where $v'_{i,j} = v_{i,j}$ for $i, j \in E(K_n)$ and $v'_{i,j} = 0$ for $j = n+1$.

For the cut polytope $\mathrm{CUT}_n^\square$ Theorem 26.5.1 of [30] states:

Theorem 8 (Theorem 26.5.1 of [30]) *Given, $v_0 \in \mathbb{R}, v \in \mathbb{R}^{|E(K_n)|}$ and zero-lifted $v' \in \mathbb{R}^{|E(K_{n+1})|}$ the following are equivalent.*

- *$v^T x \leq v_0$ is facet inducing for $CUT_n^\square$.*
- *$v'^T x \leq v_0$ is facet inducing for $CUT_{n+1}^\square$.*

It is straight forward to check that the triangle inequality:

$$x_{1,3} \leq x_{1,2} + x_{2,3} \tag{3.25}$$

is a facet of $\mathrm{CUT}_3^\square$. Applying Theorem 8 and the fact that we can relabel the nodes of the complete graph if needed (by Lemma 7) proves that the triangle inequalities are facet defining for $\mathrm{CUT}_n^\square$.

To prove Theorem 8 in Deza and Laurent's book, they prove a useful result which is stated as Lemma 26.5.2 in [30]. We state it below.

For an inequality $v^T x \leq 0$, let $R(v)$ define its set of roots, ie. $R(v) = \{x : x \in CUT_n, v^T x = 0\}$. For $F \subseteq E(K_n)$ let $\bar{F} = E(K_n) \setminus F$ and for $x \in \mathbb{R}^{|E(K_n)|}$, let $x_F = (x_e)_{e \in F}$ be the projection of x onto the edges of F. If X is a subset of $\mathbb{R}^{E(K_n)}$ then let $X_F = \{x_F | x \in X\}$ and $X^F = \{x | x \in X, x_F = 0\}$.

Lemma 9 (26.5.2 of [30]) *Let $v^T x \leq 0$ be a valid inequality for CUT_n and let $R(v)$ denote its set of roots. Let F be a subset of $E(K_n)$.*

(i) If $rank(R(v)_F) = |F|$ and $rank(R(v)^F) = |\bar{F}| - 1$, then the inequality $v^T x \leq 0$ is facet inducing.

(ii) If the inequality $a^T x \leq 0$ is facet inducing and $v_{\bar{F}} \neq 0$, then $rank(R(v)_F) = |F|$.

We will use variations of this lemma later on when proving some result on the directed cut polyhedra.

A fourth useful operation mentioned at the start of the chapter is known as switching. Given every facet of the cut cone, CUT_n, every facet of the cut polytope, $\text{CUT}_n^\square$, can be defined as follows.

Let S be a subset of V, then the *S-switching of an inequality* $a^T x \leq a_0$ is an inequality $(a')^T x \leq a_0 - a^T \delta_G(S)$ where $a' \in \mathbb{R}^{|E(G)|}$ is defined by $a'_{i,j} = (-1)^{\delta_{i,j}(S)} a_{i,j}$. Such an inequality is said to be switching equivalent to $a^T x \leq a_0$x.

The *switching mapping* is an affine bijection that maps:

$$r_B(x)_e = \begin{cases} 1 - x_e & if e \in B \\ x_e & otherwise. \end{cases} \tag{3.26}$$

Proposition 26.3.6 of [30] states:

Proposition 10 (Proposition 26.3.6 of [30]) *Let $\mathcal{A}$ be a collection of subsets of E that is closed under the symmetric difference. Suppose that*

$$C(A) = \{x \in \mathbb{R}^{|E|} : v_i^T x \leq 0 \ for \ i = 1, ..., m\}.$$

Then,

$$P(\mathcal{A} = \{x \in \mathbb{R}^{|E|} : (v_i^B)^T x \leq -v_i(B) \ for \ i = 1, ..., m \ and \ B \in \mathcal{A}\}.$$

For cuts the corollary of this proposition is:

Proposition 11 *Suppose that*

$$CUT_n = \{x \in \mathbb{R}^{|E(K_n)|} : v_i^T x \leq 0 \ for \ i = 1, ..., m\}.$$

Then,

$$CUT_n^\square = \{x \in \mathbb{R}^{|E(K_n)|} | (v_i^{\delta(A)})^T x \leq -v_i^T \delta(A) \ for \ i = 1, ..., m \ and \ A \subseteq V(K_n)\}.$$

This proposition provides the relation between the cut cone and polytope and was the key to work on characterizing the projection of the cut polytope by Barahona and Mahjoub [12]. We will present this work in Chapter 6.

In the following section, we develop some theory related to directed cut polyhedra and their relaxations. This theory will include definitions of relaxations of the directed cut polyhedra and operations that prove inequalities are valid and/or facet defining.

3.4 Directed cut polyhedra

We would like to define a distance space that relates directed cuts to a metric in much the same way as the undirected case. Let $\vec{K}_n$ denote the complete directed graph with n nodes. Let $\delta^+ : S \to \mathbb{R}^{A(\vec{K}_n)}, S \subseteq V_n$ be the directed cut vector for a set S, where $\delta^+(S)_{ij} = 1$ if $i \in S$ and $j \notin S$.

The distance space definition is too restrictive for directed cuts as symmetry does not hold: $\delta^+(S)_{ij}$ is not equal to $\delta^+(S)_{ji}$ if either is equal to 1. The directed cuts have a more general three point symmetry,

$$\delta^+(S)_{ij} + \delta^+(S)_{jk} + \delta^+(S)_{ki} \quad = \quad \delta^+(S)_{ji} + \delta^+(S)_{kj} + \delta^+(S)_{ik}. \tag{3.27}$$

This is evident as both sides of equation (3.27) are equal to 0 if $i, j, k \in S$ or $i, j, k \notin S$. If S contains only one of i, j, k say i and $j, k \notin S$ then $\delta^+(S)_{ij} = 1$ and the other two terms on the right hand side of (3.27) are equal to 0 while the left hand side has $\delta^+(S)_{ik} = 1$ and the other two terms are equal to zero. Finally, if S contains two nodes, say i, k and $j \notin S$ then $\delta^+(S)_{ij} = 1$ and $\delta^+(S)_{kj} = 1$ and all other terms are 0, so (3.27) is sissified. By relabeling nodes if needed it is easy to see that (3.27) holds for all possible sets S.

We define a *directed distance* as a function that satisfies such a three point symmetry as well as $d_{xx} = 0 \; \forall x \in X$. If the directed distance satisfies the following triangle inequality:

$$d_{ik} \leq d_{ij} + d_{jk}, \tag{3.28}$$

then we will call (S, d) a *directed-semimetric* space. One can easily check that for $S \subseteq V(\vec{K}_n)$, (S, δ^+) is a directed semimetric. We can define the directed-semimetric cone DMET_n as the cone defined by the symmetric equalities (3.27), non-negativity constraints $x_{ij} \geq 0 \; \forall 1 \leq i \neq j \leq n$ and following triangle inequalities:

$$x_{ik} - x_{ij} - x_{jk} \quad \leq \quad 0 \qquad 1 \leq i, j, k \leq n \textit{ all distinct} \tag{3.29}$$

and the directed-semimetric polytope $\mathrm{DMET}_n^{\square}$ as the polytope defined by the semimetric cone with the additional perimeter triangle inequalities:

$$x_{ij} + x_{jk} + x_{ki} \;\leq\; 1 \qquad 1 \leq i, j, k \leq n \textit{ all distinct.} \tag{3.30}$$

We will show in Corollary 14 below that all $0, 1$ solutions of the polytope $\mathrm{DMET}_n^{\square}$ are directed cuts vectors and all directed cut vectors are included in $\mathrm{DMET}_n^{\square}$. This implies that $\mathrm{DMET}_n^{\square}$ is a relaxation of the directed cut polytope.

This is not the first time that a relaxation of the directed cut polytope based on the directed triangle inequalities has been considered. In [20], Charikar et al. look at what they define as a *directed semimetric space* as the non-negativity constraints, along with the triangle inequalities of the form (3.29). Their goal was not to approximate the cut polytope but instead to investigate extending results on metric embedding related to partitioning problems in undirected graphs to problems involving directed graphs.

If we consider the triangle inequalities and cycle equalities that only involve a given node $1 \in V(\vec{K}_n)$ we can define rooted relaxations of the directed-semimetric cone and polytope. We define the rooted directed semimetric cone, RDMET_n as the cone defined by, for each $ij \in A(\vec{K}_n)$:

$$\begin{aligned}
x_{1i} + x_{ij} + x_{j1} &= x_{1j} + x_{ji} + x_{i1} && (3.31)\\
x_{1i} - x_{1j} - x_{ji} &\leq 0 && (3.32)\\
x_{i1} - x_{ij} - x_{j1} &\leq 0 && (3.33)\\
x_{ij} - x_{i1} - x_{1j} &\leq 0 && (3.34)\\
x_{ij} &\geq 0. && (3.35)
\end{aligned}$$

The rooted directed semimetric polytope, $\mathrm{RDMET}_n^{\square}$, is the polytope defined by the inequalities above and the perimeter inequalities:

$$x_{1i} + x_{ij} + x_{j1} \leq 1 \qquad (3.36)$$

where we take all $ij \in A(\vec{K}_n)$. By definition, the rooted directed semimetric cone and polytope are relaxations of the directed semimetric cone and polytope respectively. Lemma 13 establishes that the rooted semimetric polytope is a relaxation of the directed cut polytope, ie. the only $0-1$ vectors in $\mathrm{RDMET}_n^{\square}$ are the directed cut vectors and every directed cut vector is in $\mathrm{RMET}_n^{\square}$.

In Chapter 4 we prove that inequalities (3.29) are facets of the directed cut cone and polytope and inequalities (3.30) are facets of the directed cut polytope. Lemma 13 and Corollary 14 below imply the following relation:

$$DCUT_n \subseteq DMET_n \subseteq RDMET_n \textit{ and } DCUT_n^{\square} \subseteq DMET_n^{\square} \subseteq RDMET_n^{\square}. \qquad (3.37)$$

The defined polyhedra for the complete directed graph $\vec{K}_n$ are not full dimensional, which is evident from the linearities (3.27). It can be useful to consider the projection onto a graph with fewer arcs where the defined polyhedra are full dimensional.

Lemma 12 *The directed cut polytope $DCUT_n^\square$, the directed semimetric polytope $DMET_n^\square$ and the rooted directed semimetric polytope $RDMET_n^\square$ have dimension $\binom{n}{2} + n - 1$.*

Proof. By the relationship of (3.37) it suffices to show an upper bound on the dimension of $\mathrm{RDMET}_n^\square$ and a lower bound on the dimension of $\mathrm{DCUT}_n^\square$ which are both given by the formula in the statement of the lemma. Consider the polytope defined by the inequalities (3.32), (3.33), (3.34), (3.36), the non-negativity constraints $x_{ij} \geq 0$ and the three point symmetries $x_{1j}+x_{ji}+x_{i1} = x_{1i}+x_{ij}+x_{j1}$. These equations define $\mathrm{DMET}_n^\square$. By performing Gaussian-Elimination on this system of equations, one can replace each occurrence of x_{ji} with $x_{1i}+x_{ij}+x_{j1}-x_{1j}-x_{i1}$ where $j > i \geq 2$. This eliminates $\binom{n}{2} - n + 1$ variables leaving $\binom{n}{2} + n - 1$ variables and proves the upper bound.

For the lower bound, a set of $\binom{n}{2}+n-1$ linearly independent directed cut vectors are constructed. Let $S_{i,j} = \{k : k \leq i \text{ or } j < k \leq n\}$ for $(1 \leq i < j \leq n)$ and let $T_i = \{k : 2 \leq k \leq i\}$ for $i = 2 \cdots n$. We claim that the set of directed cut vectors $C = \{\delta^+(S_{i,j}) : 1 \leq i < j \leq n\} \cup \{\delta^+(T_i) : 2 \leq i \leq n\}$ are linearly independent. By construction, the matrix M formed by using these directed cut vectors as the rows where the columns are indexed by ij for $1 \leq i < j \leq n$ followed by the columns $i = 1$, $j = 2, \cdots, n$ is lower triangular with all 1's on the diagonal. ■

It will be useful to consider the directed cut polytope and cone on the graph $\vec{J}_n$ with n nodes and arc set $A(\vec{J}_n) = \{ij : 1 \leq i < j \leq n\} \cup \{i1 : 2 \leq i \leq n\}$

as opposed to the complete directed graph $\vec{K}_n$. The polyhedra DCUT_n, $\mathrm{DCUT}_n^{\square}$, DMET_n, $\mathrm{DMET}_n^{\square}$, RDMET_n and $\mathrm{RDMET}_n^{\square}$ become full-dimensional when restricted to the arc set $A(\vec{J}_n)$. From here on we will use this representation.

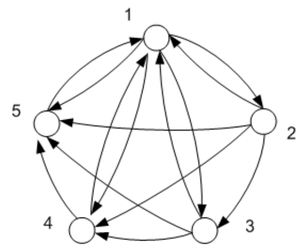


Figure 3–1: The directed graph $\vec{J}_5$.

The set of inequalities that define DMET_n, $\mathrm{DMET}_n^{\square}$, RDMET_n and $\mathrm{RDMET}_n^{\square}$ need to be modified slightly as some of the arcs used to index the inequalities do not exist in $A(\vec{J}_n)$. As mentioned in the proof of Lemma 12 we can substitute each occurrence of x_{ji}, for $i < j$, in inequalities $x_{ji} \geq 0$ and (3.29) with $x_{1i} + x_{ij} + x_{ji} - x_{i1} - x_{j1}$ for $2 \leq i < j \leq n$. This yields the following set of equations that define the cone DMET_n:

For $2 \leq i \leq n$,

$$x_{i1} \geq 0, \quad x_{1i} \geq 0 \tag{3.38}$$

For $2 \leq i < j \leq n$,

$$x_{ij} \geq 0 \tag{3.39}$$

$$-x_{ij} - x_{i1} + x_{1i} - x_{1j} + x_{j1} \leq 0 \tag{3.40}$$

$$-x_{ij} + x_{i1} - x_{j1} \leq 0 \tag{3.41}$$

$$-x_{ij} - x_{1i} + x_{1j} \leq 0 \tag{3.42}$$

$$x_{ij} - x_{i1} - x_{1j} \leq 0 \tag{3.43}$$

For $2 \leq i < j < k \leq n$,

$$-x_{ij} + x_{ik} - x_{jk} \leq 0 \tag{3.44}$$

$$x_{ij} - x_{ik} - x_{jk} + x_{1k} - x_{k1} + x_{j1} - x_{1j} \leq 0 \tag{3.45}$$

$$-x_{ij} - x_{ik} + x_{jk} + x_{i1} - x_{1i} + x_{1j} - x_{j1} \leq 0. \tag{3.46}$$

The inequalities (3.38)-(3.40) above are the non-negativity constraints, and the remaining inequalities are the triangle inequalities. All of the above inequalities along with the following *perimeter inequalities* define $\mathrm{DMET}_n^{\square}$.

For $2 \leq i < j \leq n$,

$$x_{1i} + x_{ij} + x_{j1} \leq 1 \tag{3.47}$$

For $2 \leq i < j < k \leq n$,

$$x_{ij} + x_{jk} + x_{ik} + x_{1i} - x_{i1} + x_{k1} - x_{1k} \leq 1. \tag{3.48}$$

While we have stated that $\mathrm{RDMET}_n^\square$ and $\mathrm{DMET}^\square$ are relaxations of the directed cut polytope, the following lemma provides further validation of their relation to directed cuts.

Lemma 13 *The only integral vectors of $RDMET_n^\square$ are the directed cut vectors $\delta^+(S)$ for $S \subseteq V_n$ and every directed cut vector is a vertex of $RDMET_n^\square$.*

Proof. In the first part of the proof we will use the full dimensional definition of $\mathrm{RDMET}_n^\square$ given in terms of all $n(n-1)$ variables. The non-negativity and the perimeter inequalities imply that the only integral vectors in $\mathrm{RDMET}_n^\square$ are 0/1 valued. Let $x \in RDMET_n^\square \cap \{0,1\}^{A_n}$. Let $I = \{i : x_{i1} = 1\}$ and $J = \{i : x_{1i} = 1\}$. We first show that one of $|I| = \emptyset$ or $|J| = \emptyset$. Indeed, if $i \in I$ and $j \in J$, $i \neq j$ the rooted perimeter inequality $x_{i1} + x_{1j} + x_{ji} \leq 1$ would be violated by x. Otherwise, if there exists $i \in I \cap J$ then $x_{1i} + x_{i1} = 2$ but summing $\mathrm{RDMET}_n^\square$ inequalities:

$$
\begin{aligned}
x_{1i} + x_{ij} + x_{j1} &\leq 1 && (3.49)\\
x_{i1} - x_{ij} - x_{j1} &\leq 0 && (3.50)
\end{aligned}
$$

together yields

$$x_{1i} + x_{i1} \leq 1. \tag{3.51}$$

If both I and J are empty then x corresponds to the cut $\delta^+(V_n)$ since $x_{ij} = 1$ implies that at least one of $x_{i1} = 1$ or $x_{1j} = 1$ by (3.34).

Assume $I \neq \emptyset$, consider an index $i \in I$. For any $j \in I, i \neq j$ the perimeter inequalities (3.36) for arcs ij and ji prove that $x_{ij} = x_{ji} = 0$. Therefore all arcs ij with both endpoints in I have $x_{ij} = 0$.

Now consider any $j \notin I$. The perimeter inequality (3.36) for ji implies that $x_{ji} = x_{1j} = 0$. As this inequality is satisfied as an equation, by the linearity (3.31) we have that

$$x_{1i} + x_{ij} + x_{j1} = 1.$$

However, $x_{j1} = 0$ since $j \notin I$ and $x_{1i} = 0$ by (3.51), so $x_{ij} = 1$. Lastly, if $j, k \notin I$ then $x_{jk} = 0$ follows from the fact that $x_{jk} \leq x_{j1} + x_{1k}$, $x_{j1} = 0$ and $x_{1k} = 0$ as J is empty. We have shown that $x = \delta^+(I)$.

Assume $J \neq \emptyset$, then by the inequalities given by $RDMET_n^\square$ we can show as above that $x_{ij} = 0$ if $i, j \in J$, $i, j \notin J$ or $j \in J$ and $i \notin J$ and $x_{ij} = 1$ if $i \in J$ and $j \notin J$. This proves that $x = \delta^+(J)$.

To show that every dicut vector is a vertex of $\mathrm{RDMET}_n^\square$ we use induction on n. In this part of the proof we use the full dimensional definition of $\mathrm{RDMET}_n^\square$ formed by eliminating variables x_{ji} for $j > i \geq 2$ using the linearities. This makes verification of linear independence simpler. For the base case $n = 3$, one can easily check that $\mathrm{RDMET}_3^\square = \mathrm{DCUT}_3^\square$, see the Appendix.

For $n \geq 4$ we assume inductively that a dicut vector x that corresponds to a directed cut $\delta^+(S)$ in $\vec{K}_{n-1}$ satisfies a set of $\binom{n-1}{2} + (n-1) - 1$ linearly independent inequalities with equality. Call this set of inequalities T. We will extend T to a set of $\binom{n}{2} + n - 1$ linearly independent inequalities from $\mathrm{RDMET}_n^\square$ that are satisfied with equality. Doing so involves considering the possible cases of whether or not nodes 1 and n are in S.

Case 1: $1 \in S$ and $n \in S$.

The inequalities $x_{in} \geq 0$ for $i = 1, ..., n-1$ and $x_{n1} \geq 0$ are satisfied with equality and these n inequalities along with the inequalities in T are linearly independent.

Case 2: $1 \notin S$ and $n \notin S$.

The inequalities $x_{n1} \geq 0$, $x_{in} \geq 0$ for $i \notin S$, and $x_{in} + x_{n1} + x_{1i} \leq 1$ for $i \in S$ are satisfied with equality. These n inequalities along with the inequalities in T are all linearly independent.

Case 3: $1 \in S$ and $n \notin S$.

Firstly suppose $|S| = n - 1$. Then the n inequalities $x_{in} + x_{n1} + x_{1i} \leq 1$, $2 \leq i \leq n-1$, $x_{n1} \geq 0$, and $-x_{1n} + x_{2n} + x_{12} - x_{21} + x_{n1} \geq 0$ (corresponding to $x_{n2} \geq 0$) are satisfied as equalities, and together with the inequalities in T are linearly independent. Otherwise, $|S| \leq n-2$. Then the n inequalities $x_{in} \geq 0$ for $i \notin S$, $x_{n1} \geq 0$, $x_{1i} + x_{in} + x_{n1} \leq 1$ for all $i \in S, i \neq 1$ and $-x_{1n} + x_{jn} + x_{1j} - x_{j1} + x_{n1} \geq 0$ for some $j \notin S$, $j \neq n$ are satisfied with equality and linearly independent. Note that the last inequality corresponds to $x_{nj} \geq 0$ and the index j exists by the assumption on the cardinality of S. These inequalities along with the inequalities in T are all linearly independent.

Case 4: $1 \notin S$ and $n \in S$.

The n inequalities $x_{in} \geq 0$ for $i = 1, ..., n-1$ and $x_{n1} + x_{1i} + x_{in} \leq 1$ for some $i \notin \{1, n\}$ together with the inequalities in T are all linearly independent and are satisfied with equality. ■

The following corollary is evident as $\mathrm{RDMET}_n^{\square}$ contains a subset of the inequalities that define $\mathrm{DMET}_n^{\square}$ and no directed cuts violate inequalities of $\mathrm{DMET}_n^{\square}$.

Corollary 14 *The only integral vector of $DMET_n^\square$ are the directed cut vectors $\delta^+(S)$ for $S \subseteq V(\vec{K}_n)$ and every directed cut vector is a vertex of $DMET_n^\square$.*

The operations of permuting and collapsing that we reviewed for the cut polytope can similarly be defined for the directed cut polytope.

For a permutation σ of the nodes $\{1, ..., n\}$ and a vector $v \in \mathbb{R}^{A(\vec{K}_n)}$ we define $\sigma(v) \in \mathbb{R}^{A(\vec{K}_n)}$ as $\sigma(v)_{ij} = v_{\sigma(i)\sigma(j)}$. The following lemma trivially holds as the nodes in $\vec{K}_n$ can be relabelled.

Lemma 15 *Given $v \in \mathbb{R}^{A(\vec{K}_n)}$, $v_0 \in \mathbb{R}$ and σ a permutation of $\{1, ..., n\}$, the following statements are equivalent:*

- *The inequality $v^T x \leq v_0$ is valid (resp. facet inducing) for $DCUT_n^\square$.*
- *The inequality $\sigma(v)^T x \leq v_0$ is valid (resp. facet inducing) for $DCUT_n^\square$.*

We can define a similar type of collapsing operation that constructs a valid inequality for $\mathrm{DCUT}_m^\square$ from a valid inequality for $\mathrm{DCUT}_n^\square$, where $m < n$. Let $\pi = (M_1, ..., M_m)$ a partition of $V(\vec{K}_n)$ into m non-empty sets. If $v \in \mathbb{R}^{A(\vec{K}_n)}$ the collapse of v according to π is:

$$v_{ij}^\pi = \sum_{s \in M_i, t \in M_j} v_{st}. \tag{3.52}$$

The directed collapsing operation has many similar properties to the undirected collapsing operation. For instance, if S^π is defined to be $\cup_{k \in S} M_k$ for $S \in \{1, ..., m\}$ then $v^{\pi T} \delta^+(S) = v^T \delta^+(S^\pi)$. This gives the following lemma which has the undirected equivalent stated as Lemma 26.4.1 in Deza and Laurent's book [30].

Lemma 16 *Let $v \in \mathbb{R}^{A(\vec{K}_n)}$, $v_0 \in \mathbb{R}$ and $\pi = (M_1, ..., M_m)$ be a partition of the vertices of $V(\vec{K}_n)$. The following are true:*

1. *If $v^T x \leq v_0$ is a valid inequality for $DCUT_n^\square$ then $v_\pi^T x \leq v_0$ is a valid inequality for $DCUT_m^\square$.*
2. *If $\delta^+(S)$, $S \subseteq \{1, ..., m\}$ is a root of inequality $v^{\pi T} x \leq v_0$ then $\delta^+(S^\pi)$ is a root of $v^T x \leq v_0$.*

We leave the switching and zero-lifting operations on the directed cut polytope until later.

In characterizing some of the structural properties of directed cut polyhedra, we will show a powerful relation between the directed cut polyhedra and the undirected cut polyhedra. To this end, we begin by considering a partition of the set of all subsets of nodes of $\vec{J}_n$, $V(\vec{J}_n) = \{1, ..., n\}$, into two sets, $\mathcal{S}_1$ and $\mathcal{S}_2$. Where $\mathcal{S}_1$ contains all subsets S such that $1 \in S$ and $\mathcal{S}_2$ contains all subsets S with $1 \notin S$.

Define the polytope $\mathcal{P}_{n,1}$ to be the convex hull of directed cut vectors associated with the subsets of $\mathcal{S}_1$, ie. $\mathcal{P}_{n,1} = conv\{\delta^+(S) : S \in \mathcal{S}_1\}$. Similarly, define $\mathcal{P}_{n,2} = conv\{\delta^+(S) : S \in \mathcal{S}_2\}$. Clearly, the directed cut polytope is the convex hull of the two polytopes $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$ with the two polytopes only intersecting in a single point, the cut vector $\delta^+(V(\vec{J}_n)) = \delta^+(\emptyset)$.

The benefit of defining polytopes $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$ is the fact that both are bijections of the undirected cut polytope. We define the mappings ξ_1 (resp. ξ_2) between the cut polytope CUT_n and $\mathcal{P}_{n,1}$ (resp. $\mathcal{P}_{n,2}$) below. In Chapter 4, these mappings are used to obtain valid inequalities and facets of the directed cut polytope from valid inequalities and facets of the cut polytope.

$$\xi_1 : \mathbb{R}^{\binom{n}{2}} \to (\mathbb{R}^{\binom{n}{2}}, \{0\}^{n-1}) \ and \ \xi_2 : \mathbb{R}^{\binom{n}{2}} \to (\mathbb{R}^{\binom{n}{2}}, \{0\}^{n-1})$$

The mapping ξ_1 is defined by,

$$\begin{cases} x_{i1} = 0 & for\ 2 \le i \le n \\ x_{1i} = x_{1,i} & for\ 2 \le i \le n \\ x_{ij} = \frac{1}{2}(x_{i,j} + x_{1,j} - x_{1,i}) & for\ 2 \le i < j \le n. \end{cases}$$

equivalently ξ_1^{-1} is defined by,

$$\begin{cases} x_{1,i} = x_{1i} & for\ 2 \le i \le n \\ x_{i,j} = x_{ij} + x_{ji} = x_{1i} - x_{1j} + 2x_{ij} & for\ 2 \le i < j \le n \end{cases}$$

The mapping ξ_2 is defined by,

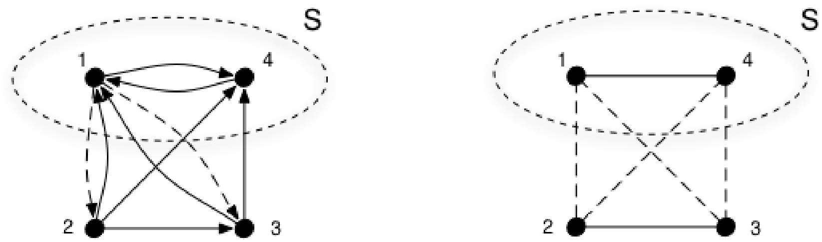
$$\begin{cases} x_{i1} = x_{1,i} & for\ 2 \le i \le n \\ x_{1i} = 0 & for\ 2 \le i \le n \\ x_{ij} = \frac{1}{2}(x_{i,j} + x_{1,i} - x_{1,j}) & for\ 2 \le i < j \le n. \end{cases}$$

equivalently ξ_2^{-1} is defined by,

$$\begin{cases} x_{1,i} = x_{i1} & for\ 2 \le i \le n \\ x_{i,j} = x_{ij} + x_{ji} = x_{j1} - x_{i1} + 2x_{ij} & for\ 2 \le i < j \le n \end{cases}$$

For any $S \subseteq \mathcal{S}_1$, ξ_1 has the property that,

$$\xi_1(\delta(S)) = \delta^+(S).$$

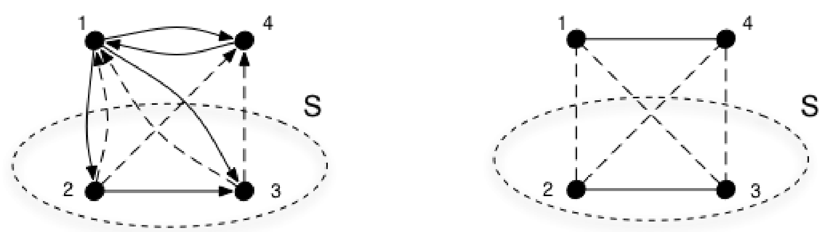


The above figure is an example for $S = \{1, 4\}$.

$$
\begin{aligned}
\xi_1^{-1}(\delta^+_{\vec{J}_4}(S)) &= \xi_1^{-1}(x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{31}, x_{34}, x_{41}) \\
&= \xi_1^{-1}((1,1,0,0,0,0,0,0)) \\
&= (1,1,0,0,1,1) \\
&= (x_{1,2}, x_{1,3}, x_{1,4}, x_{2,3}, x_{2,4}, x_{3,4}) = \delta_{K_4}(S)
\end{aligned}
$$

Similarly, for any subset S of $\mathcal{S}_2$,

$$\xi_2(\delta(S)) = \delta^+(S).$$



The above figure is an example for $S = \{2, 3\}$.

$$\begin{aligned}
\xi_2^{-1}(\delta^+_{\bar{J}_4}(S)) &= \xi_2^{-1}(x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{31}, x_{34}, x_{41}) \\
&= \xi_2^{-1}((0, 0, 0, 0, 1, 1, 1, 0)) \\
&= (1, 1, 0, 0, 1, 1) \\
&= (x_{1,2}, x_{1,3}, x_{1,4}, x_{2,3}, x_{2,4}, x_{3,4}) = \delta_{K_4}(S)
\end{aligned}$$

It follows that $\xi_1(CUT_n) = \mathcal{P}_{n,1}$ and $\xi_2(CUT_n) = \mathcal{P}_{n,2}$. This observation yields the following proposition:

Proposition 17 *The directed cut polytope is the convex hull of the linear transformation of two cut polytopes (undirected) that only intersect in a single point, the directed cut* $\delta^+(V_n) = \delta^+(\emptyset) = (0, 0, \cdots, 0)$.

Proof. By our choice of $\mathcal{S}_1$ and $\mathcal{S}_2$ the theorem trivially holds by the fact the polyhedra $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$ are linear bijective mappings of $\mathrm{CUT}_n^\square$. ■

The relation between $\mathrm{CUT}_n^\square$, $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$ gives rise to the following proposition which has a similar flavour to Proposition 6.

Proposition 18 *Let* $a \in \mathbb{R}^{\binom{n}{2}}$, $b, c \in \mathbb{R}^{\binom{n}{2}+n-1}$ *and* $\alpha \in \mathbb{R}$ *where,*

$$\begin{cases}
b_{1i} = 0 & for\ 2 \leq i \leq n \\
b_{i1} = a_{1,i} + \sum_{k=2}^{i-1} a_{k,i} - \sum_{j=i+1}^{n} a_{i,j} & for\ 2 \leq i \leq n \\
b_{ij} = 2a_{i,j} & for\ 2 \leq i < j \leq n
\end{cases}$$

and

$$\begin{cases} c_{1i} = a_{1,i} - \sum_{k=2}^{i-1} a_{k,i} + \sum_{j=i+1}^{n} a_{i,j} & for\ 2 \leq i \leq n \\ c_{i1} = 0 & for\ 2 \leq i \leq n \\ c_{ij} = 2a_{i,j} & for\ 2 \leq i < j \leq n. \end{cases}$$

The inequality,

$$\sum_{1 \leq i < j \leq n} a_{i,j} x_{i,j} \leq \alpha$$

is valid (resp. facet defining) for the cut polytope if and only if the inequality

$$\sum_{i=2}^{n} b_{i1} x_{i1} + \sum_{1 \leq i < j \leq n} b_{ij} x_{ij} \leq \alpha$$

is valid (resp. facet defining) for the polytope $\mathcal{P}_{n,1}$ which is in turn valid (resp. facet defining) if and only if the inequality

$$\sum_{i=2}^{n} c_{i1} x_{i1} + \sum_{1 \leq i < j \leq n} c_{ij} x_{ij} \leq \alpha$$

is valid (resp. facet defining) for the polytope $\mathcal{P}_{n,2}$.

We obtain the following table of relations between facets of CUT_n, $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$.

$CUT_n^{\square}$	$\mathcal{P}_{n,2}$	$\mathcal{P}_{n,1}$
$x_{1,j} - x_{1,i} - x_{i,j} \leq 0$	$x_{ij} \geq 0$	$x_{1j} - x_{1i} - x_{ij} \leq 0$
$x_{1,i} - x_{1,j} - x_{i,j} \leq 0$	$x_{i1} - x_{ij} - x_{j1} \leq 0$	$x_{ij} \geq 0$
$x_{i,j} - x_{1,i} - x_{1,j} \leq 0$	$x_{ij} \leq x_{i1}$	$x_{ij} \leq x_{1i}$
$x_{1,i} + x_{i,j} + x_{1,j} \leq 2$	$x_{i1} + x_{ij} \leq 1$	$x_{1i} + x_{ij} \leq 1$

for $2 \leq i < j \leq n$, and

$CUT_n^{\square}$	$\mathcal{P}_{n,2}$	$\mathcal{P}_{n,1}$
$x_{i,k} - x_{i,j} - x_{j,k} \leq 0$	$x_{ik} - x_{ij} - x_{jk} \leq 0$	$x_{ik} - x_{ij} - x_{jk} \leq 0$
$x_{i,j} - x_{i,k} - x_{j,k} \leq 0$	$x_{ij} - x_{ik} - x_{jk} + x_{j1} - x_{k1} \leq 0$	$x_{ij} - x_{ik} - x_{jk} - x_{1j} + x_{1k} \leq 0$
$x_{j,k} - x_{i,j} - x_{i,k} \leq 0$	$x_{jk} - x_{ik} - x_{ij} + x_{i1} - x_{j1} \leq 0$	$x_{jk} - x_{ik} - x_{ij} + x_{1j} - x_{1i} \leq 0$
$x_{i,j} + x_{j,k} + x_{i,k} \leq 2$	$x_{ij} + x_{ik} + x_{jk} - x_{i1} + x_{k1} \leq 1$	$x_{ij} + x_{ik} + x_{jk} + x_{1i} - x_{1k} \leq 1$

for $2 \leq i < j < k \leq n$. In the following chapter we will use this relation between directed and undirected cuts to prove that certain inequalities are facets of the directed cut polytope.

CHAPTER 4
Facets of the directed cut polytope and cone

In this chapter we will use the relation between the directed and undirected cuts established by the mappings ξ_1 and ξ_2 to extend previously known structural properties of the cut polyhedra to the directed cut polyhedra. Theorem 19 below allows us to characterize many different facets of the directed cut polytope from knowledge of past work on the cut polytope.

To begin, we will need to define some terms and notation. For a graph $G = (V, E)$ the *support graph*, $G(a) = (V(a), E(a))$, of a vector $a \in \mathbb{R}^E$ is the graph with edges $E(a) = \{e | a_e \neq 0 \, e \in E\}$ and nodes $V(a)$ such that every nodes in $V(a)$ is an endpoint of at least one edge in $E(a)$. For $a \in \mathbb{R}^E$ and $a_0 \in \mathbb{R}$ inequality $a^T x \leq a_0$ is said to be completely support by $F \subset E$ when $E(a) \subseteq F$. We refer to an inequality $a^T x \leq \alpha$ as *non-trivial* if its support graph doesn't have one node common to every edge, ie. $G(a)$ is not a star.

Theorem 19 *If $a^T x \leq \alpha$ is a facet of the undirected cut polytope $CUT_n^\square$ then*

$$\sum_{2 \leq i < j \leq n} 2a_{i,j} x_{ij} + \sum_{i=2}^{n} c_{i1} x_{i1} + \sum_{i=2}^{n} b_{1i} x_{1i} \quad \leq \quad \alpha \tag{4.1}$$

is a facet of the directed cut polytope $DCUT_n^\square$.

To prove Theorem 19, Lemma 20 on facets of the cut polytope will be needed. This lemma is a generalization of Lemma 26.5.2 of [30] which was stated in Section 3.3 as Lemma 9.

Our version of Lemma 26.5.2 below deals with inequalities of the form $v^T x \leq \alpha$ where α is not necessarily zero. This is needed as the directed cut polyhedra are not closed under switching.

Lemma 20 *Let $v^T x \leq \alpha$ be a valid inequality for $CUT_n^{\square}$ and let F be a subset of $E(K_n)$. If the inequality $a^T x \leq \alpha$ is facet inducing and $v_{\bar{F}} \neq 0$, then $rank(R(v)_F) = |F|$.*

We follow much of the same proof structure as Lemma 26.5.2 in [30] with modifications due to the fact the Lemma 26.5.2 deals with homogeneous inequalities and we have non-homogeneous inequalities of the form $v^T x \leq \alpha$ where α can be strictly positive.

Proof.

If $a^T x \leq \alpha$ is facet inducing we can find a set A of $\binom{n}{2}$ affinely independent roots $x_1, ..., x_{\binom{n}{2}}$. Let $T = \{x_i - x_{\binom{n}{2}} : 1 \leq i \leq \binom{n}{2} - 1\}$. The vectors in T are linearly independent. Consider a $\binom{n}{2} - 1 \times \binom{n}{2}$ matrix M where the rows of M are the $\binom{n}{2} - 1$ linearly independent vectors of T. Since the rank of M is $\binom{n}{2} - 1$, $\binom{n}{2} - 1$ of the columns are linearly independent.

If the columns of M corresponding to the set F have rank $|F| - 1$, consider partitioning the set T into three disjoint sets T_1, T_2 and T_3. Let T_1 consist of $|F| - 1$ vectors from T whose projection on F are linearly independent. Let T_2 be the set of vectors in T whose projection on F are 0 and let T_3 be $T \setminus (T_1 \cup T_2)$. We know that $|T_2 \cup T_3|$ is $\binom{n}{2} - |F|$ and the set $T_2 \cup T_3$ is linearly independent and consists of roots of $v^T x \leq 0$ as $y_i = x_i - x_{\binom{n}{2}}$ for $y_i \in T$ and $v^T y = v^T x_i - v^T x_{\binom{n}{2}} = \alpha - \alpha = 0$.

The projection of a vector from T_3 onto F can be expressed in terms of a convex combination of vectors from the set T_1. For $x \in T_3$ we can write:

$$x_F = \sum_{x_i \in T_1} \lambda_i (x_i)_F.$$

Create the set T_3' from T_3 by replacing $x \in T_3$ by:

$$x' \;\;=\;\; x - \sum_{x_i \in T_1} \lambda_i x_i$$

where $\sum_{x_i \in T_1} \lambda_i = 1$. The new set T_3' has the property that the projection of its elements onto F are zero and they are roots of $v^T x = 0$ since $x' = x - \sum_{x_i \in T_1} \lambda_i x_i$ and $v^T x' = v^T x - \sum_{x_i \in T_1} \lambda_i v^T x_i = 0 - \sum_{x_i \in T_1} \lambda_i 0 = 0$.

The set T_3' was constructed using elementary row operations from T_1 and T_3 where the vectors in $T_1 \cup T_2 \cup T_3$ are linearly independent. It follows that $T_2 \cup T_3'$ are linearly independent. Any $x \in T_2 \cup T_3'$ has $x_F = 0$ and $v^T x = 0$ which means $v_{\bar{F}} = 0$ as $|T_2 \cup T_3'| = \binom{n}{2} - |F| = |\bar{F}|$.

If all the columns of M corresponding to the set F are linearly independent, ie. $rank(A_F) = |F|$, then columns of M corresponding to $\bar{F}$ have rank $|\bar{F}| - 1$ and we can use the above reasoning to deduce that $v_F = 0$ and $v_{\bar{F}} \neq 0$. ■

With Lemma 20 in hand we can proceed with the proof of Theorem 19. A graph is called a *star* if it has a node that is common to every edge.

Proof of Theorem 19.

To show that (4.1) defines a facet, we begin by considering the case where $G(a)$ is not a star in which all edges contain vertex 1.

Since $a^T x \leq \alpha$ is a facet of the cut polytope, it follows that we can find $\binom{n}{2}$ affinely independent roots $\delta(S_i)$ $(1 \leq i \leq \binom{n}{2})$ of $a^T x \leq \alpha$ such that $1 \in S_i$. Choose $F = \{(1,2),(1,3),\cdots,(1,n)\}$, applying Lemma 20 we get that $rank(R(a)_F) = |F| = n-1$ as the facet inducing inequality is non-trivial. Let $\delta(T_i)$ $(1 \leq i \leq n-1)$ be $n-1$ roots of $a^T x \leq \alpha$ whose projections on F are linearly independent. We can assume that $1 \notin T_i$, since if $1 \in T_i$ we can replace T_i with $V(K_n) \backslash T_i$ (ie. $\delta^+(T_i) = \xi_2(\delta(T_i))$).

We claim the set of directed cut vectors $C = \{\delta^+(S_i) : 1 \leq i \leq \binom{n}{2}\} \cup \{\delta^+(T_i) : 1 \leq i \leq n-1\}$ are $\binom{n}{2} + n - 1$ affinely independent roots of the inequality $\sum_{2 \leq i<j \leq n} 2a_{ij}x_{ij} + \sum_{i=2}^{n} b_{1i}x_{1i} + \sum_{i=2}^{n} c_{i1}x_{i1} \leq \alpha$. By construction, every cut in C is a root, so we simply need to show that they are affinely independent.

Consider the square matrix M whose rows are first the $\binom{n}{2}$ directed cut vectors $\delta^+(S_i)$ followed by the $n-1$ vectors $\delta^+(T_i)$, Index the columns of M by the sets $I \cup J$ where $I = \{ij : 1 \leq i < j \leq n\}$ and $J = \{i1 : 2 \leq i \leq n\}$. M has the form:

$$M = \begin{pmatrix} X & 0 \\ Z & Y \end{pmatrix}$$

The matrix X is affinely independent as the vectors $\delta^+(S_i)$ are affinely independent. The matrix Y has full row rank, since its rows $\delta^+(T_i)_J = \delta(T_i)_F$ are linearly independent.

To complete the proof, we will show that the support graph $G(a)$ cannot be a star with all edges containing vertex 1. For suppose it was, then the inequality $a^T x \leq \alpha$ becomes

$$\sum_{1j \in E(G(a))} a_{1,j}x_{1,j} \leq \alpha \tag{4.2}$$

Let the cut vector $\delta(S)$ be a root of (4.2), so that $a^T\delta(S) = \alpha$. We may assume that $1 \in S$. Suppose first that for each $j \in \overline{S}$, $a_{1,j} > 0$. If $S = \{1\}$, (4.2) does not define a facet, since it is a non-negative combination of valid inequalities of the form $a_{1,j} \leq 1$. Otherwise let k be any other element of S. If $a_{1,k} > 0$ then we have a contradiction, since $a^T\delta(S \setminus \{k\}) > \alpha$. So $a_{1,k} < 0$ for all $k \in S$ and it follows that (4.2) does not have any roots besides $\delta(S)$, a contradiction. Therefore there must be some $j \in \overline{S}$ for which $a_{1,j} < 0$. We again have a contradiction because $a^T\delta(S \cup \{j\}) > \alpha$. ■

4.1 The triangle inequalities

Using the bijections ξ_1 and ξ_2 one can obtain sets of facets for DCUT_n and $\mathrm{DCUT}_n^{\square}$ from the triangle inequalities of the cut cone and polytope. Recall that the triangle inequalities for the cut cone and polytope are:

$$\begin{aligned} x_{i,k} - x_{i,j} - x_{j,k} &\leq 0 \\ x_{i,j} - x_{i,k} - x_{j,k} &\leq 0 \\ x_{j,k} - x_{i,j} - x_{i,k} &\leq 0 \end{aligned} \tag{4.3}$$

for $1 \leq i < j < k < n$. The additional triangle inequalities, known as the perimeter inequalities, for the cut polytope are:

$$x_{i,j} + x_{i,k} + x_{j,k} \leq 2 \tag{4.4}$$

for $1 \leq i < j < k < n$.

Using Theorem 19 and the fact that (4.3) are facets of CUT_n one can easily see that the following corollary is true.

Corollary 21 *The following inequalities:*

$$x_{ij} \geq 0 \tag{4.5}$$

$$x_{ik} \leq x_{ij} + x_{jk} \tag{4.6}$$

$$x_{ik} - x_{ij} - x_{jk} \leq 0 \tag{4.7}$$

$$x_{ij} - x_{ik} - x_{jk} + x_{j1} - x_{k1} \leq 0 \tag{4.8}$$

$$x_{jk} - x_{ik} - x_{ij} + x_{i1} - x_{j1} \leq 0 \tag{4.9}$$

for $1 \leq i < j < k \leq n$ *are facet defining inequalities of* $DCUT_n$.

Similarly, the fact that the perimeter inequalities (4.4) are facet inducing inequalities of $\mathrm{CUT}_n^\square$ implies the following corollary which is a straight forward application of Theorem 19.

Corollary 22 *The inequalities:*

$$x_{ij} + x_{jk} + x_{ki} \leq 1 \tag{4.10}$$

for $1 \leq i < j < k \leq n$ *are facet inducing inequalities of* $DCUT_n^\square$.

4.2 Pentagonal inequalities

The *pentagonal inequalities* are a class of facet defining inequalities for CUT_n that are slightly more complex than the triangle inequalities. They have the general form:

$$x_{i,j} + x_{j,k} + x_{k,i} + x_{l,m} \leq x_{i,l} + x_{j,l} + x_{k,l} + x_{i,m} + x_{j,m} + x_{k,m}. \tag{4.11}$$

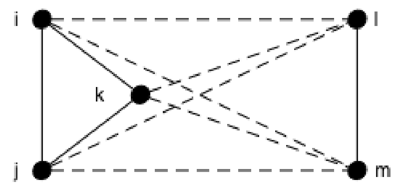


Figure 4–1: The pentagonal inequality implies that the sum of weights on the dashed edges must be at least as large as the sum of weights on the solid edges

Using Theorem 19 we get a class of inequalities that are facet defining for DCUT_n and $\mathrm{DCUT}_n^\square$. The pentagonal inequalities for $\mathrm{DCUT}_5^\square$ are:

$$\begin{aligned}
x_{21} - x_{23} + x_{24} - x_{25} - x_{31} - x_{34} + x_{35} + x_{41} - x_{45} - x_{51} &\leq 0 \\
-x_{12} + x_{13} - x_{23} - x_{24} + x_{25} + x_{34} - x_{35} + x_{41} - x_{45} - x_{51} &\leq 0 \qquad (4.12) \\
-x_{12} + x_{13} - x_{14} + x_{15} - x_{23} + x_{24} - x_{25} - x_{34} + x_{35} - x_{45} &\leq 0 \\
-x_{14} + x_{15} + x_{21} - x_{23} - x_{24} + x_{25} - x_{31} + x_{34} - x_{35} - x_{45} &\leq 0 \\
-x_{13} + x_{14} - x_{15} - x_{21} + x_{23} - x_{24} + x_{25} - x_{34} + x_{35} - x_{45} &\leq 0 \\
-x_{15} - x_{21} - x_{23} + x_{24} + x_{25} + x_{31} - x_{34} - x_{35} - x_{41} + x_{45} &\leq 0 \\
-x_{12} - 2x_{13} + 2x_{14} + x_{15} + x_{23} - x_{24} - x_{25} + x_{31} & \\
-x_{34} - x_{35} - x_{41} + x_{45} &\leq 0 \qquad (4.13) \\
-x_{13} - 2x_{14} + 2x_{15} - x_{21} + x_{23} + x_{24} - x_{25} + x_{34} & \\
-x_{35} + x_{41} - x_{45} - x_{51} &\leq 0 \\
-x_{13} + x_{14} + x_{21} + x_{23} - x_{24} - x_{25} + 2x_{31} - x_{34} & \\
-x_{35} - 2x_{41} + x_{45} - x_{51} &\leq 0 \\
-x_{12} + x_{13} - x_{15} + 2x_{21} - x_{23} - x_{24} - x_{25} - 2x_{31} & \\
+x_{34} + x_{35} - x_{41} + x_{45} &\leq 0
\end{aligned}$$

The figure below shows the directed pentagonal inequality for the directed cut polyhedra.

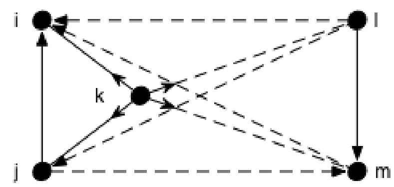


Figure 4–2: The directed pentagonal inequality implies that the sum of weights on the dashed arcs must be at least as large as the sum of weights on the solid arcs

For graph $\vec{J}_n$ other forms of pentagonal arise as arcs ji do not exist for $j > i$. The inequalities (4.12) and (4.13) have such forms. When arcs ji for $j > i$ exist in a directed pentagonal inequality, the form valid for $\vec{J}_n$ can be obtained by replacing x_{ji} by $x_{ij} + x_{j1} + x_{1i} - x_{1j} - x_{i1}$.

Using Theorem 19 we get:

Corollary 23 *The directed pentagonal inequalities:*

$$x_{ik} + x_{km} + x_{im} + x_{jl} \leq x_{ij} + x_{il} + x_{jk} + x_{jm} + x_{kl} + x_{lm} \tag{4.14}$$

are facet defining inequalities for $DCUT_n$ *and* $DCUT_n^\square$ *when* $n \geq 5$.

4.3 Hypermetric inequalities

The hypermetric inequalities are valid inequalities for $\mathrm{CUT}_n^\square$ and CUT_n which generalize the triangle and pentagonal inequalities. Similarly, they give rise to valid inequalities of the directed cut cone and polytope. Let $b = (b_1, \cdots, b_n)$ be an integral vector such that $\sum_{i=1}^n b_i = 1$, the inequality:

$$\sum_{1 \leq i < j \leq n} b_i b_j x_{i,j} \leq 0 \tag{4.15}$$

is known as a hypermetric inequality. Every hypermetric inequality is known to be valid for the cut cone, and the roots of a hypermetric inequality are the cut vectors $\delta(S)$ for which $\sum_{i\in S} b_i$ is 0 or 1.

Using our mapping from the cut cone to the directed cut cone, hypermetric inequalities for DCUT_n take the form:

$$\sum_{i=2}^{n}(b_1 - \sum_{k=2}^{i-1} b_k + \sum_{j=i+1}^{n} b_j)b_i x_{1i} + \sum_{i=2}^{n}(b_1 + \sum_{k=2}^{i-1} b_k - \sum_{j=i+1}^{n} b_j)b_i x_{i1} + \sum_{2\leq i<j\leq n} b_i b_j x_{ij} \leq 0 \quad (4.16)$$

where $\sum_{i=1}^{n} b_i = 1$.

The *pure hypermetric* inequalities have the form $b = (1, \cdots, 1, -1, \cdots, -1)$. The pentagonal facet for CUT_n has $b = (1, 1, 1, -1, -1)$. Observe that, as noted above, the pure hypermetric facet for CUT_n generalizes the triangle and pentagonal facets.

4.4 Zero-lifting the directed cut polytope

To prove a zero-lifting theorem for the directed cut polytope, we will need a second variant of Lemma 26.5.2. This variant, Lemma 24, is an identical result to Lemma 26.5.2 of [30] but for the directed cut polytope not the cut polytope.

Lemma 24 *Let $v^T x \leq 0$ be a valid inequality for $DCUT_n^{\square}$ and let $R(v)$ denote its set of roots. Let F be a subset of $A(\vec{J}_n)$.*

(i) If $rank(R(v)_F) = |F|$ and $rank(R(v)^F) = |\bar{F}| - 1$, then the inequality $v^T x \leq 0$ is facet inducing.

(ii) If the inequality $v^T x \leq 0$ is facet inducing and $v_{\bar{F}} \neq 0$, then $rank(R(v)_F) = |F|$.

The proof of Lemma 24 is a straight forward from the proof of Lemma 26.5.2 of [30]. We include it here for completeness but it required no substantial alterations.

Proof. (i) By the assumptions, a set A of $|F|$ linearly independent roots can be found whose projections on the arcs F are linearly independent. Likewise, a set B of roots of $v^T x \leq 0$ can be found whose projections on F are the zero vector where the vectors of B are linearly independent and $|B| = |\bar{F}| - 1$. It is easy to see that the vectors $A \cup B$ are a set of $\binom{n}{2} + n - 2$ linearly independent roots of $v^T x \leq 0$ which imply that $v^T x \leq 0$ is a facet of DCUT_n.

(ii) If $v^T x \leq 0$ is a facet of DCUT_n, we can find a set A of $\binom{n}{2} + n - 2$ linearly independent roots of $v^T x \leq 0$. If we construct a matrix M by using the vectors A as the rows, we have a $(\binom{n}{2} + n - 2) \times (\binom{n}{2} + 2 - 1)$ matrix with linearly independent rows. This means that all but one column of M are linearly independent.

If all of the columns corresponding to arcs of F aren't linearly independent then $rank(A_F) = |F| - 1$. Let $T_1 \subseteq A$ be $|F| - 1$ vectors whose projection on F are linearly independent, let $T_2 \subseteq A$ be the vectors of A whose projection on F are the zero vector and let $T_3 = A \setminus (T_1 \cup T_2)$.

For $x \in T_3$ we can express x_F (the projection of x onto the arcs of F) as a convex combination of vectors of T_1. ie. $x_F = \sum_{y_i \in T_1} \lambda_i (y_i)_F$. A new set T_3' can be constructed where for each x in T_3, we add $x' = x - \sum_{y_i \in T_1} \lambda_i y_i$ to T_3'. The vectors in the set $T_2 \cup T_3'$ are linearly independent. It follows that $v_{\bar{F}} = 0$ as we have $|T_2 \cup T_3'| = |\bar{F}|$ linearly independent vectors satisfying $v^T x = 0$ with $x_F = 0$.

If the columns corresponding to the arcs of F are linearly independent then $rank(A_{\bar{F}}) = |\bar{F}| - 1$ and a similar argument as above can be applied to show that $v_{\bar{F}} \neq 0$. ■

We can now state and prove our zero-lifting theorem for the directed cut cone.

Theorem 25 *Given* $v \in \mathbb{R}^{|A(\vec{K}_n)|}$ *and zero-lifted* $v' \in \mathbb{R}^{|A(\vec{K}_{n+1})|}$ *the following are equivalent.*

- $v^T x \leq 0$ *is facet inducing for* $DCUT_n$*.*
- $v'^T x \leq 0$ *is facet inducing for* $DCUT_{n+1}$*.*

Proof. Assume that $v'^T x \leq 0$ is facet inducing for DCUT_{n+1} and let $R(v')$ denote its roots. Let $F = \{(1, n+1), ..., (n, n+1)\} \cup (n+1, 1)$ and $\bar{F} = A(\vec{J}_n)$. Using Lemma 24 and the fact that $v'_F = 0$ we know that the rank of $R(v')_F$ is equal to $\binom{n}{2} + n - 2$ which implies that $v^T x \leq 0$ is a facet of DCUT_n.

Assume that $v^T x \leq 0$ is facet inducing for DCUT_n. Let $F = \{1n, 2n, ..., (n-1, n)\}$, as $v \neq 0$ we can assume that $v_{\bar{F}} \neq 0$ (the zero-lifting theorem is not needed for trivial facets like the non-negativity constraint). By Lemma 24, $rank(R(v)_F) = |F|$. Let $T_j \subseteq V(\vec{J}_n)$, $j = 1, ..., n-1$ be $|F|$ sets such that the projections of $\delta^+(T_j)$ $j = 1, ..., n-1$ onto the arc set F are linearly independent. Let S_k, $k = 1, ..., \binom{n}{2} + n - 2$ be subsets of $V(\vec{J}_n)$ such that $\delta^+(S_k)$ are linearly independent roots of $v^T x \leq 0$.

Let $S'_k = S_k \cup \{n+1\}$ for $k = 1, ..., \binom{n}{2} + n - 2$. We claim the vectors $\delta^+(S'_k) \cup \delta^+(T_j) \cup \delta^+(\{1, ..., n\}) \cup \delta^+(\{n+1\})$ for $j = 1, ..., n-1$ and $k = 1, ..., \binom{n}{2} + n - 2$ form $n - 1 + \binom{n}{2} + n - 2 + 2 = \binom{n+1}{2} - (n+1) - 2 = Dim(DCUT_{n+1}) - 1$ linearly independent roots of $v'^T x \leq 0$. This proves that $v'^T x \leq 0$ is facet inducing for DCUT_{n+1}.

To see that this claim is true consider the matrix M consisting of the vectors $\delta^T+(T_j)$, $\delta^+(S'_k)$ and $\delta^+(\{n+1\})$ as rows for $j=1,...,n-1$ and $k=1,...,\binom{n}{2}+n-2$. Let the columns of M be indexed by the arcs: $21, 31, ..., n1$ then ij for $1 \le i < j \le n$ followed by $(1,n+1),(2,n+1),...,(n-1,n+1)$ then $(n+1,1)$ and lastly $(n,n+1)$. The matrix M has the form:

$$M = \begin{pmatrix} X & 0 & A & 0 \\ Z & Y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where the columns of the submatrix X are indexed by the arcs $21, 31, ..., n1$ followed by arcs ij for $1 \le i < j \le n$. Matrix X is linearly independent as it corresponds to the $\binom{n}{2}+n-2$ linearly independent vectors $\delta^+(S'_k)$. The entry A is a column vector corresponding to the $(n+1,1)$ entry of the $\delta^+(S'_k)$ vectors. Submatrix Y's columns corresponds to the arcs $(1,n+1),(2,n+1)...,(n-1,n+1)$. As the entries of $(i,n+1)$ are identical to in in the $\delta^+(T_j)$ vectors it follows that Y is linearly independent. As X and Y are linearly independent it is straight forward to see that M is linearly independent. ■

In the next section we will use Theorem 25 to show that a family of inequalities are facets of $\mathrm{DCUT}_n^{\square}$. This family seems unrelated to the mapping of facets of the cut polytope so we can't apply Theorem 19.

4.5 Facets of $\mathbf{DCUT}_4^{\square}$

The vertices of $\mathrm{DCUT}_4^{\square}$ are the directed cuts:

x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
(0	0	0	0	0	0	0	0	0	0	0	0)
(1	1	1	0	0	0	0	0	0	0	0	0)
(0	0	0	1	1	1	0	0	0	0	0	0)
(0	1	1	0	1	1	0	0	0	0	0	0)
(0	0	1	0	0	1	0	0	1	0	0	0)
(0	0	0	1	0	1	1	0	1	0	0	0)
(1	0	1	0	0	0	0	1	1	0	0	0)
(0	0	0	0	0	0	1	1	1	0	0	0)
(0	0	0	1	0	0	1	0	0	1	0	0)
(0	0	0	1	1	0	0	0	0	1	0	1)
(1	0	0	0	0	0	0	1	0	0	1	0)
(0	0	0	0	0	0	1	1	0	1	1	0)
(0	0	0	0	0	0	0	0	0	1	1	1)
(1	1	0	0	0	0	0	0	0	0	1	1)
(0	1	0	0	1	0	0	0	0	0	0	1)

Computing the facet defining inequalities for $\mathrm{DCUT}_4^{\square}$ using lrs [4] we get the linearities:

$$x_{13} + x_{32} + x_{21} = x_{31} + x_{12} + x_{23} \tag{4.17}$$

$$x_{12} + x_{24} + x_{41} = x_{14} + x_{42} + x_{21} \tag{4.18}$$

$$x_{13} + x_{34} + x_{41} = x_{14} + x_{43} + x_{31}. \tag{4.19}$$

The non-negativity constraints (note that the variables x_{32}, x_{42} and x_{43} have been removed by lrs as $DCUT(\vec{K}_4)$ is not full dimensional):

$$x_{12} \geq 0 \tag{4.20}$$

$$x_{13} \geq 0 \tag{4.21}$$

$$x_{14} \geq 0 \tag{4.22}$$

$$x_{21} \geq 0 \tag{4.23}$$

$$x_{23} \geq 0 \tag{4.24}$$

$$x_{24} \geq 0 \tag{4.25}$$

$$x_{31} \geq 0 \tag{4.26}$$

$$x_{34} \geq 0 \tag{4.27}$$

$$x_{41} \geq 0 \tag{4.28}$$

and following inequalities:

$$
\begin{aligned}
1 - x_{12} + x_{14} - x_{34} - x_{41} &\geq 0 && (4.29)\\
1 - x_{12} - x_{23} - x_{31} &\geq 0 && (4.30)\\
1 - x_{13} - x_{34} - x_{41} &\geq 0 && (4.31)\\
1 - x_{12} + x_{14} + x_{21} - x_{23} - x_{24} - x_{34} - x_{41} &\geq 0 && (4.32)\\
1 - x_{12} - x_{21} - x_{41} &\geq 0 && (4.33)\\
1 - x_{12} + x_{21} - x_{23} - x_{41} &\geq 0 && (4.34)\\
x_{13} - x_{14} + x_{21} - x_{23} + x_{24} + x_{34} &\geq 0 && (4.35)\\
x_{13} - x_{14} + x_{34} &\geq 0 && (4.36)\\
1 - x_{13} + x_{14} - x_{24} - x_{41} &\geq 0 && (4.37)\\
x_{12} - x_{13} + x_{14} + x_{23} - x_{24} + x_{34} &\geq 0 && (4.38)\\
x_{23} - x_{24} + x_{34} &\geq 0 && (4.39)\\
x_{12} - x_{13} - x_{21} + x_{23} + x_{31} &\geq 0 && (4.40)\\
x_{21} + x_{23} + x_{31} &\geq 0 && (4.41)\\
1 - x_{13} - x_{21} + x_{31} - x_{34} &\geq 0 && (4.42)\\
x_{12} - x_{13} - 2x_{21} + x_{23} + x_{24} + 2x_{31} - x_{34} + x_{41} &\geq 0 && (4.43)\\
x_{12} - x_{13} + x_{23} &\geq 0 && (4.44)\\
x_{14} + x_{21} - x_{24} &\geq 0 && (4.45)\\
x_{31} + x_{34} + x_{41} &\geq 0 && (4.46)\\
1 - x_{12} + x_{21} - x_{24} - x_{31} &\geq 0 && (4.47)\\
x_{21} + x_{23} - x_{24} - x_{31} + x_{34} + x_{41} &\geq 0 && (4.48)\\
x_{14} - x_{21} + x_{23} + x_{24} + x_{31} - x_{34} &\geq 0 && (4.49)\\
x_{21} + x_{24} + x_{41} &\geq 0 && (4.50)\\
x_{12} - x_{13} - x_{21} + x_{23} + x_{24} + x_{31} - x_{34} &\geq 0 && (4.51)\\
x_{14} + x_{31} - x_{34} &\geq 0 && (4.52)\\
x_{12} - x_{14} + x_{24} &\geq 0 && (4.53)\\
x_{13} + x_{21} - x_{23} &\geq 0 && (4.54)\\
x_{13} - x_{14} - x_{23} + x_{24} - x_{31} + x_{34} + x_{41} &\geq 0 && (4.55)\\
x_{12} - x_{14} - x_{21} + x_{24} + x_{41} &\geq 0 && (4.56)\\
x_{13} - x_{14} - x_{31} + x_{34} + x_{41} &\geq 0 && (4.57)\\
1 + x_{13} - x_{14} - x_{23} - x_{31} &\geq 0 && (4.58)\\
x_{12} + 2x_{13} - 2x_{14} - x_{23} + x_{24} - x_{31} + x_{34} + x_{41} &\geq 0. && (4.59)
\end{aligned}
$$

Now one can easily check that the directed cut vector $\delta^+(\{3\})$ has a total of 27 incident facets which are: (4.29), (4.30), (4.22), (4.31), (4.25), (4.32), (4.24), (4.23), (4.44), (4.45), (4.46), (4.28), (4.47), (4.48), (4.49), (4.50), (4.51), (4.52), (4.21), (4.53), (4.54), (4.20), (4.55), (4.56), (4.57), (4.58) and (4.59).

While the directed cut vector $\delta^+(\{1,2\})$ has a total of 29 incident facets which are: (4.30), (4.31), (4.32), (4.33), (4.34), (4.35), (4.26), (4.36), (4.37), (4.27), (4.38), (4.39), (4.40), (4.42), (4.23), (4.44), (4.45), (4.46), (4.28), (4.47), (4.48), (4.53), (4.54), (4.20), (4.55), (4.56), (4.57), (4.58) and (4.59). The result on the structure of these two vertices implies that there is no hope of finding a facets preserving automorphism like switching that takes a given vertex to any other *arbitrarily* chosen vertex. As different vertices have fundamentally different structures.

The relaxation $\mathrm{DMET}_4^\square$ of $\mathrm{DCUT}_4^\square$ has a total of 21 vertices, 15 correspond to directed cuts and 6 half-integral fractional vertices. The fractional entries are:

x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
(0,	1/2,	0,	1/2,	1/2,	1/2,	1/2,	0,	0,	1/2,	1/2,	1/2)
(0,	0,	1/2,	1/2,	1/2,	1/2,	1/2,	1/2,	1/2,	1/2,	0,	0)
(1/2,	0,	0,	1/2,	0,	0,	1/2,	1/2,	1/2,	1/2,	1/2,	1/2)
(1/2,	1/2,	1/2,	0,	1/2,	0,	0,	1/2,	0,	1/2,	1/2,	1/2)
(1/2,	1/2,	1/2,	0,	0,	1/2,	1/2,	1/2,	1/2,	0,	1/2,	0)
(1/2,	1/2,	1/2,	1/2,	1/2,	1/2,	0,	0,	1/2,	0,	0,	1/2)

To complete the description of the convex hull of $\mathrm{DCUT}_4^\square$, other inequalities are needed. These inequalities are not related to the previously discussed triangle or hypermetric inequalities. They are of two types, one has the form:

$$x_{13} + x_{24} \leq x_{12} + x_{34} + x_{14} + x_{23}. \tag{4.60}$$

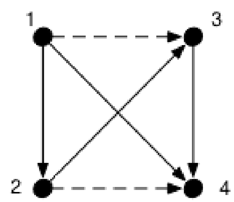


Figure 4–3: The sum of weights on the solid arcs must be greater than or equal to the sum of the weights on the dashed arcs.

The other has the form:

$$x_{31} + x_{12} + x_{24} - x_{21} \leq 1 \tag{4.61}$$

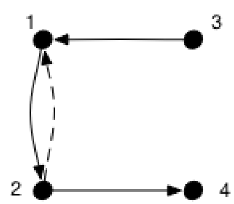


Figure 4–4: The sum of the weights on the solid arcs minus the weight on the dashed arc must be less than or equal to 1

To prove that these inequalities are facet defining for $\mathrm{DCUT}_n^{\square}$ we simply need to show that they are facets of $\mathrm{DCUT}_4^{\square}$ and zero-lift them.

Theorem 26 *The inequality:*

$$x_{ik} + x_{jl} \leq x_{ij} + x_{kl} + x_{il} + x_{jk} \tag{4.62}$$

is facet inducing for $DCUT_n$.

Proof. As DCUT_4 has dimension 9, listing 9 affinely independent roots proves that (4.62) is a facet of DCUT_4. The following are such a set of 9 cuts:

$$\delta^+(\emptyset), \delta^+(\{2\}), \delta^+(\{1,2\}), \delta^+(\{2,3\}), \delta^+(\{2,3,4\}), \delta^+(\{3,4\}), \delta^+(\{4\}), \delta^+(\{1,4\}), \delta^+(\{1,2,4\}).$$

Applying Theorem 25 gives the result that (4.62) is a facet inducing inequality for DCUT_n and $\mathrm{DCUT}_n^\square$. ■

To prove that (4.61) is a facet defining inequality, a zero lifting result would be needed for non-homogeneous inequalities on the directed cut polytope. We prove such a lifting result in Section 6.2. Proving this result is more difficult than proving it for the undirected cut polytope. For the undirected case proving a zero-lifting theorem for a homogeneous inequality implies a zero-lifting theorem for non-homogeneous inequalities by switching. We discuss switching the directed cut further in the following section.

4.6 Switching directed cuts

Since $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$ are linear bijections of the cut polytope there must be an analogous operation to switching for $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$. Given $A \subseteq V_n$ such that $1 \in A$, $\xi r_{\delta(A)} \xi_1^{-1}$ would map:

$$\mathcal{P}_{n,1} \xrightarrow{\xi^{-1}} CUT_n^\square \xrightarrow{r_{\delta(A)}} CUT_n^\square \xrightarrow{\xi_1} \mathcal{P}_{n,1}$$

Using the bijections ξ_1 and ξ_2 we can write out the switching mappings for $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$ but a simpler way to view the switching operation is in terms of the set of the vertices on each side of the cut.

For $\mathcal{P}_{n,1}$ the switching mapping $\phi_{1,A}$ on a directed cut is given by:

$$\begin{aligned} x'_{i1} &= x_{i1} && (4.63) \\ x'_{1i} &= \begin{cases} 1 - x_{1i} & if\ i \notin A \\ x_{1i} & otherwise \end{cases} && (4.64) \\ x'_{ij} &= \begin{cases} 1 - x_{ij} - x_{1i} & if\ ij \in \delta^+(A) \\ x_{1j} - x_{ij} & if\ i \notin A, j \in A \\ x_{ij} + x_{1i} - x_{1j} & if\ i, j \notin A \\ x_{ij} & if\ i, j \in A. \end{cases} && (4.65) \end{aligned}$$

Similarly, the switching mapping for $\mathcal{P}_{n,2}$ for a directed cut $\delta^+(A)$ where $1 \notin A$ is defined by:

$$\begin{aligned} x'_{1i} &= x_{1i} && (4.66) \\ x'_{i1} &= \begin{cases} 1 - x_{i1} & if\ i \in A \\ x_{i1} & otherwise \end{cases} && (4.67) \\ x'_{ij} &= \begin{cases} 1 - x_{ij} - x_{j1} & if\ (i, j) \in \delta^+(A) \\ x_{i1} - x_{ij} & if\ i \notin A, j \in A \\ x_{ij} + x_{j1} - x_{i1} & if\ i, j \in A \\ x_{ij} & if\ i, j \notin A. \end{cases} && (4.68) \end{aligned}$$

We know from the structure of $\mathrm{DCUT}_4^\square$ given in Section 4.5 that vertices of $\mathrm{DCUT}_n^\square$ can be very different. This switching for $\mathcal{P}_{n,1}$ and $\mathcal{P}_{n,2}$ can only map directed cut vectors to other directed cut vectors where node 1 does not change side in the node partition.

CHAPTER 5
The rooted cut polytope

As mentioned, the simplest facets of the cut polytope are those defined by the triangle inequalities:

$$
\begin{aligned}
x_{i,j} - x_{i,k} - x_{j,k} &\leq 0, \\
-x_{i,j} + x_{i,k} - x_{j,k} &\leq 0, \\
-x_{i,j} - x_{i,k} + x_{j,k} &\leq 0, \qquad (5.1) \\
x_{i,j} + x_{j,k} + x_{k,i} &\leq 2, \qquad (5.2)
\end{aligned}
$$

that define the metric polytope $\mathrm{MET}_n^\square$ which was introduced in Section 3.2.

If we instead only consider the triangle inequalities involving a single node 1 we get the following system of inequalities that define the rooted semimetric polytope $\mathrm{RMET}_n^\square$, previously introduced in Section 3.2.

$$
\begin{aligned}
x_{1,j} - x_{1,k} - x_{j,k} &\leq 0, \\
-x_{1,j} + x_{1,k} - x_{j,k} &\leq 0, \\
-x_{1,j} - x_{1,k} + x_{j,k} &\leq 0, \qquad (5.3) \\
x_{1,j} + x_{j,k} + x_{k,1} &\leq 2, \qquad (5.4)
\end{aligned}
$$

The following theorem was stated in [17] in the terms of a correlation setting:

Theorem 27 *For $c \in \mathbb{R}^{E(K_n)}$, $max\{c^T x : x \in RMET_n^\square\} = max\{c^T x : x \in MET_n^\square\}$ if and only if* $\max\{c^T x : x \in RMET_n^\square\}$ *has an integral optimal solution.*

One direction is straight forward, If $\max\{c^T x : x \in RMET_n^\square\}$ has an integral optimal solution then this optimal solution is a cut vector. This follows from Proposition 27.2.1 of [30] which states that the only integral vectors of both $\mathrm{RMET}_n^\square$ and $\mathrm{MET}_n^\square$ are the cut vectors and every cut vector is a vertex of both $\mathrm{RMET}_n^\square$ and $\mathrm{MET}_n^\square$. As $CUT_n^\square \subseteq MET_n^\square \subseteq RMET_n^\square$, Theorem 27 implies that this cut vector maximizes the objective function over the polytope $\mathrm{CUT}_n^\square$ and $\mathrm{MET}_n^\square$ as well and:

$$\max\{c^T x : x \in CUT_n^\square\} = \max\{c^T x : x \in MET_n^\square\} = \max\{c^T y : y \in RMET_n^\square\}.$$

This theorem therefore gives a certificate of the optimality of maximizing $c^T x$ over $\mathrm{CUT}_n^\square$. Such certificates are not expected to exist for all objective functions, since optimizing over $\mathrm{CUT}_n^\square$ is known to be NP-hard. In the alternative proof that we present, we will show how to find this optimum integer solution.

Our alternate proof of the other direction is in terms of a cut setting and the switch mapping. The first part of the proof uses the following two lemmas:

Lemma 28 *Let u be a vector (vertex) in $MET_n^\square$, then for any $i \in 1, ..., n$ there exists a vector (vertex) $v \in MET_n^\square$ that has $v_{1,k} = u_{1,k}$ with $i \neq k$ and $v_{1,i} = 1 - u_{1,i}$.*

Proof. We can obtain the point v via switching u with the cut $S = \{i\}$. ■

Lemma 29 *Every vector $x \in MET_n^\square$ can be expressed as a convex combination of vectors of $RMET_n^\square$ with at least one being integral.*

Proof. Let $U = \{i : x_{1,i} < \frac{1}{2}\}$. By Lemma 28 and reordering the labels if needed we can perform a series of $0 \leq |U| \leq n$ switches on x such that the following holds:

$$1 \geq x_{1,2} \geq x_{1,3} \geq ... \geq x_{1,n} \geq \frac{1}{2}$$

Let s be the index such that $x_{1,i} > \frac{1}{2}$ for $i \leq s$ and $x_{1,i} = \frac{1}{2}$ for $s < i \leq n$.

Break the set of indicies i, j, $i < j$ into two sets, $A = \{i, j : i \leq s\}$ and $B = \{i, j : i > s\}$. For $i, j \in A$,

$$x_{i,j} \quad < \quad x_{1,i} + x_{1,j} \tag{5.5}$$

since $x_{i,j} \leq 1$, $x_{1,i} > \frac{1}{2}$ and $x_{1,j} \geq \frac{1}{2}$. Combining (5.5) with $x_{i,j} + x_{1,i} + x_{1,j} \leq 2$ yields $x_{i,j} < 1$.

If $i, j \in B$ then $x_{i,j}$ either satisfies:

$$x_{i,j} \quad = \quad x_{1,i} + x_{1,j} = 1 \tag{5.6}$$

or

$$x_{i,j} \quad < \quad x_{1,i} + x_{1,j} = 1. \tag{5.7}$$

For the proof we want to avoid $x_{i,j} = 1$, case (5.6). Consider the following series of switches to eliminate equalities of type (5.6) from occurring.

for $k \leftarrow s, ..., n-1$ **do**
    Set $S_k = \{1, ..., k\}$
    for $j \leftarrow k+1, ..., n$ **do**
        if $x_{k,j} < 1$ **then**
            Set $S_k = S_k \cup \{j\}$.
        end if

    end for

    Switch on the set S_k.

end for

Let x' be the vector x at the end of the series of switches. The series of switches performed do not alter the values of the $x_{1,i}$ edges. If $i < s$ then every switch performed has nodes 1 and i in S_k and therefore leaves $x'_{1,i} = x_{1,i}$. If $i \geq s$ then $x_{1,i} = \frac{1}{2}$ and any switch with $i \notin S_k$ sets $x'_{1,i} = 1 - \frac{1}{2} = \frac{1}{2}$. If $i < s$ then $x'_{i,j} < 1$ as $x'_{1,i} > \frac{1}{2}$, $x'_{1,j} > \frac{1}{2}$ and $x'_{i,j} + x'_{1,i} + x'_{1,j} \leq 2$.

We now show that for $i \geq s$, $x'_{i,j} < 1$. Assume for a contradiction that there exists at least one $x'_{i,j} = 1$. Choose the maximum value of i such that there exists a $x'_{i,j} = 1$ and choose a value of j such that it is minimum with respect to this value of i. If $x_{i,j} = 1$ when $k = i$ in the switching algorithm, S_k would contain i and but not j and $x_{i,j}$ would become 0. It follows that an assignment $x'_{i,j} = 1$ occurs at a later stage than when $k = i$. Assume it occurs at the stage $k = l > i$. As S_l doesn't contain j it follows that $l < j$ and $x_{l,j} = 1$ before the switch S_l is performed. The switching operation preserves the integrality of $0, 1$ values which means that $x'_{l,j}$ is either 0 or 1 at the end of the algorithm.

Since x' satisfies the triangle inequalities $x'_{i,j} \leq x'_{i,l} + x'_{l,j}$. If $x'_{l,j} = 1$ we have a contradiction to our choice of i and thus $x'_{l,j} = 0$ and $x'_{i,l} = 1$. But this contradicts our choice of j as $l < j$. Therefore, $x'_{i,j} < 1$ for all $1 < i < j \leq n$.

Now x' can be expressed as the following convex combination:

$$x' = \epsilon z' + (1 - \epsilon) v'$$

where z' is the cut $\delta(\{1\})$ and v' is given by:

$$\begin{aligned} v'_{1,j} &= \frac{x'_{1,j} - \epsilon}{1 - \epsilon} && (5.8) \\ v'_{i,j} &= \frac{x'_{i,j}}{(1 - \epsilon)} && (5.9) \end{aligned}$$

It remains to show that v' is in $\mathrm{RMET}_n^\square$. Choose ϵ to be $\frac{1}{2}(1 - \max_{i,j:i<j,i\neq 1} x'_{i,j})$. This choice ensures that $\epsilon > 0$ and $v'_{i,j} \leq 1$ since $x'_{i,j} < 1$ for all edges i, j.

Checking the triangle inequalities for $\mathrm{RMET}_n^\square$, v' must satisfy $v'_{1,j} + v'_{1,i} + v'_{i,j} \leq 2$ which is equivalent to:

$$\frac{x'_{1,j} - \epsilon}{1 - \epsilon} + \frac{x'_{1,i} - \epsilon}{1 - \epsilon} + \frac{x'_{i,j}}{1 - \epsilon} \leq 2 \tag{5.10}$$

or:

$$x'_{1,j} + x'_{1,i} + x'_{i,j} \leq 2, \tag{5.11}$$

which is satisfied since x' is in $\mathrm{MET}_n^\square$. The value of v' must also satisfy:

$$v'_{1,i} \leq v'_{1,j} + v'_{i,j} \ (or\ v'_{1,j} \leq v'_{1,i} + v'_{i,j}). \tag{5.12}$$

This is equivalent to:

$$\frac{x'_{1,i} - \epsilon}{1 - \epsilon} \leq \frac{x'_{i,j}}{1 - \epsilon} + \frac{x'_{1,j} - \epsilon}{1 - \epsilon} \tag{5.13}$$

or:

$$x'_{1,i} \leq x'_{i,j} + x'_{1,j}, \tag{5.14}$$

which is satisfied by x'. The last type of inequality that must be satisfied has the form:

$$v'_{i,j} \leq v'_{1,i} + v'_{1,j} \tag{5.15}$$

or:

$$\frac{x'_{i,j}}{1-\epsilon} \quad \leq \quad \frac{x'_{1,i}-\epsilon}{1-\epsilon} + \frac{x'_{1,j}-\epsilon}{1-\epsilon}, \tag{5.16}$$

which is satisfied as $x'_{1,i} + x'_{1,j} \geq 1$ and our choice of ϵ ensures that $x'_{i,j} \leq 1 - 2\epsilon$.

Performing the series of switches on the vectors v' and z' in the reverse order with the sets S_k, $k = n-1, ..., s$ followed by the $|U|$ switches with the sets $\{i\}$ for all $i \in U$ yields the vectors v and z where $x = \epsilon z + (1-\epsilon)v$. ■

The proof of Theorem 27 is now fairly straight forward.

Proof. (Theorem 27) If $\max\{c^T x : x \in RMET_n^{\square}\}$ has an integral optimal solution then this optimal solution is a cut vector. This direction was proved earlier after the theorem statement.

To prove the other direction assume that $\max\{c^T x : x \in MET_n^{\square}\} = \max\{c^T y : y \in RMET_n^{\square}\}$. Let $x \in MET_n^{\square}$ maximize $c^T x$, by Lemma 29 the vector x can be expressed as a convex combination of the vectors z and v where z is a cut vector and $v \in RMET_n^{\square}$. Using Proposition 27.2.1 of [30] again implies that the cut vector z obtained by applying Lemma 29 is a vertex of $\mathrm{RMET}_n^{\square}$ and $\mathrm{MET}_n^{\square}$. Let $y \in RMET_n^{\square}$ maximize $c^T y$, by the statement of the theorem $c^T y = c^T x = c^T(\epsilon z + (1-\epsilon)v)$ where $z, v \in RMET_n^{\square}$. Since y maximized $c^T y$ it follows that $c^T y = c^T v = c^T z$. Therefore, given c if:

$$\max\{c^T x : x \in MET_n^\square\} \;=\; \max\{c^T y : y \in RMET_n^\square\}$$

there exists a cut vector z that is also optimal with respect to c. ■

The proof of Lemma 29 was constructive in the sense that following the series of switches outlined we can construct v and z. This gives the following corollary.

Corollary 30 *If* $\max_{x \in RMET_n} c^T x = \max_{x \in MET_n} c^T x$*, the cut vector* z *maximizing* $c^T z$ *can be found in polynomial time.*

The proof of Theorem 27 as presented does not immediately generalize to the directed case. If it could be shown that $\mathrm{DMET}_n^\square$ was the convex hull of two $\mathrm{MET}_n^\square$ polytopes and $\mathrm{RDMET}_n^\square$ was likewise the convex hull of two $\mathrm{RMET}_n^\square$ polytopes in the way that $\mathrm{DCUT}_n^\square$ can be expressed as the convex hull of the linear bijective mapping of two $\mathrm{CUT}_n^\square$ polytopes a similar result to Theorem 27 would be easy to show. However, this is not the case, in fact one needs to only look as far as $n = 4$ to see that $\mathrm{DMET}_n^\square$ and $\mathrm{RDMET}_n^\square$ are not the convex hulls of two instances of linear transformations of $\mathrm{MET}_n^\square$ and $\mathrm{RMET}_n^\square$ respectively.

$\mathrm{MET}_4^\square$ has 8 vertices which are precisely the 8 cut vectors on 4 vertices since for the graph K_4, $\mathrm{MET}_4^\square = \mathrm{CUT}_4^\square$. Computing the vertices of $\mathrm{DMET}_4^\square$ with the software package lrs [4] gives fractional values which imply that $\mathrm{DMET}_4^\square$ is a strict superset of $\mathrm{DCUT}_4^\square$. Lemma 13 implies that the 15 cut vectors of $\vec{K}_4$ are vertices of both $\mathrm{DCUT}_4^\square$ and $\mathrm{DMET}_4^\square$ which means $\mathrm{DMET}_4^\square$ has strictly more than 15 vertices, which is larger than one would have hoped for from the convex hull of two $\mathrm{MET}_4^\square$ polytopes intersecting only at the origin. Using lrs [4] to enumerate the set of vertices of

$\mathrm{DMET}_4^{\square}$ supported this observation; it outputted that $\mathrm{DMET}_4^{\square}$ had 21 vertices. We have included the enumeration of vertices of $\mathrm{DMET}_4^{\square}$ in Appendix 10. Enumerating the set of vertices for $\mathrm{RMET}_4^{\square}$ and $\mathrm{RDMET}_4^{\square}$ yielded similar results. $\mathrm{RMET}_4^{\square}$ was found to have 12 vertices, 8 cut vectors plus and additional 4 vectors with some fractional entries. $\mathrm{RDMET}_4^{\square}$ was found to have $35 > 2 * 12 - 1$ vertices, 15 integral directed cut vectors and an additional 20 fractional vectors.

CHAPTER 6
Projecting the directed cut polyhedra

Solving an optimization problem on the cut polytope is computationally difficult in general. Hence, the study of the metric and rooted metric polyhedra. They are natural LP relaxations that can often be solved efficiently, giving bounds on the optimum integer solution. Solving these relaxations also provides a good starting point for an integer program solver when attempting to solve optimization problem on the cut polytope in practice. In some special cases the solution obtained when solving the relaxed problem can be shown to always give the optimal solution, ie. the optimal solutions on the metric and cut polytope coincide.

Given a graph $G \subset K_n$ with n nodes, the following notation: $\mathrm{CUT}^{\square}(G)$, $\mathrm{MET}^{\square}(G)$ and $\mathrm{RMET}^{\square}(G)$ is used to refer to the projections of $\mathrm{CUT}^{\square}_n$, $\mathrm{MET}^{\square}_n$ and $\mathrm{RMET}^{\square}_n$ respectively onto $\mathbb{R}^{|E(G)|}$, the edge set of graph G.

While the Fourier-Motzkin elimination method can be used to obtain descriptions of the projected polyhedra from the polyhedra corresponding to the complete graph, it is computationally difficult to compute such a projection. Barahona and Mahjoub [12] have found an explicit linear description of the semimetric polyhedra based on the cycle inequalities, which are of the form:

$$\sum_{e \in F} x_e - \sum_{e \in C \setminus F} x_e \;\; \leq \;\; |F| - 1 \tag{6.1}$$

where C is a cycle of G and $F \subseteq C$ has an odd number of edges $|F|$. Their results lead to the following theorem which is listed as Theorem 27.3.3 in [30]:

Theorem 31 (Barahona [10], Barahona and Mahjoub [12]) *For a graph G,*

$$\begin{aligned} MET(G) &= \{x \in \mathbb{R}^E_+ : x_e - x(C \setminus \{e\}) \leq 0 \ for\ C\ cycle\ of\ G, e \in C\}, \\ MET^{\square}(G) &= \{x \in \mathbb{R}^E_+ : x_e \leq 1 \ for\ e \in E, \\ & \qquad x(F) - x(C \setminus F) \leq |F| - 1 \ for\ C\ a\ cycle\ of\ G, F \subseteq C, |F|\ odd\}. \end{aligned}$$

- *For C a cycle of G, $e \in C$ and $F \subseteq C$ with $|F|$ odd, $x_e - x(C \setminus \{e\}) \leq 0$ is a facet of MET(G) if and only if C is a cordless circuit.*
- *The inequality $x_e \geq 0$ defines a facet of MET(G) if and only if e does not belong to a triangle.*

Furthermore, if G does not contain a K_5-minor then:

$$MET(G) = CUT(G)\ and\ MET^{\square}(G) = CUT(G).$$

The result for cones is due to Seymour [75], while the extension to polytopes is due to Barahona and Mahjoub [12] via switching. These characterizations imply a polynomial time algorithm for the max cut problem on graph without a K_5-minor. One could simply solve the optimization problem on the semimetric polytope for the complete graph where edges that do not appear in G have a weight of 0 in the objective function.

Alternatively, Barahona and Mahjoub presented a separation algorithm that can find a violated cycle inequality (6.1) in polynomial time. The algorithm begins by checking if any non-negativity constraint is violated. Then it constructs an auxiliary graph G' by taking two copies of G, say G_1 and G_2, so $V(G') = V(G_1) \cup V(G_2)$. Let i_1 denote the copy of vertex $i \in V(G)$ in graph G_1 and let i_2 be the copy of vertex $i \in V(G)$ in $V(G_2)$. If edge $(i,j) \in E(G)$ then $(i_1, j_1) \in E(G')$ and $(i_2, j_2) \in E(G')$. The graph G' will also contain edges of the form (i_1, j_2) and (i_2, j_1) if edge $(i,j) \in E(G)$.

Weights $x_{(i,j)}$ are assigned to edges (i_1, j_1) and (i_2, j_2) while weights $1 - x_{(i,j)}$ are assigned to edges of the form (i_1, j_2) and (i_2, j_1). The cycle inequality (6.1) can be rewritten in the form:

$$\sum_{ij \in C \setminus F} x_{ij} + \sum_{ij \in F} (1 - x_{ij}) \quad \geq \quad 1 \tag{6.2}$$

Now finding a violated inequality of the form (6.2) if one exists can be accomplished by finding the shortest path in G' between nodes i_1 and i_2.

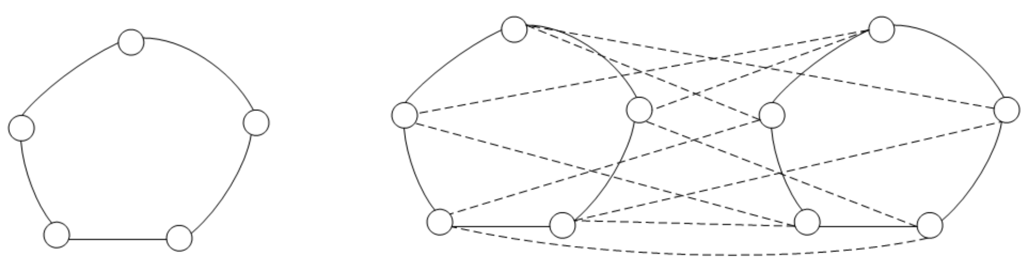


Figure 6–1: A graph G and its auxiliary graph G'. The dashed edges have weight $1 - x_e$

If there exists a path between i_1 and i_2 of length < 1, one can see that it corresponds to a violated inequality (6.2) where the edges (i_1, j_2) and (i_2, j_1) in the path form the set F and the other edges in the path form $C \setminus F$. As i_1 and i_2 are on opposite sides of the vertex sets $V(G_1)$ and $V(G_2)$ it follows that $|F|$ must be odd and it is straight forward to see that such a C and F must correspond to a violated cycle inequality.

If x violates a cycle inequality (6.2) and C' is the cycle and F' the odd subset of edges of C' in the violated inequality let i be a node on the cycle C'. Let $f_1, ..., f_{|F'|}$ be the edges in F in the order they appear when traversing the cycle C' starting at node i, let $C'_1 \subseteq C' \setminus F'$ be the set of edges appear between edges f_i and f_{i+1} where i is even and let $C'_2 \subseteq C' \setminus F'$ be the set of edges appearing between f_i and f_{i+1} when i is odd. The path in G' from i_1 to i_2 that is made up of the edges $B = \{j_1, k_2 : f_i = j, k \textit{ and } i \textit{ is odd}\} \cup \{j_2, k_1 : f_i = j, k \textit{ and } i \textit{ is even}\}$ along with the edges $\{i_1, j_1 : i, j \in C'_1\} \cup \{i_2, j_2 : i, j \in C'_2\}$ form a path from i_1 to i_2 in G' where the length of the path is equal to the violated inequality (6.2).

As the shortest path algorithm can be run in $O(n^2)$ time, the whole separation algorithm takes $O(n^3)$ as the shortest path algorithm must be run for each vertex $i \in V(G)$. Using this separation algorithm one can optimize in polynomial time using the ellipsoid method and the analysis of Grötschel, Lovász and Schrijver [44].

6.1 Projected facets of the directed cut polytope

For a directed graph $G \subset \vec{J}_n$ with n nodes we can define: $\mathrm{DCUT}^{\square}(G)$, $\mathrm{DMET}^{\square}(G)$ and $\mathrm{RDMET}^{\square}(G)$ to be the projections of DCUT_n, DMET_n and RDMET_n, respectively, onto the subspace $\mathbb{R}^{|A(G)|}$ indexed by the arcs of G. We will limit our focus to subgraphs of $\vec{J}_n$, using the linearities (3.27) these results can be extended to graphs that are subgraphs of $\vec{K}_n$ but not $\vec{J}_n$.

In investigating the projection of the directed cut polyhedra and relaxations we begin by considering some simple inequalities and give necessary and sufficient conditions for when they are facet inducing.

We begin by considering a generalization of the triangle inequality $x_{ij} \leq x_{ik}+x_{kj}$, a facet of DCUT_n. Assume there exists a path from i to j with that $i < j$ and by relabeling the arcs if needed $P_{ij} = \{(i, i+1), (i+1, i+2), ..., (j-1, j)\}$. We define path P_{ij} to be *induced* in G if $A(G)$ does not contain any arc $kl \notin P_{ij}$ such that $k < l$ and nodes k and l are in path P_{ij} and $kl \neq ij$.

Lemma 32 *Let G be a directed graph, P_{ij} be a directed path from i to j in G and let ij be an arc in G. The inequality:*

$$x_{ij} \leq \sum_{a \in P_{ij}} x_a \tag{6.3}$$

is facet inducing for DCUT(G) if and only if P_{ij} is an induced path.

To prove Lemma 32 we will use a technique referred to in the undirected cut literature as triangular elimination which was first proposed by Avis, Imai, Ito and Sasaki [5] and further refined in [6]. Triangular elimination is a combination of zero-lifting and Fourier-Motzkin elimination that can be used to map facet inducing inequalities for $\text{CUT}^{\square}(G)$ to facet inducing inequalities $\text{CUT}^{\square}(G')$ for graphs G and G' satisfying certain conditions.

Let $G = (V, E)$ be a graph and let u, v be and edge of E. The graph $G' = (V', E')$ is a *graph triangular elimination* of G (with respect to u, v) if $V' = V \cup \{w\}$, $w, u \in E'$ $w, v \in E'$ and $E' \cap E = E \setminus \{u, v\}$ where $w \notin V$.

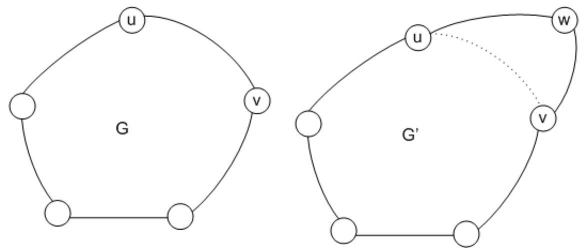


Figure 6–2: A graph G and G', its triangular elimination with respect to edge u, v.

Let G' be a triangular elimination of $G = (V, E)$. Then the inequality $(a')^T x \leq 0$ is an *inequality triangular elimination* of $a^T x \leq 0$ with respect to the edge u, v if:

$$a'^T x \;=\; a^T x - a_{u,v} x_{u,v} + a_{u,v} x_{v,w} - a_{u,v} x_{u,w}.$$

The following theorem appears as Proposition 4 in [6].

Proposition 33 (Proposition 4 of [6]) *Let $G' = (V', E')$ be a graph triangular elimination of $G = (V, E)$, and let $a'^T x \leq 0$ be the inequality triangular elimination of $a^T x \leq 0$. Then $a'^T x \leq 0$ is facet inducing for CUT(G') if $a^T x \leq 0$ is facet inducing for CUT(G) and $a^T x \leq 0$ is not completely supported by u, v.*

More complex versions of triangular elimination exist in [6], however their proofs often use the switching operation which is not available in the directed cut framework. We will extend the notion of triangular elimination to the directed cut polyhedra, giving sufficient conditions on G and G' so that a set of facet inducing inequalities for DCUT(G) can be mapped to facet inducing inequalities for DCUT(G'). Analogous results to Proposition 4 of [6] will be presented. These results will allow us to characterize classes of facet inducing inequalities for the projected directed cut polyhedra.

For our purpose, we will define multiple types of directed triangular elimination of a graph and inequality. To prove Lemma 32, we will prove a form of triangular elimination that takes a facet defining inequality and adds a multiple of the inequality:

$$x_{uw} \quad \leq \quad x_{uv} + x_{vw} \tag{6.4}$$

to produce a new inequality that is facet defining under certain criteria.

In Figure 6–3, we depict an example where an arc uv is eliminated from an inequality by adding inequality (6.4).

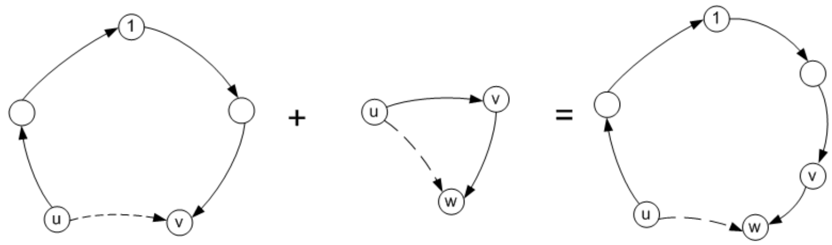


Figure 6–3: A depiction of the triangular elimination of an inequality of type (6.3).

We now state and prove a directed triangle elimination lemma for a multiple of inequality (6.4).

Lemma 34 *Let $G = (V, A)$ be a directed graph and $G' = (V', A')$ be the directed graph with nodes $V' = V \cup \{w\}$ and arcs $A' = (A \setminus \{uv\}) \cup \{wu, wv\}$. If $a^T x \leq 0$ is a facet inducing inequality for DCUT(G) with $a_{uv} < 0$ then*

$$a'^T x \quad = \quad a^T x - a_{uv} x_{uv} + a_{uv} x_{wv} - a_{uv} x_{wu} \leq 0$$

is a facet inducing inequality for DCUT(G').

Proof. Let $S_1, ..., S_{|A|-1}$ be subsets of $V(G)$ such that $\delta^+_G(S_1), ..., \delta^+_G(S_{|A|-1})$ are linearly independent roots of $a^T x \leq 0$.

Let

$$S'_i = \begin{cases} S_i \cup \{w\} & if\ u \in S_i \\ S_i & otherwise. \end{cases}$$

As G is the collapsing of $\pi_{uw}(G')$, the sets $\delta^+_{G'}(S'_1), ..., \delta^+_{G'}(S'_{|A|-1})$ are linearly independent and roots of the equation $a'^T x \leq 0$.

The inequality $a'^T x \leq 0$ is a facet of DCUT(G') if there exists a set of $|A'| - 1 = |A|$ linearly independent roots. The directed cut vectors $\delta^+_{G'}(S'_i), i = 1, ..., |A| - 1$ and the cut vector $\delta^+_{G'}(\{w\})$ are linearly independent as the arc wu only appears in the cut vector $\delta^+_{G'}(\{w\})$. The cut vector $\delta^+_{G'}(\{w\})$ is a root of $a'^T x \leq 0$ as the LHS of the inequality becomes $a_{uv}x_{wv} - a_{uv}x_{wu} = a_{uv} - a_{uv} = 0$.

Therefore, the cut vectors $(\cup_{i=1}^{|A|-1} \delta^+_{G'}(S_i)) \cup \delta^+_{G'}(\{w\})$ form a set of $|A|$ linearly independent roots of $a'^T x \leq 0$. ■

We can now use Lemma 34 to prove that the class of inequalities presented in Lemma 32 are facet inducing.

Proof of Lemma 32. We will prove this by induction on the number of arcs in the path P_{ij}. For the base case of two arcs Corollary 21 states that in the full dimensional complete directed graph, $\vec{J}_n$, $x_{ij} \leq x_{(i,i+1)} + x_{(i+1,j)}$ is a facet.

By relabeling the vertices if needed assume that $i < j$ and $kl \in P_{ij}$ implies $k = l - 1$. Assume P_{ij} is not an induced path, ie. $\exists$ an arc kl such that $l > k + 1$ and nodes l and k appear on path P_{ij} and let P'_{ij} be the path from i to j that uses arcs from P_{ij} between i and k, then arc kl and finally arcs from P_{ij} from l to j. Let P'_{kl} be the arcs in P_{ij} from k to l.

Then the inequalities:

$$x_{kl} \leq \sum_{a \in P'_{kl}} x_a \tag{6.5}$$

$$x_{ij} \leq \sum_{a \in P'_{ij} \setminus P_{kl}} x_a \tag{6.6}$$

are either facet defining if the the paths are induced, by the induction hypothesis or they are the sum of facet inducing inequalities.

The sum of equations (6.5) and (6.6) is:

$$x_{ij} \leq \sum_{a \in P_{ij}} x_a$$

which implies that it is not a facet.

Now assume that P_{ij} is an induced path. Consider the graph $G' = (V', A')$ where $V' = V \setminus \{i\}$ and $E' = (E \setminus \{ij, (i, i+1)\}) \cup \{(i+1, j)\}$. By the induction hypothesis the inequality:

$$x_{(i+1,j)} \leq \sum_{a \in P_{ij} \setminus \{(i,i+1)\}} x_a \tag{6.7}$$

is a facet inducing inequality for DCUT(G'). We can now apply Lemma 34 to inequality (6.7) using $x_{ij} \leq x_{(i,i+1)} + x_{(i+1,j)}$ for the triangular elimination. The resulting inequality:

$$x_{(i+1,j)} \leq \sum_{a \in P_{ij} \setminus \{(i,i+1)\}} x_a - x_{ij} + x_{(i,i+1)} + x_{(i+1,j)} \tag{6.8}$$

is a facet in DCUT(G) as required. ■

The second family of inequalities that we are interested in proving are facets of the projected directed cut polyhedra are based on the non-negativity inequalities. We know the inequality $x_a \geq 0$ is a facet inducing inequality for DCUT$_n$ by Corollary 21. It follows that $x_a \geq 0$ is a facet for DCUT(G) for any directed graph G containing arc a. When the directed cut cone for the complete directed graph $\vec{K}_n$ was projected to the full dimensional directed cut cone for DCUT($\vec{J}_n$) we used the linearities $x_{ji} = x_{ij}+x_{j1}+x_{1i}-x_{i1}-x_{1j}$ to eliminate entries corresponding to arc ji for all $i < j$. Since the choice of labelling vertices is arbitrary for $\vec{K}_n$ due to symmetry, the inequality $x_{ji} \geq 0$ is a facet for DCUT$_n$ projected onto the space indexed by the arcs of a graph containing arc ji, which implies that:

$$x_{ij} + x_{j1} + x_{1i} \quad \geq \quad x_{i1} + x_{1j} \tag{6.9}$$

is a facet of DCUT(G) for a graph G not containing arc ji.

The inequality (6.9) can be generalize to:

$$x_{i1} + x_{1j} \quad \leq \quad \sum_{a \in C} x_a \tag{6.10}$$

where C is a directed cycle in G containing nodes i, j and 1. To prove (6.10) is a facet inducing inequality when the cycle C contains no shorter cycle, we will use another form of directed triangular elimination. This form is based on using a multiple of an inequality of the form:

$$x_{uv} \quad \leq \quad x_{uw} + x_{wv} \tag{6.11}$$

to eliminate arc uv. Figure 6–4 depicts such an elimination.

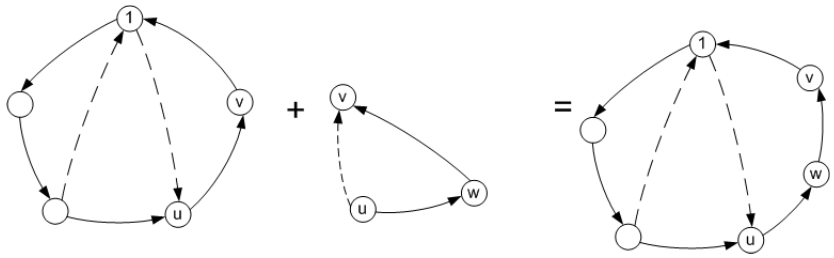


Figure 6–4: A depiction of our second type of triangular elimination using a multiple of inequality (6.11).

We now state our second type of triangular elimination formally.

Lemma 35 *Let $G = (V, A)$ be a directed graph and let $a^T x \leq 0$ be a facet inducing inequality for DCUT(G) containing the term $a_{uv}x_{uv}$ where $a_{uv} < 0$ and there exists at least one root of $a^T x \leq 0$ that has a non-zero x_{uv}. Let $G' = (V', A')$ be a directed graph with nodes $V' = V \cup \{w\}$ and arcs $A' = A \cup \{uw, wv\} \setminus \{uv\}$. Then the inequality $a'^T x' = a^T x - a_{uv}x_{uv} + a_{uv}x_{uw} + a_{uv}x_{wv} \leq 0$ is a facet inducing inequality for DCUT(G').*

Proof. Let $\delta^+_G(S_1), ..., \delta^+_G(S_{|A|-1})$ be a set of $|A| - 1$ linearly independent roots of $a^T x \leq 0$. Let

$$S'_i = \begin{cases} S_i \cup \{w\} & if\ u \in S_i \\ S_i & otherwise. \end{cases}$$

We claim that the set of cut vectors $\delta^+_{G'}(S'_i)$, $i = 1, ..., |A| - 1$ are linearly independent roots of $a'^T x' \leq 0$.

Let M be the matrix created by using the vectors $\delta^+_G(S_i)$ as rows and let M' be the matrix formed by using vectors $\delta^+_{G'}(S'_i)$ as rows. If we index the columns of M

and M' by their corresponding head and tail nodes we see that the columns ij in M and M' are identical for $ij \notin \{uv, uw, wv\}$. The columns M_{uv} and M'_{vw} are identical by construction. Therefore $rank(M) = rank(M')$ and $\delta^+_{G'}(S'_i)$ for $i = 1, ..., |A| - 1$ are linearly independent. The vectors $\delta^+_{G'}(S'_i)$ are roots of $a'^T x' \leq 0$ as $\delta^+_{G'}(S'_i)_{uw} = 0$ by construction and if $\delta^+_G(S_i)_{uv} = 1$ then $\delta^+_{G'}(S'_i)_{wv} = 1$ and if $\delta^+_G(S_i)_{uv} = 0$ then $\delta^+_{G'}(S'_i)_{wv} = 0$ so $a'^T \delta^+_{G'}(S'_i) = a^T \delta^+_G(S_i) = 0$.

If the column M_{uv} is all zeros, we can consider a new root with non-zero x_{uv} (which we know exists by the lemma statement) and replace any row of M by this vector. We can now assume that M contains at least one cut vector with $x_{uv} = 1$, by relabeling if needed let S_1 be a set such that $\delta^+_G(S_1)$ has $u \in S_1$ and $v \notin S_1$.

Let $S'_{|A|} = S_1$ and recall that $S'_1 = S_1 \cup \{w\}$. The vector $\delta^+_{G'}(S'_{|A|})$ is a root of $a'^T x \leq 0$ as $\delta^+_{G'}(S'_{|A|})_{uw} = \delta^+_G(S_1)_{uv} = 1$, $\delta^+_{G'}(S'_{|A|})_{wv} = 0$, $a'_{uv} = 0$, and $a'_{uw} = a_{uv}$. Now append $\delta^+_{G'}(S'_{|A|})$ to the end of M'. We claim that the rows of M' are linearly independent and we know that there are $|A|$ rows. We know that the rows of M' without the last row are linearly independent and by construction the entries of column uw are all zero, since the last row of M' has a one in column uw it follows that the rows of M' are linearly independent. ∎

Using Lemma 35 we can generalize inequalities of the form (6.9) to get:

Lemma 36 *If $C = \{12, 23, ..., (h-1, h), h1\}$ is a directed cycle in $G' = (V', A')$ and i and j are two nodes in C', $i < j$ for which $i1, 1j \in A$ then the inequality:*

$$x_{i1} + x_{1j} \leq \sum_{a \in C'} x_a \tag{6.12}$$

is a facet inducing inequality of $DCUT(G')$ if and only if G' doesn't contain any arcs $kl \notin C'$ such that $k, l \in C'$, $kl \notin C'$, $k < l$ and either: $1 \leq k < l \leq i$, $i \leq k < l \leq j$ or $j \leq k < l \leq h$ and G' doesn't contain an arc $k1$, $j < k < h$.

Proof. Assume G' contains an arc kl such that either $1 \leq k < l \leq i$, $i \leq k < l \leq j$ or $j \leq k < l \leq h$ and let P_{kl} be the path from k to l in C'. The inequality

$$x_{kl} \leq \sum_{a \in P_{kl}} x_a \tag{6.13}$$

is valid for DCUT(G'). Let C be the directed cycle $C' \setminus \{P_{ij}\} \cup \{kl\}$ in G', the inequality

$$x_{i1} + x_{1j} \leq \sum_{a \in C} x_a \tag{6.14}$$

is also valid for DCUT(G'). If we sum inequalities (6.13) and (6.14) we get inequality (6.10). Hence, it is not facet inducing when such an arc kl exists. Using a similar approach one can show that if an arc $k1$, $j < k < h$ exists then (6.10) is not a facet.

If no such arcs exist and C' is an induced directed cycle we can proceed by induction on the length of C', since we know that $x_{ij} + x_{j1} + x_{1i} \leq x_{1j} + x_{i1}$ is a facet for DCUT$_n$ by Corollary 21 for our base case, a cycle of length 3.

Let w be a node of C' such that $w \notin \{1, i, j\}$. Consider a graph $G = (V, A)$ such that $V = V' \setminus \{w\}$ and $A = (A' \setminus \{(w-1, w), (w, w+1)\}) \cup \{(w-1, w+1)\}$. Let C be the directed cycle in G such that $C = (C' \setminus \{(w-1, w), (w, w+1)\}) \cup \{(w-1, w+1)\}$. By the induction hypothesis we know that:

$$x_{i1} + x_{1j} \leq \sum_{a \in C} x_a \tag{6.15}$$

is a facet of DCUT(G). This implies that:

$$x_{i1} + x_{1j} \;\; \leq \;\; \sum_{a \in C'} x_a$$

is a facet of DCUT(G') by applying Lemma 35 with $u = w - 1$ and $v = w + 1$ as $\delta^+_G(\{i, w-1\})$ is a root of 6.15 with $x_{(w-1,w+1)} = 1$. ■

Inequalities of the type (6.3) and (6.10) make up the building blocks for the facets of the projection of DMET$_n$ to the space indexed by the arcs of arbitrary directed graphs. We will discuss the projection of DMET$_n$ in further detail in Chapter 7.

We will introduce a few more triangular elimination techniques for graph where instead of using the triangle inequality $x_{ij} + x_{jk} \leq x_{ik}$ we will use the inequality:

$$x_{ij} + x_{j1} + x_{1i} \;\; \geq \;\; x_{1j} + x_{i1}. \tag{6.16}$$

However, we can not simply try to add this inequality to eliminate an arc $1i$ or $j1$. For example, consider adding inequality (6.16) to a facet inducing inequality:

$$x_{1u} + x_{ui} + x_{iw} + x_{w1} \;\; \geq \;\; x_{u1} + x_{1i} \tag{6.17}$$

for the graph $G = (V, A)$. The resulting inequality for the graph G' where $V(G') = \{1, u, w, i, j\}$ and $A(G') = \{1u, ui, ij, j1, iw, w1, i1, 1j, u1\}$ is:

$$x_{1u} + x_{ui} + x_{ij} + x_{j1} + x_{iw} + x_{w1} \;\; \geq \;\; x_{u1} + x_{i1} + x_{1j}. \tag{6.18}$$

However, (6.18) is not a facet of DCUT(G') as it is the sum of valid inequalities: $x_{i1} \leq x_{iw} + x_{w1}$ and $x_{u1} + x_{1j} \leq x_{1u} + x_{ui} + x_{ij} + x_{j1}$.

Instead of using inequality (6.16) for these further eliminations we use either:

$$x_{w1} + x_{1u} \leq x_{1w} + x_{wu} + x_{ui} + x_{i1} \tag{6.19}$$

or

$$x_{u1} + x_{1w} \leq x_{1j} + x_{ju} + x_{uw} + x_{w1}. \tag{6.20}$$

Lemma 37 *Let $G = (V, A)$ be a directed graph with arc $i1 \in A$ and let $a^T x \leq 0$ be a facet inducing inequality with $a_{i1} > 0$ for DCUT(G). Let $G' = (V', A')$ be a directed graph with nodes $V' = V \cup \{u, w\}$ and arcs $A' = (A \setminus \{i1\}) \cup \{1w, w1, 1u, wu, ui\}$. Then the inequality:*

$$\begin{aligned} a'^T x &= a^T x + a_{i1}(x_{w1} + x_{1u} - x_{1w} - x_{wu} - x_{ui} - x_{i1}) \\ &\leq 0 \end{aligned}$$

is facet inducing for DCUT(G').

Proof. The proof has a similar flavour to the proofs of Lemmas 34 and 35. Let S_j for $j = 1, ..., |A| - 1$ be subsets of V such that the vectors $\delta_G^+(S_j)$ are linearly independent and roots of $a^T x \leq 0$.

Let

$$S'_j = \begin{cases} S_j \cup \{w\} & if\ 1 \in S_j \text{ and } i \notin S_j \text{ or } if\ 1, i \notin S_j \\ S_j \cup \{u, w\} & if\ i \in S_j \text{ and } 1 \notin S_j \text{ or } if\ 1, i \in S_j. \end{cases}$$

By construction the directed cut vectors $\delta_{G'}^+(S'_j)$ are linearly independent roots of $a'^T x \leq 0$. As $|A'| = |A| + 4$, four additional roots are needed that are linearly independent together with the roots already defined to prove that $a'^T x \leq 0$ is facet defining.

Let $T_1 = \{w\}$, $T_2 = \{u, w\}$, $T_3 = V' \setminus \{u\}$ and let $T_4 = V' \setminus \{u, w\}$. It is easy to check that the vectors $\delta^+_{G'}(T_i)$ are roots of $a'^T x \leq 0$ for $i = 1, ..., 4$.

Let N be the $|A| - 1 \times |A|$ matrix constructed by using the vectors $\delta^+_G(S_j)$ as rows. By renumbering the sets S_j if needed assume the rows are ordered by the sets S_j such that $1, i \notin S_j$ followed by the sets S_j such that $i, 1 \in S_j$ followed by S_j such that $i \in S_j$ and $1 \notin S_j$ and lastly by S_j such that $1 \in S_j$ and $i \notin S_j$. By our choice of S_j's N is linearly independent. Let $i1$ be the arc the final column of matrix N corresponds to and let N' be matrix N without column $i1$.

Let M' be the $|A| + 3 \times |A| + 4$ matrix constructed by using the vectors $\delta^+_{G'}(S'_j)$ for $j = 1, ..., |A| - 1$ followed by $\delta^+_{G'}(T_k)$ for $k = 1, ..., 4$ as rows. If M' has full row rank then $a'^T x \leq 0$ is facet inducing as each row is a root of $a'^T x \leq 0$.

Let N_1 be the subset of rows of N' with $1 \in S_j$ and $i \notin S_j$. N_2 be the subset of rows of N' with $i \in S_j$ and $1 \notin S_j$. N_3 be the subset of rows of N' with $1, i \in S_j$. N_4 be the subset of rows of N' with $1, i \notin S_j$. Now M' has the form:

$$M' = \begin{array}{cccccc} A \setminus \{i1\} & w1 & 1w & 1u & wu & ui \\ \end{array}$$

$$M' = \left(\begin{array}{cccccc} N_1 & 0 & 1 & 1 & 0 & 0 \\ N_2 & 1 & 0 & 0 & 0 & 0 \\ N_3 & 0 & 0 & 0 & 0 & 0 \\ N_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{array} \right)$$

By construction, the matrix N appears as the upper left submatrix of M':

$$N = \begin{array}{cc} A \setminus \{i1\} & w1 \end{array}$$

$$N = \left(\begin{array}{cc} N_1 & 0 \\ N_2 & 1 \\ N_3 & 0 \\ N_4 & 0 \end{array} \right)$$

We can perform linearly reversible matrix operations to M' to show that it is linearly independent. As the last row of M' has all zeros in the columns corresponding to N we can subtract the last row of M' from each row of M' that corresponds to N_1. Let M'' be the resulting matrix.

Let

$$X = \begin{array}{c} \begin{array}{cccc} 1w & 1u & wu & ui \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

The matrix X has full row rank and now our matrix M'' has the form:

$$M'' = \begin{pmatrix} M & 0 \\ Z & X \end{pmatrix},$$

where M is a linear transformation of N. As M'' is full rank we know that $a'^T x \leq 0$ is a facet defining inequality for DCUT(G'). ■

Next we prove a similar lemma by using the inequality:

$$x_{u1} + x_{1w} \;\; \leq \;\; x_{1j} + x_{ju} + x_{uw} + x_{w1}. \tag{6.21}$$

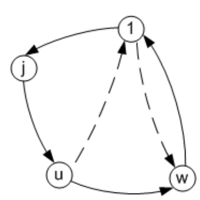


Figure 6–5: A depiction of the elimination inequality used in Lemma 38.

Lemma 38 *Let $G = (V, A)$ be a directed graph such that $1j \in A$ and $a^T x \leq 0$ is a facet inducing inequality for DCUT(G) where $a_{1j} > 0$. Let $G' = (V', A')$ be a directed* *graph with* $V' = V \cup \{u, w\}$ *and let* $A' = A \setminus \{1j\} \cup \{ju, uw, w1, 1w, u1\}$. *Then the inequality*

$$\begin{aligned} a'^T x &= a^T x + a_{1j}(x_{u1} + x_{1w} - x_{1j} - x_{ju} - x_{uw} - x_{w1}) && (6.22) \\ &\leq 0 && (6.23) \end{aligned}$$

is a facet inducing inequality for $DCUT(G')$.

Proof. Let $S_1, ..., S_{|A|-1}$ be a set of subsets of V such that $\delta^+_G(S_i)$ are linearly independent and roots of $a^T x \leq 0$.

Let

$$S'_i = \begin{cases} S_i & if\ 1 \in S_i\ and\ j \notin S_i\ or\ if\ 1, j \notin S_i \\ S_i \cup \{u, w\} & if\ 1, j \in S_i \\ S_i \cup \{u\} & if\ j \in S_i\ and\ 1 \notin S_i. \end{cases}$$

By construction, the vectors $\delta^+(S'_i)$ form $|A| - 1$ linearly independent roots of $a'^T x \leq 0$. An additional 4 more linearly independent roots are needed to prove that $a'^T x \leq 0$ is a facet of DCUT(G').

Let $T_1 = V(G') \setminus \{u, w\}$, $T_2 = \{u, w\}$, $T_3 = V(G') \setminus \{w\}$ and $T_4 = \{u\}$. One can easily check that the vectors $\delta^+_{G'}(T_i)$ are roots of $a'^T x \leq 0$ for $i = 1, ..., 4$.

We claim that the vectors $\delta^+_{G'}(T_k)$ for $k = 1, ..., 4$ with the $\delta^+_{G'}(S'_i)$ for $i = 1, ..., |A| - 1$ form the required linearly independent roots to prove $a'^T x \leq 0$ is a facet of DCUT(G'). As we did in the proof of Lemma 37 we will construct a matrix from these vectors to show that they are linearly independent. It is easy to check that they are roots. Let M' be the matrix formed by using the vectors $\delta^+_{G'}(S'_i)$ as rows where $1 \in S'_i$ and $j \notin S'_i$, let N_1 be these vectors restricted to the arcs $A \setminus \{1j\}$. Let the next set of rows of M' be the vectors $\delta^+G'(S'_i)$ where $j \in S'_i$ and $1 \notin S'_i$ and let N_2 be these vectors restricted to the arcs $A \setminus \{1j\}$. Let these vectors be followed by $\delta^+_{G'}(S'_i)$ where $1, j \in S'_i$ and let N_3 be these vectors restricted to the arcs $A \setminus \{1j\}$. Let M' next contain the vectors $\delta^+_{G'}(S'_i)$ where $1, j \notin S'_i$ as rows and let N_4 be these vectors restricted to the arcs $A \setminus \{1j\}$. Let the last four vectors of M' be $\delta^+_{G'}(T_k)$ for $k = 1, ..., 4$. The matrix M' has the form:

$$
M' = \begin{array}{c} \begin{array}{cccccc} A \setminus \{1j\} & 1w & ju & uw & w1 & u1 \end{array} \\ \begin{pmatrix} N_1 & 1 & 0 & 0 & 0 & 0 \\ N_2 & 0 & 0 & 1 & 0 & 1 \\ N_3 & 0 & 0 & 0 & 0 & 0 \\ N_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}
$$

By construction, the upper left section of matrix M' is:

$$
N = \begin{array}{c} \begin{array}{cc} A \setminus \{i1\} & w1 \end{array} \\ \begin{pmatrix} N_1 & 1 \\ N_2 & 0 \\ N_3 & 0 \\ N_4 & 0 \end{pmatrix} \end{array}
$$

where N is the matrix formed by using $\delta_G^+(S_i)$ for $i = 1, ..., |A| - 1$ as rows, hence full row rank. By using the last row of M' and subtracting it for each row of M' corresponding to an entry of N_2, M' takes the form:

$$M'' = \begin{pmatrix} N' & 0 \\ Z & X \end{pmatrix}$$

where N' is obtained from N by subtracting a row of N_4 from each row of N_2, which implies that N' is full row rank. The submatrix X is:

$$X = \begin{array}{c} \begin{array}{cccc} 1w & 1u & wu & ui \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

which has full row rank. This implies that the rows of M'' are linearly independent, hence the rows of M' are linearly independent and $a'^T x \leq 0$ is a facet of DCUT(G').

■

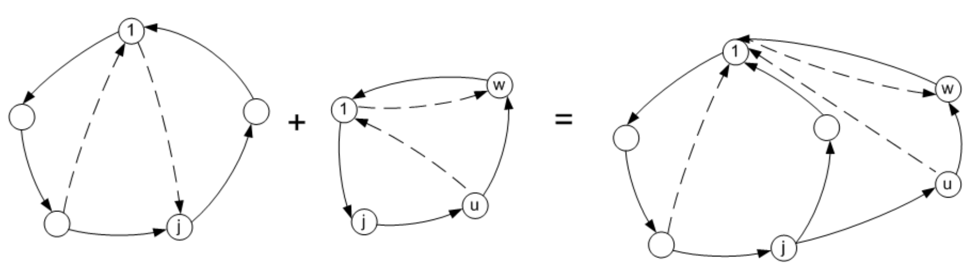


Figure 6–6: A depiction of the facet preserving elimination of Lemma 38.

With the operations of Lemmas 34, 35, 37 and 38, we can construct many different types of facet preserving inequalities from those already presented in Chapter 3. In Chapter 7 we discuss the construction of valid inequalities using DMET_n and triangular elimination further.

6.2 Zero lifting projected inequalities

It would be nice to have a zero lifting result for the projected polyhedra similar to the result of Theorem 25. For the projected cut polyhedra $\mathrm{CUT}(G)$ and $\mathrm{CUT}^{\square}(G)$, De Simone [24] proves a zero lifting theorem for graphs other than the complete graph under certain properties. Her theorem states:

Theorem 39 (De Simone [24]) *Let $G = (V, E)$ be a graph with $n \geq 3$ vertices, and let $H = (V \cup \{r\}, F)$ where F induced on the nodes V is E. If $N(r) - \{v\} \subseteq N(u)$ for some $u \in V(G)$ then:*

If the non-trivial inequality $a^T x \leq d$ defines a facet of $CUT^{\square}(G)$ then $[a, 0]^T x \leq d$ defines a facet of $CUT^{\square}(H)$.

An inequality is *non-trivial* if its support graph contains at least three vertices. The notation $[a, 0] \in \mathbb{R}^{|F|}$ denotes the vector a with zeros for entries that correspond to edges in $F \setminus E$.

We will prove a variation of De Simone's theorem for directed subgraphs of the full dimensional directed graph $\vec{J}_n$. Let $N^+_G(v)$ denote the set of nodes which are tail nodes of an arc directed towards v and let $N^-_G(v)$ denote the set of nodes which are head nodes of arcs directed from v in the directed graph G.

The proof of our theorem follows the same basic approach to that of De Simone's with some modifications. The proof of De Simone's theorem uses the fact that the support graph of all non-trivial facets for the cut polyhedra are two connected. This is not the case for the directed cut polyhedra. For instance in Section 4.5 it was stated that:

$$x_{ij} + x_{jk} + x_{kl} - x_{kj} \leq 1 \tag{6.24}$$

is facet inducing for $\mathrm{DCUT}_4^{\square}$, but the support graph is not two connected in the directed or undirected sense. However, we can avoid this problem by putting a requirement on the support graph of the facet inequality we are trying to lift.

For a directed graph $G = (V, A) \subseteq \vec{J}_n$ with $n \geq 3$ vertices, define H be a *r-copy* of G if:

- $V(H) = V(G) \cup \{r\}$.
- The graph induced by H on the vertices $V(G) \setminus \{r\}$ is G
- There exist $u \in V(G)$ such that $N_H^-(r) \setminus \{u\} \subseteq N_H^-(u)$ and $N_H^+(r) \setminus \{u\} \subseteq N_H^+(u)$.
- The vertex u has $|N_G^-(u)| \geq 2$ and $|N_G^+(u)| \geq 2$.

Theorem 40 *Let $G = (V, A) \subseteq \vec{J}_n$ be a directed graph with $n \geq 3$ vertices and let $H = (V \cup \{r\}, F)$ be an r-copy of G. If $a^T x \leq \alpha$ is a facet inducing non-trivial inequality of $DCUT^{\square}(G)$ where the underlying support graph $G(a)$ doesn't contain a node incident to every arc (not a directed star) then $[a, 0]^T x \leq \alpha$ is facet inducing for $DCUT^{\square}(H)$.*

Proof. We can assume that $\text{DCUT}^{\square}(G)$ is full dimensional as it is a subgraph of $\vec{J}_n$. Let m be the dimension of $\text{DCUT}^{\square}(G)$, which by assumption is equal to the number of arcs of G, and let $x^1, ..., x^m$ be affinely independent roots of $a^T x \leq \alpha$ that correspond to directed cuts $\delta_G^+(S_1), ..., \delta_G^+(S_m)$ where $S_i \subseteq V(G)$.

A well known property of facets (see [88] or [76]) states that if $\bar{a}$ is a vector such that $\bar{a}^T x^i = \bar{a}^T x^j$ for all $i, j \in \{1, ..., m\}$ then $\bar{a}$ must be a multiple of a.

Let y^k and z^k be the incidence vectors corresponding to cuts $\delta_H^+(S_k \cup \{r\})$ and $\delta_H^+(S_k)$ respectively. The new points satisfy $[a, 0]^T y^k = [a, 0]^T z^k = \alpha$. Let $c^T x = b$ be an arbitrary hyperplane through the $2m$ points y^k and z^k for $k = 1, ..., m$. Such a hyperplane exists as $c = [a, 0], b = \alpha$ is such a hyperplane. We want to show that $c = [\tilde{c}, \hat{c}] = \lambda[a, 0]$ and $b = \lambda\alpha$. Clearly, $\tilde{c}$ is a multiple of a as $\tilde{c}^T x^i = \tilde{c}^T x^j$ for all $i, j \in \{1, ..., m\}$.

Since $c^T(y^k - z^k) = ([\lambda a, \hat{c}])^T([x^k, \hat{y}^k] - [x^k, \hat{z}^k]) = b - b = 0$ for all k, it follows that:

$$\sum_{w \in S_k \cap N^-(r)} \hat{c}_{wr} = \sum_{w \in (V(H) \setminus S_k) \cap N^+(r)} \hat{c}_{rw} \tag{6.25}$$

Furthermore, $c^T(y^k + z^k) = 2(\lambda a)^T x^k + \hat{c}^T \hat{y}^k + \hat{c}^T \hat{z}^k = 2b$. This implies that:

$$\sum_{w \in S_k \cap N^-(r)} \hat{c}_{wr} + \sum_{w \in (V(H) \setminus S_k) \cap N^+(r)} \hat{c}_{rw} = 2b - 2\lambda\alpha. \tag{6.26}$$

Combining (6.25) and (6.26) we get that:

$$\begin{aligned} \sum_{w \in S_k \cap N^-(r)} \hat{c}_{wr} &= \sum_{w \in (V(H) \setminus S_k) \cap N^+(r)} \hat{c}_{rw} & (6.27) \\ &= b - \lambda\alpha. & (6.28) \end{aligned}$$

As we want to show that $\hat{c} = 0$ we begin by assuming that there exists at least one non-zero entry in $\hat{c}$ to obtain a contradiction. We want to have at least one arc other than ur and ru to have a non-zero $\hat{c}_{jr}$ or $\hat{c}_{rj}$ for some $j \in V(G) \setminus \{u\}$.

Assuming that $\hat{c}_{ur}$ is the only non-zero entry of $\hat{c}$, (6.25) implies that $u \in V(G) \setminus S_k$ for all k. Which implies that $\sum_{w \in S_k \cap N^-(r)} \hat{c}_{wr} = 0$ for all $k = 1, ..., m$ and $c^T y^k = c^T z^k = \alpha$.

As $|N_G^+(u)| \geq 2$, let i and j be two nodes such that $ui, uj \in A(G)$. The vectors $\delta_G^+(S_i)$ for $i = 1, ..., m$ are affinely independent which means the set of vectors $T = \{\delta_G^+(S_i) - \delta_G^+(S_m) : 1 \leq i \leq m-1\}$ is linearly independent. If we construct a matrix using the vectors of T as rows then the columns that correspond to iu and ju are all zeros. This contradicts T being a set of $m-1$ linearly independent vectors. Therefore, $\hat{c}_{ur}$ can not be the only non-zero entry of $\hat{c}$.

If we assume that $\hat{c}_{ru}$ is the only non-zero, equation (6.25) implies that $u \in S_k$ for $k = 1, ..., m$. As $|N_G^-(u)| \geq 2$, at least two nodes i and j exist such that $iu, ju \in A(G)$. If we consider the set T as constructed above, the columns corresponding to iu and ju will be all zeros which means that T is not linearly independent and the only non-zero entry can't be $\hat{c}_{ru}$.

By (6.25) there can not be exactly two non-zeros c_{ru} and c_{ur} since if $u \in S_k$ for a given k, the LHS of (6.25) is zero and the RHS is non-zero, likewise if $u \in V(G) \setminus S_k$ the RHS of (6.25) is zero and the LHS is non-zero. Therefore, at least one entry of $\hat{a}$ that is not a_{ru} or a_{ur} is non-zero.

Define a new vector $\bar{c} \in \mathbb{R}^{A(G)}$ where:

$$\begin{aligned} \bar{c}_{vu} &= \hat{c}_{vr} \qquad if\ v \in N^-(r) && (6.29)\\ \bar{c}_{uv} &= \hat{c}_{rv} \qquad if\ v \in N^+(r) && (6.30)\\ \bar{c}_{vw} &= 0 \qquad otherwise && (6.31) \end{aligned}$$

We will show that $\bar{c}$ is a multiple of a to obtain a contradiction on the structure of the support graph $G(a)$.

If $u \in S_k$ then by (6.28):

$$\begin{aligned} \sum_{w \in N^-(r)\setminus\{u\}} \hat{c}_{wr} x^k_{wu} + \sum_{w \in N^+(r)\setminus\{u\}} \hat{c}_{rw} x^k_{uw} &= \sum_{w \in (V(G)\setminus S_k)\cap N^+(r)} \hat{c}_{rw} && (6.32)\\ &= b - \lambda\alpha && (6.33) \end{aligned}$$

If $u \in V(G) \setminus S_k$ then:

$$\begin{aligned} \sum_{w \in N^-(r)\setminus\{u\}} \hat{c}_{wr} x^k_{wu} + \sum_{w \in N^+(r)\setminus\{u\}} \hat{c}_{rw} x^k_{uw} &= \sum_{w \in S_k \cap N^-(r)} \hat{c}_{wr} && (6.34)\\ &= b - \lambda\alpha && (6.35) \end{aligned}$$

Combining these equations we find that for all k:

$$\begin{aligned} \bar{c}^T x^k = \sum_{vw \in A(G)} \bar{c}_{vw} x^k_{vw} &= \sum_{w \in N^-(r)\setminus\{u\}} \hat{c}_{wr} x^k_{wu} + \sum_{w \in N^+(r)\setminus\{u\}} \hat{c}_{rw} x^k_{uw} && (6.36)\\ &= d - \lambda\alpha. && (6.37) \end{aligned}$$

This implies that $\bar{c} = \lambda' a$, ie. $\bar{c}$ is non-zero by construction and is a multiple of a. Hence, the vector a has zeros for entries corresponding to all arcs not containing node u which is a contradiction to the lemma assumptions. This implies that $\hat{c}$ must be zero. ■

Theorem 40 generalizes Theorem 25 for zero lifting directed cut polyhedra of the complete graph. If we know that $a^T x \leq \alpha$ is a facet of $\text{DCUT}_n^\square$ for $n \geq 4$ we can take node $n-1$ to be u and let node $n+1$ be the node r and apply Theorem 40, since the support graphs for non-trivial facet inducing inequalities for DCUT_n and $\text{DCUT}_n^\square$ don't have a single common node to every arc. Nodes 1 and $n-2$ have arcs directed toward $n-1$, node $n-1$ has arcs directed towards nodes n and 1, so $|N_{K_n}^+(n-1)| \geq 2$ and $|N_{K_n}^-(n-1)| \geq 2$.

Theorem 40 also deals with non-homogeneous facets. In particular we can now complete Section 4.5 by proving:

Lemma 41 *The inequality:*

$$x_{ki} + x_{ij} + x_{jl} - x_{ji} \quad \leq \quad 1 \tag{6.38}$$

is a facet inducing inequality for $DCUT_n^\square$ for $n \geq 4$.

Proof. For $\text{DCUT}_4^\square$ it is easy to check that the following cuts are roots of the inequality $x_{31} + x_{12} + x_{24} - x_{21} \leq 1$:

$$\delta^+(\{1\}), \delta^+(\{1,2\}), \delta^+(\{1,2,3\}), \delta^+(\{2,3\}), \delta^+(\{1,3\}),$$
$$\delta^+(\{3\}), \delta^+(\{1,3,4\}), \delta^+(\{3,4\}), \delta^+(\{1,4\}).$$

This proves that (6.38) is a facet of $\text{DCUT}_4^\square$ and as the support graph of (6.38) does not contain a node common to all arcs we can apply Theorem 40. ■

CHAPTER 7
Future work on the directed cut polyhedra

7.1 Directed metric polyhedra projections

As was mentioned at the start of Chapter 6, the projection of MET_n and $\mathrm{MET}_n^\square$ onto the arc set of a subgraph of K_n has a very nice characterization. The following is due to a result of Barahona:

Lemma 42 (Barahona [10])

$$MET(G) = \{x \in \mathbb{R}_+^{E(G)} | x_e - x(C \setminus \{e\}) \leq 0 \ for \ C \ a \ cycle \ of \ G, e \in C\}.$$

Using Proposition 11 on switching, Barahona and Mahjoub expressed the structure of $\mathrm{MET}^\square(G)$:

Lemma 43 (Barahona and Mahjoub [12]) $MET^\square(G) = \{x \in \mathbb{R}_+^E | x_e \leq 1 for\ e \in E, x(F) - x(C \setminus F) \leq |F : -1 \ for \ C \ a \ cycle \ of \ G, F \subseteq C, |F| \ odd\}$.

While this set of cycle inequalities used in this lemma can be exponential and the Fourier-Motzkin procedure can be used to describe the projection of a polyhedron using an exponential number of constraints, there also exists an efficient separation algorithm for the cycle inequalities. This algorithm was described at the start of Chapter 6. If one was only interested in optimizing over $\mathrm{MET}(G)$, this could be accomplished easily by solving the optimization problem on $\mathrm{MET}_n^\square$ with weights of zero on edges not appearing in G. This would not require a characterization or separation algorithm for $\mathrm{MET}(G)$. However, there are problems where people are more interested in determining if a vector violates an inequality of MET(G), an example of such problems are known as partial completion problems [57]. In this case, having an explicit characterization of MET(G) is useful. It could also be the case that considering $n(n-1)$ arcs is impractical for large n if G has far fewer arcs.

In investigating if a similar result holds for the projection of the directed metric cone, we did not find such a nice characterization. To begin describing what we think the characterization of the projection of the directed semimetric polytope is, we will begin by focusing on two types of inequalities already introduced in Chapter 6. It was proved in Section 6.1 that both inequalities (6.3):

$$x_{ij} \leq \sum_{a \in P_{ij}} x_a \tag{7.1}$$

and (6.10):

$$x_{i1} + x_{1j} \leq \sum_{a \in C'} x_a \tag{7.2}$$

are facet inducing inequalities under certain conditions. It can be shown that under these conditions (6.3) and (6.10) are facets of the projection of DMET_n onto the arc set of a graph G, ie. DMET(G).

To check if an inequality of type (6.3) or (6.10) is violated by a given vector $y \in \mathbb{R}^{|A(G)|}$ is straight forward. To check if an inequality of type (6.3) is violated, for each arc ij we can try and find a shortest path in the graph $G \setminus \{ij\}$ from i to j and check it the path length is less than y_{ij}, if so we have found a violated inequality.

For a subraph of $\vec{J}_n$ every cycle must include the node 1 as the graph induced on $\vec{J}_n \setminus 1$ is acyclic. For inequalities of type (6.10) and the pair of nodes i and j with

$i < j$, we can check if $y \in \mathbb{R}^{|A(G)|}$ violates the inequality by summing the length of the shortest path from node 1 to i, the shortest path from i to j and the shortest path from j to 1 and checking if this sum is less than $y_{i1} + y_{1j}$. Doing this for each possible pair i and j will find a violated inequality of type (6.10) if one exists.

To begin to characterize DMET(G) we will construct a auxiliary graph G' where G' contains two types of nodes. One type of node, which we refer to as type A, corresponds to 4−tuples of nodes of G, where $(i, j, k, l) \in V(G')$ if $i, j, k, l \in V(G)$ for $i < j < k < l$. This node corresponds to the inequality (6.10):

$$x_{j1} + x_{1k} \;\; \leq \;\; \sum_{a \in C} x_a$$

where the cycle $C \in G$ is the arc $1i$, a directed path from i to j, a directed path from j to k, a directed path from k to l and the arc $l1$. If any of the arcs $1i$,$j1$, $1k$ and $l1$ do not exist in $A(G)$ then the Fourier-Motzkin elimination method tells us they must have been eliminated through the addition of another valid inequality. This is the basis of the construction of the auxiliary graph G'. If the arc $j1$ does not appear in $A(G)$ then node (i, j, k, l) in G' has an arc directed to nodes of the form $(i', j', k', l') \in G'$ where $i' < j' < k' < l'$ and $j = l'$. Similarly, if arc $1k$ does not appear in G then node (i, j, k, l) has an arc directed to nodes of the form (i', j', k', l') in $V(G')$ if $i = k'$ for each such $(i', j', k', l') \in V(G')$.

The second type of nodes in G', which we refer to as type B, will correspond to the 4− tuple (i, p, q, l) where $pq \in A(G)$, $i < p$ and $l < q$. This node is based on the inequality of the form (6.3):

$$x_{pq} \leq \sum_{a \in P_{pq}} x_a$$

where P_{pq} is a path from p to l in $G \backslash \{pq\}$ followed by the arcs $l1$ and $1i$ followed by a directed path from i to p. If arc $l1$ does not exist in $A(G)$ then a node (i', j', k', l') of type A in G' will have an arc directed to (i, p, q, l) if $l = j'$ for each such $(i', j', k', l') \in V(G')$. If arc $i1$ does not exist in $A(G)$ then nodes of type A in G' of the form (i', j', k', l') will have an arc directed to (i, p, q, l) if $i = k'$.

Each node of G' will be assigned a node weight. A node (i, j, k, l) of type A will have a weight equal to:

$$y_{j1} + y_{1k} - y_{l1} - y_{1i} - \sum_{a \in P_{ij}} y_a - \sum_{a \in P_{jk}} y_a \sum_{a \in P_{kl}} x_a. \tag{7.3}$$

Here P_{ij}, P_{jk} and P_{kl} are the shortest paths between nodes i and j, j and k, and k and l respectively in G with arc weights from the vector y. If an arc does not exist in G its corresponding y value is zero in (7.3).

Nodes of type B will have a weight equal to:

$$y_{pq} - y_{1i} - y_{l1} \sum_{a \in P_{pl}} y_a - \sum_{a \in P_{iq}} y_a. \tag{7.4}$$

We are now interested in subgraphs $T \subseteq G'$ with the following properties. For all nodes $(i, j, k, l) \in T$ of type A, if arc $i1$ is not in G then T contains a node (i', j', k', l') where $i = k'$ and the arc from (i', j', k', l') to (i, j, k, l). If $1l$ is not in $A(G)$ then T must contain a node (i', j', k', l') of type A where $l = j'$ and the arc from (i', j', k', l') to (i, j, k, l).

For all nodes (i,p,q,l) of type B in T, if $l1$ is not in $A(G)$ then T must contain a node (i',j',k',l') of type A where $l=j'$ and the arc from (i',j',k',l') to (i,j,k,l). If $1i$ is not in $A(G)$ then T must contain a node (i',j',k',l') of type A where $i=k'$ and the arc from (i',j',k',l') to (i,j,k,l).

We will refer to such a subgraph T as a *directed inequality subgraph* of G' and we denote this as $T \subseteq_{DIS} G'$. If we sum the weights assigned to nodes of any subgraph T with the desired properties the result must be non-positive or we have found an inequality which the vector y violates, we denote the weight of a directed inequality subgraph with respect to a vector x as $x(T)$. By construction it is easy to see that the inequalities that the subgraph T correspond to are valid for DMET(G).

Conjecture 44 *Let $G \subseteq \vec{J}_n$ and let G' be the constructed auxiliary directed inequality graph for G. Then,*

$$DMET(G) \quad = \quad \{x : x \in \mathbb{R}^{|A(G)|}, x_a \geq 0 \forall a \in A(G), \forall T \subseteq_{DIS} G', x(T) \leq 0\}. \tag{7.5}$$

One fundamental difference between the projection of $\mathrm{MET}_n^\square$ onto a graph H and the projection of $\mathrm{DMET}_n^\square$ onto the arcs of a directed graph G is that using the triangular elimination to eliminate edges one at a time, one can produce any cycle inequality in the undirected case, and hence the set of constraints that describe MET(H). For the directed semimetric polytope this is not the case. For example, the inequality presented in the following example, the last inequality in Figure 7–3, is a facet of DMET(G), however this inequality can not be obtained by a series of addition of inequalities that define $\mathrm{DMET}_n^\square$ that eliminate one arc per addition. In the following construction, two arcs are eliminated in the final inequality addition.

Figure 7–1 depicts the elimination of two arcs, $i1$ and $1j$ by two additions and the resulting inequality. The triangular elimination methods presented in Chapter 6 can be used to produce the inequality depicted in Figure 7–2.

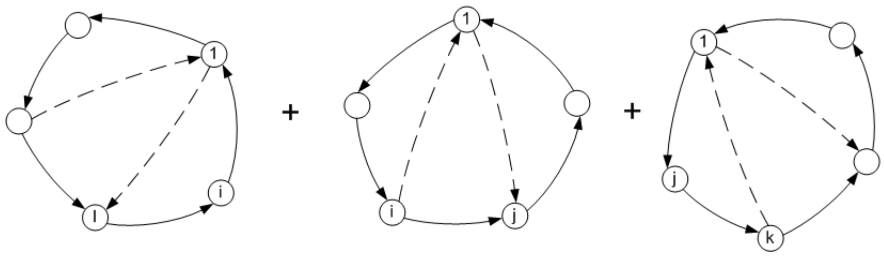


Figure 7–1: A depiction of eliminating two arcs by two additions. The sum of weights on solid arcs is greater than or equal to the sum of weights on dashed arcs.

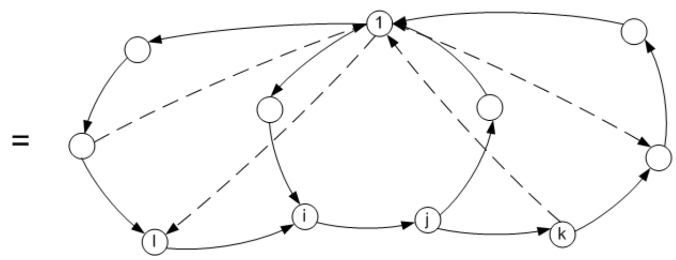


Figure 7–2: A depiction of the resulting inequality. The sum of weights on solid arcs is greater than or equal to the sum of weights on the dashed arcs.

Figure 7–3 shows the elimination of two arcs with one addition. The resulting inequality is facet inducing for DMET(G) if G is the support graph of the inequality depicted. But it does not appear to us that it is possible to produce this inequality by eliminating at most one arc per addition starting inequalities that define $\mathrm{DMET}_n^\square$.

If Conjecture 44 is true, it still only provides an exponential description of DMET(G). An exponential description is already available via Fourier-Motzkin elimination. To make the description given in Conjecture 44 useful, a polynomial time

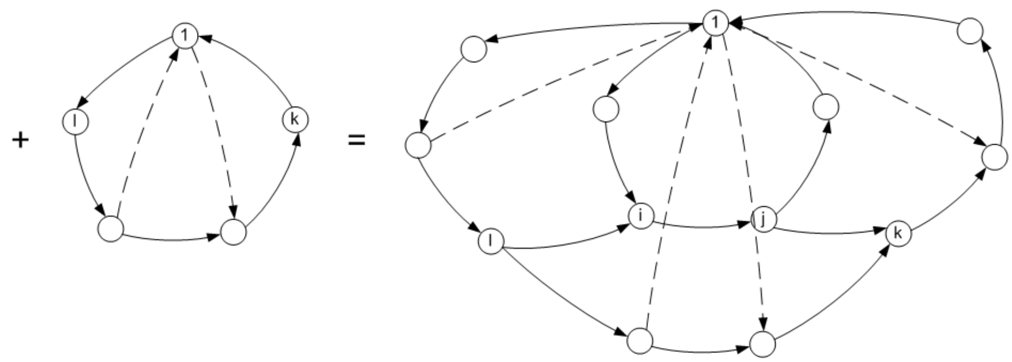


Figure 7–3: A depiction of adding another valid inequality to the inequality of Figure 7–2 and the resulting inequality. The inequality obtained has two arcs eliminated. The sum of the weights on the solid arcs must be greater than or equal to the sum of weights on dashed arcs.

separation algorithm would be needed as well. We are therefore interested in algorithms for finding a directed inequality subgraph T of minimum weight or of positive weight in the constructed graph G'.

7.2 Forbidden minors

A graph H is a *graph minor* of a graph G if it can be obtained by a series of edge contractions and edge deletions. A contraction of an edge i, j involves replacing nodes i and j with a new node and making it adjacent to $N(i) \cup N(j)$, the neighbours of i and j.

As mentioned in Section 6 the following theorem due to Seymour characterizes when the triangle inequalities describes the cut cone.

Theorem 45 ([75]) *If G does not contain a K_5 minor then CUT(G) =MET(G).*

It would be nice to characterize when the projection of DCUT(G) is completely characterized by DMET(G). Through the use of the vertex enumeration software lrs [4] we have come up with a list of subgraphs that ensure that DMET(G) $\neq$DCUT(G).

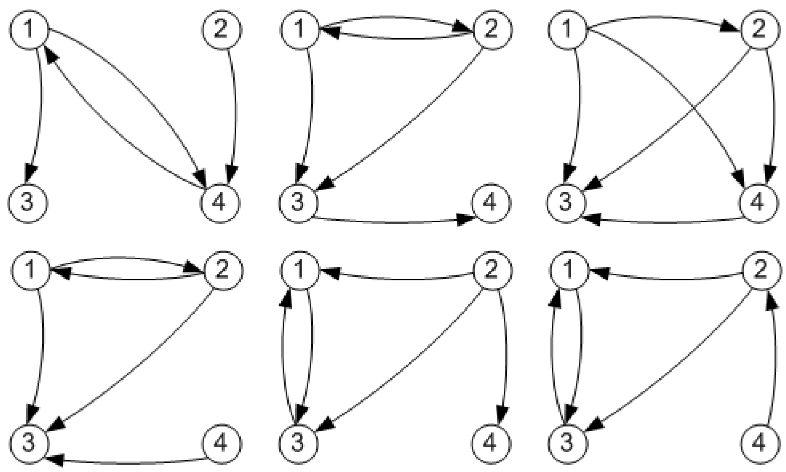


Figure 7–4: A list of directed graphs for which DMET(G) has fractional vertices.

We will define a directed graph G to be a *directed minor* of a graph H if G can be obtained from H by performing a series of arcs deletions and directed contractions. A directed contraction in this context will mean contracting two nodes i and j into a new node v, if either ij or ji (or both) are arcs of $A(G)$, and adding arcs $\{vu : u \in V(G), iu \textit{ or } ju \in A(G)\}$ and arcs $\{uv : u \in V(G), ui \textit{ or } uj \in A(G)\}$. Note, there are multiple definitions of directed minors, this definition is only for the context of this section. It appears to us that a graph containing any of the graphs in Figure 7–4 as a directed graph minor ensures that $DMET(H) \neq DCUT(H)$.

We would like to come up with a theorem proving that DCUT(G) = DMET(G) if and only graph G doesn't contain a directed minor of a given set of graphs, such as those appearing in Figure 7–4. Such a theorem would likely take a bit of work as the result of Seymour is a deep theorem that is based on the work of Wagner [85] characterizing the structure of graphs without a K_5 minor. So one would likely need a nice characterization of directed graphs not containing directed minors of a set of forbidden graphs if a similar approach is to be used.

7.3 Inequalities for the POK problem

In formulating open pit optimization problems as an directed cut problems with knapsack constraints the directed graph which arises is neither the complete directed graph $\vec{K}_n$ nor the graph $\vec{J}_n$ but a subgraph of $\vec{K}_n$. If the graph closure problem is formulated as a maximum cut as described at the end of Section 2.4 then the arcs ij representing the precedence constraints have weights of $-M$ (a large negative weight such that arc ij never appears in a maximum directed cut). Therefore, a triangle inequality of the form:

$$x_{si} + x_{ij} \quad \leq \quad x_{sj}$$

will always have $x_{ij} = 0$, so it is equivalent to $x_{si} \leq x_{sj}$. Considering all such inequalities forms a totally unimodular system as it defines the graph closure LP. To develop new valid inequalities for directed cut problems with knapsack constraints, further research on the interaction between knapsack inequalities and directed cut inequalities and the integer solutions nearby will be needed.

CHAPTER 8
Geometric complexity results

We now diverge from the study of polyhedra related to the mining optimization and focus on the complexity of some problems discussed. Finding a maximum weight closure with a cardinality constraint on the number of nodes in the closure is NP-hard in general [54], [34]. This can be shown by a reduction from max clique. From a graph G for which one wants to know if a clique of size s exists, a graph G' can be constructed on which a maximum weight closure of bounded size will be found if and only if G contains a clique of size s. To create the graph G', add nodes with weight 1 for every edge of G and add nodes of weight 0 for every vertex of G. For each edge node in G' (the nodes with weight 1) add two arcs directed to the vertices corresponding to its endpoints. It is easily shown that the graph G' has a graph closure problem with at most $\binom{s}{2} + s$ nodes of weight $\binom{s}{2}$ if and only if G has a clique of size s. This problem of finding a maximum weight closure with a cardinality constraint is also known as the maximum weight ideal problem, where the directed graph is considered as a partially ordered set (poset) and a graph closure corresponds to an ideal of the poset.

The reduction described is not very useful in our framework, however, since our directed graph has a fixed maximum degree. For a 45 degree slope constraint the 1:5:9 pattern (see [53]) produces an out and in degree of at most 14, $\delta^+(v) \leq 14$. The maximum clique problem is polynomial time solvable by simply looking at all

$\binom{n}{\Delta(G)+1}$ subsets of vertices and checking if the subset is a clique. The geometry of our graph should be considered in the reduction to validate if our instance of the constrained graph closure is still NP-hard.

As mentioned in Section 2.2 one limitation of existing pushback design algorithms is connectivity. The pushback should be physically connected and not a collection of disjoint pieces. In investigating connectivity in terms of open pit design, it can be shown that the problem of finding a maximum weight connected pit is NP-hard. In fact the proof only uses one mining level, so this can be viewed as a proof that designing an underground mine to optimize the sum of ore minus waste removed is NP-hard. The question of whether or not the POK problem (without connectivity) is NP-hard for a precedence graph created from a block model is still open.

We will prove the NP-hardness of the connected pit optimization problem by a reduction using a problem known as: connected node cover on a planar graph of maximum degree 4. A node cover of a graph $G = (V, E)$ is a subset of the vertices $S \subset V(G)$ such that every edge has at least one endpoint in the subset. A connected node cover is a node cover $S \subset V(G)$ such that the graph induced by S is connected. The decision problem associated with finding a node cover is whether or not a node cover of size k exists ($|S| = k$). It was shown in [38] that connected node cover is NP-hard even on planar graphs of maximum degree 4. From an instance of this problem, we want to construct an instance of a maximum value connected pit problem. A connected pit, will be one in which a path exists between all blocks in the pit where blocks are adjacent if they physically touch and share a common border (ie, a block will have 4 neighbours on its level).

Theorem 46 *Maximum weight connected pit is NP-hard.*

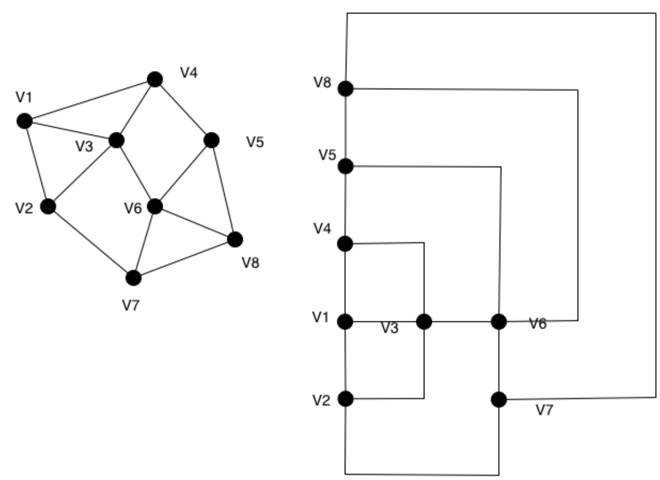


Figure 8–1: A planar graph G on the left and a grid embedding of the same graph on the right

A planar grid embedding of a graph with n vertices is a mapping of vertices to distinct Cartesian 2D grid points and edges to non-intersecting grid paths. Tamassia and Tollis [80] show that every planar graph of maximum degree 4 can be embedded in a grid of size $O(n^2)$ in linear time, where the length of every edge is $O(n^2)$. Given a planar graph $G = (V, E)$ of maximum degree 4 we can construct an instance of maximum weight connected pit where the maximum weight connected pit corresponds to the minimum weight cover. Begin by subdividing each edge of G and associating the new set of vertices $\{s_1, ..., s_m\}$ with the edge that they subdivided. The resulting graph is clearly still planar and all the new vertices have degree of $2 < 4$. Then use Tamassia and Tollis' algorithm to construct a grid embedding of our subdivided graph. Assume that the grid we have our graph embedded on is of size n' by n' (where n' is of the order of n).

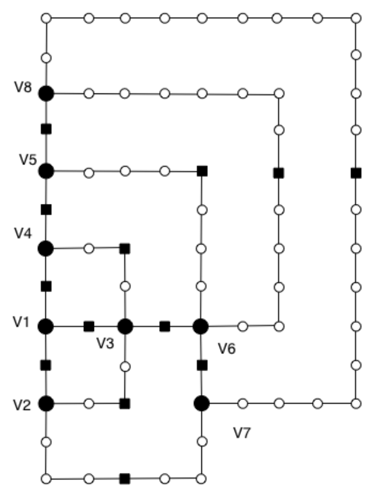


Figure 8–2: A grid embedding of G with each edge bisected (the black square nodes) and dummy nodes on every other grid point

To create an instance of a maximum connected pit problem, create a single level (bench) orebody block model of size $2n'$ by $2n'$. Note that only the top level of an orebody model is being considered, so slope constraints and angles need not be considered. For each node in our embedded graph Cartesian grid, if its location is (i, j), associate it to the block at location $(2i, 2j)$ in our orebody block model. If the node at location (i, j) is one of the subdivided edge nodes $\{s_1, ..., s_m\}$ assign the corresponding block a value of n^3. If the node at location (i, j) is one of the original nodes of our graph assign it a weight of -1. If an edge exists between two vertices v_k and v_l in our graph, assign the blocks that the edge between v_k and v_l would pass through (by taking the grid embedding of the edge and doubling its length in every direction and placing it on the block model) a value of 0, such blocks will be referred to as "zero-weight" blocks. Assign all other blocks of our block model a large negative value of $-n^4$. We claim that G has a connected node cover of size k if and only if the constructed orebody model has a connected pit of value $\geq mn^2 - k$.

Proof. If a G has a connected node cover S of size k, let $b_1, ..., b_k$ be the blocks associated with the nodes in S. Choosing all the blocks associated with the m subdivided edges and the blocks $b_1, ..., b_k$ gives a pit of value $mn^2 - k$. The blocks in this pit can be connected with blocks of value 0 since each block representing a subdivided block has a path of zero length blocks to one of $\{b_1, ..., b_k\}$.

If our constructed pit has a connected pit value of $mn^2 - k$ then the graph G has a connected node cover of size k. None of the large negative value blocks can be in our maximum connected pit, since including one will give a value of at most $mn^2 - n^4$ which is less than zero (for a planar graph m is $O(n)$), so choosing a single subdivided edge block would give a higher valued pit (n^2). It follows that our pit of value $mn^2 - k$ contains only blocks of value -1, 0, and n^2. Since there are at most m blocks of weight n^2 they must all be contained in our pit. Since each of the subdivided edge blocks (those of weight n^2) are connected through blocks of weight 0 to blocks of weight -1 every subdivided edge block must have at least one of the blocks that corresponds to its endpoints in G in the connected pit (to connect the subdivided edge block to the rest of the pit). This implies that the set of nodes, S, in our original graph that correspond to the -1 weight blocks in our pit form a connected node cover. It follows that $|S| = k$ since the connected pit would has value $mn^2 - k$. ■

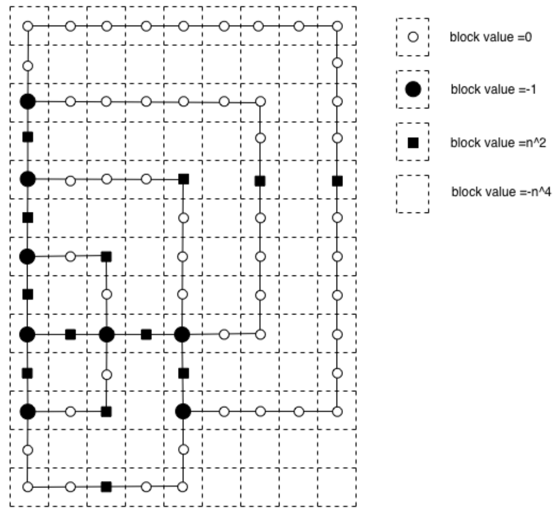


Figure 8–3: A top view of the constructed orebody model

This complexity result implies that finding a 3D dynamic programming algorithm to produces a connected pit in polynomial time is likely impossible. So there should be no way to generalize the 2D dynamic programming methods to 3 dimensions.

While this shows that added the constraint of connectedness to our problem makes the general problem harder, connectedness combined with other gap constraints such as the convex bottom, may make the problem easier. The NP-hard reduction used in this proof falls apart if we have the requirement that our pit have a convex shape at every bench. An interesting question is whether or not the problem of finding a maximum valued pit of a constrained size (no connectivity requirement) remains NP-hard.

The complexity result proved has implications on the computational complexity of underground mining. In an underground setting, blocks that are removed must be physically connected to one another. This result implies that such an optimization problem should be NP-hard as well even without a knapsack constraint.

CHAPTER 9
Experimental results - pipage rounding

A major drawback with the Lagrangian relaxation type methods is that the troublesome constraints are not strictly satisfied by the pushback produced. If the resulting pushback exceeds a constraint such as mill capacity, then a decision has to be made on which blocks should be removed from the pushback so that it adheres to the mill requirements. Similarly, if a pushback falls short of a constraint such as mill capacity, a decision must be made as to how to enlarge the pushback. These decisions are typically not done optimally in terms of maximizing discounted NPV. A second major problem with these methods is that they rely on using a predetermined cut-off grade and assigning every block a label of either waste or ore prior to the optimization stage.

We investigated the problem of finding an optimal pushback that adheres to a specific constraint using a dynamic cut-off grade. The specific constraint considered is a mill capacity for a given period and an optimal pushback is defined as one that meets the mill capacity and has the maximum profit of mining the blocks sent to the mill minus the cost of removing the waste blocks in the pushback. The integer program formulation of the problem we would like to solve is rather large. Solving such integer programs requires too much computation time in practice. However, one can often solve the linear program relaxation in a reasonable amount of time to obtain a fractional solution. We outline a method for converting the fractional solution of a linear program into an integral solution using a technique known as pipage rounding.

The general framework of pipage rounding relies on finding a non-linear function, $F(y)$, that is equal to the objective function of the IP, $f(y)$, for integral points. The linear program relaxation of the integer program is solved and the fractional optimal solution, y^*, is used to evaluate $F(y^*)$. $F(y)$ is chosen in a way such that one can round y^* to integrality while both preserving feasibility and increasing the value of $F(y)$. Informally, if the evaluated $F(y^*)$ is close to the value of $f(y^*)$ then the solution obtained from pipage rounding will be close to optimal.

The rounding step consists of taking two fractional entries, y_i and y_j, of the current vector y^* and computing $F(y)$ twice. Once with y_i rounded up and y_j rounded down, call the new vector y^+ and then for y_i rounded down and y_j rounded up, call this new vector y^-. The rounding up and down is done in such a way as to ensure that both y^+ and y^- are feasible and to ensure that at least one of y_i or y_j become integral in both y^+ and y^-. The entries of y^* are updated to y^+ if $F(y^+) > F(y^-)$ and to y^- otherwise. The process is repeated until no fractional entries remain. Clearly, there can only be a linear number of such roundings as each iteration produces at least one more integral entry.

In the application of pipage rounding to our problem, $y_i = 1$ means that block i is sent to the mill (processed) for a profit of p_j. For a block j if a block i "below" it is processed then we assume no cost for removing block j and we profit a profit c_j if block j remains in the ground (this is essentially the same as having a negative c_j value if we remove block j). One can replace a variable x_j where $x_j = 0$ if a block j is removed and $x_j = 1$ otherwise, as $x_j = \min\{1 - y_i | i \text{ } below \text{ } j\}$. This essentially puts the slope constraints into the objective function.

The LP we solve has the same optimal solution as the following optimization problem:

$$\begin{aligned} f(y) = \max \quad & \sum_{i \in V} c_i \min\{1 - y_j, (i,j) \in A(G)\} + \sum_{j \in V} p_j y_j && (9.1) \\ \textit{subject to} \quad & \sum_{j \in V} w_j y_j \leq b && (9.2) \\ & 0 \leq y_j \leq 1 \qquad \forall j \in V && (9.3) \end{aligned}$$

For the non-linear function,

$$F(y) = \sum_{i \in V} c_i \prod_{j:(i,j) \in A(G)} (1 - y_j) + \sum_{i \in V} p_i y_i$$

was chosen. We replaced,

$$\min\{1 - y_j : (i,j) \in A(G)\}$$

in the objective function with the polynomial,

$$\prod_{j:(i,j) \in A(G)} (1 - y_j).$$

For a given $i \in V$, if any y_j is 1 for $(i,j) \in A(G)$ then $\min\{1 - y_j : (i,j) \in A(G)\}$ is equal to 0 and $\prod_{j:(i,j) \in A(G)} (1 - y_j)$ is 0 as well. For a given $i \in V$, if all y_j's are zero for $(i,j) \in A(G)$ then $\min\{1 - y_j : (i,j) \in A(G)\}$ is 1 and so is $\prod_{j:(i,j) \in A(G)} (1 - y_j)$. It follows that $F(y)$ and $f(y)$ agree for integral values of vector y.

Given two fractional vector entries y_i^* and y_j^* we would like to round one up and one down while preserving constraint (9.2). Clearly, we can only decrease y_i by the minimum of y_i and the amount restricted by constraint (9.2). This amount is dependent on how much we can raise y_j, it is required that the value of $w_i y_i + w_j y_j$ remain the same after y_i is increased and y_j is decreased. An easy calculation implies that y_i can be decreased by the minimum of y_i and $(1-y_j)\frac{w_j}{w_i}$, let ϵ_1 be this minimum. Similar calculations show that we can increase y_i by at most $\epsilon_2 = \max\{1-y_i, y_j\frac{w_j}{w_i}\}$. One can deduce that the maximum amount we can increase y_j is $\epsilon_1\frac{w_i}{w_j}$ and the maximum we can decrease it is $\epsilon_2\frac{w_i}{w_j}$. If $\epsilon_1 = y_i$ then clearly y_i^* will be decreased to 0 otherwise $\epsilon_1 = (1-y_j)\frac{w_j}{w_i}$ and y_j is increased to 1. If $\epsilon_2 = (1-y_i)$ then $y_i + \epsilon_2$ is 1 otherwise $\epsilon_2 = y_j\frac{w_j}{w_i}$ and $y_j - \epsilon_2\frac{w_i}{w_j}$ is 1. This guarantees that at least one of the two fractional entries will be rounded to integrality.

The algorithm was implemented and tested on a case study of 10,000 nodes. While a malicious example could be constructed such that the non-linear function is not a good approximation to the LP, in practice it appears to work well giving a solution within 6.4% of optimal. The ability to strictly adhere to a given constraint and to determine the cut-off grade dynamically are desired features in a pushback design algorithm.

CHAPTER 10
Conclusions

The main results of this thesis concerned the polyhedral structure of the directed cut polyhedra. We primarily focused on describing valid and facet defining inequalities for the directed cut polyhedra and their relaxations. The motivation for this work was to further the understanding of these complex combinatorial objects. Such an understanding can lead to more efficient algorithms for problems that can be modelled as optimization problems on the directed cut cone or polytope.

Similar results on the undirected cut polyhedra have had impact in a very diverse array of fields as mentioned in Chapter 3. Directed cuts are also a natural formulation for many problems and have been extensively researched. There are many famous algorithms for finding minimum weight directed cuts, these algorithms are often required learning in undergraduate computer science programs. An approximate algorithm for the maximum directed cut problem appeared in a Fulkerson prize winning paper [41]. Yet the polytope associated with these maximization and minimization problem had not been researched in the same fashion as the undirected cut polytope. The polyhedral results developed are well suited to improve both cutting plane and semidefinite optimization algorithms for problems involving directed cuts.

Projections of the directed cut polytope and cone were a focus of this thesis as well. If the directed graph over which one wishes to find an optimal directed cut has a specific structure then knowing the exact form of the inequalities that define the projected cut polytope, or a relation of it, can be used to improve an algorithm's efficiency. While the optimization problem can be solved on the directed metric polytope for the complete directed graph, for a large n it may be impractical to consider $O(n^2)$ arcs when the directed graph you are interested in has an order of magnitude fewer arcs.

Having established a close link between the undirected cut polyhedra and directed cut polyhedra, we presented an algorithm for optimizing over the cut polytope when optimizing over the rooted semimetric and semimetric relaxations had the same objective value. This result can be extended to the directed cut polyhedra by using this algorithm to optimize over both of the defined polyhedra $\mathcal{P}_{1,n}$ and $\mathcal{P}_{2,n}$ and taking the better directed cut as the optimal solution. Deciding whether there exists an optimum integer solution of the same value as that of a given fractional solution can be very useful. Particularly, since in the result presented an efficient algorithm exists for computing the integral solution.

In investigating problems related to open pit mining, the main focus was not to develop an algorithm that worked well on real data sets in practice. The focus instead has mainly been theoretical in nature. An understanding of the structure of the polyhedra related to the problem can help in the development of efficient algorithms. Commercial software has had success applying techniques like Lagrangian relaxation [15] and branch and bound [19]. The cutting plane approaches that could be developed from the work of this thesis need not compete with these techniques but instead improve them.

While optimization in the mining industry has been around for many years, there still exist many decisions that could be modelled as an optimization problems that are decided either heuristically or by hand by an experienced planner. The process that many companies use for developing their long term schedule for an open pit mine ignores key factors that can greatly impact the NPV of the operation. These include optimizing cut-off grades, blending, stock piling, economic discounting, precisely meeting mill and transportation requirements, and incorporating tax models. With recent advances in orebody modelling multiple simulations of a single deposit are available to the long term planner. New economic models of mining operations involving real options are gaining acceptance. The optimization algorithms used in long term planning will need to change to reflect these recent advances.

When multiple deposits are potentially feeding the same processing facility or multiple deposits can feed multiple processing facilities, optimization algorithms need to simultaneously schedule the long term plan for each facility and deposit. An area of study termed “global optimization” has recently become of interest to mining companies [83]. This field can be simply viewed as taking multiple decisions that have traditionally been made in a piecewise fashion and instead produce algorithms that make these decisions concurrently to obtain better solutions. These solutions can lead to substantial improvements in the NPV of a mining operation. While problems related to mining width and connectivity may still remain elusive, many global optimization problems can easily be modelled using directed cuts and knapsack constraints. They can be modelled in such a way as to provide solutions that solve the other issues discussed in Chapter 2.1.

The area of optimization in the mining industry remains an underdeveloped area of research. One major problem seems to be related to a lack of communication between industry and optimization reserchers. The mining process is a complex operation. It takes years of hands-on experience at an operational level to understand all of the optimization opportunities that exist. It can be difficult to obtain a solid perspective on what can be safely changed. To solve these problems lines of communication between operations and research must be fostered. The work in this thesis is a result of such collaboration at McGill's COSMO mine planning laboratory. It would also seem beneficial to bring people with optimization backgrounds into an operational setting so that they can better understand the problems and opportunities that arise.

With growing concerns on emissions related to large-scale industry, the study of better optimization algorithms is now more important than ever. Being able to effectively model every aspect of a mining operation as an optimization problem is not only of interest economically but environmentally. The feasibility of projects in the future may rely on modelling and limiting carbon emissions related to projects, such problems seem to fall under the framework of precedence constraints and knapsack inequalities.

Appendix A

Vertices of DMET and RDMET:

Table 10–1: Vertices of $\mathrm{DMET}_3^{\square}$ and $\mathrm{RDMET}_3^{\square}$

Table 10–1

x_{12}	x_{13}	x_{21}	x_{23}	x_{31}	x_{32}
0	0	0	0	0	0
1	1	0	0	0	0
0	1	0	1	0	0
0	0	1	1	0	0
0	0	1	0	1	0
0	0	0	0	1	1
1	0	0	0	0	1

Table 10–2: Vertices of $\mathrm{RDMET}_4^{\square}$

Table 10–2

x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0

Table 10–2

x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
0	0	0	1	1	1	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0	0
0	1/2	0	1/2	1/2	1/2	1/2	0	0	1/2	1/2	1/2
0	1/2	1/2	1/2	1	1	0	0	1/2	0	0	1/2
0	1/2	0	1/2	1	1/2	0	0	0	1/2	1/2	1
1/2	1/2	1/2	0	1/2	0	0	1/2	0	0	0	0
1/2	1/2	1/2	0	0	0	0	0	1/2	0	0	1/2
1/2	1/2	1/2	1/2	1/2	1/2	0	0	1/2	0	0	1/2
0	0	1/2	1/2	1/2	1	1/2	1/2	1	0	0	0
0	0	1	0	0	1	0	0	1	0	0	0
0	0	0	1	0	1	1	0	1	0	0	0
1/2	0	1/2	0	0	1/2	1/2	1	1	0	1/2	0
1	0	1	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	1/2	0	0	1/2	0	1/2	1/2	0	1/2
1/2	0	0	1/2	0	0	1/2	1/2	1/2	1/2	1/2	1/2
0	0	0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2
1/2	0	0	0	0	0	1/2	1	1/2	1/2	1	1/2
1/2	1/2	1/2	0	1/2	1/2	0	1/2	1/2	0	1/2	1/2

Table 10–2

x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
0	0	0	1	0	0	1	0	0	1	0	0
0	0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	0	0
0	0	0	1/2	1/2	0	1/2	1/2	0	1/2	0	0
0	0	0	1	1	0	0	0	0	1	0	1
1/2	1/2	1/2	0	0	1/2	0	0	0	0	1/2	0
1/2	1/2	1/2	0	1/2	0	0	1/2	0	1/2	1/2	1/2
1/2	1/2	1/2	0	0	1/2	1/2	1/2	1/2	0	1/2	0
1	0	0	0	0	0	0	1	0	0	1	0
0	0	0	1/2	0	1/2	1/2	0	0	1/2	1/2	0
0	0	0	0	0	0	1	1	0	1	1	0
0	0	0	0	0	0	0	0	0	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1
1/2	1/2	0	0	1/2	0	0	1/2	0	1/2	1	1
0	1	0	0	1	0	0	0	0	0	0	1

Table 10–3: Vertices of $\mathrm{DMET}_4^\square$

Table 10–3

x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0	0
0	1/2	0	1/2	1/2	1/2	1/2	0	0	1/2	1/2	1/2
0	0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	0	0
0	0	1	0	0	1	0	0	1	0	0	0
0	0	0	1	0	1	1	0	1	0	0	0
1	0	1	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0
1/2	0	0	1/2	0	0	1/2	1/2	1/2	1/2	1/2	1/2
0	0	0	1	0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0	0	1	0	1
1/2	1/2	1/2	0	1/2	0	0	1/2	0	1/2	1/2	1/2
1/2	1/2	1/2	0	0	1/2	1/2	1/2	1/2	0	1/2	0
1	0	0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	1	1	0	1	1	0
0	0	0	0	0	0	0	0	0	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1

Table 10–3

x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
0	1	0	0	1	0	0	0	0	0	0	1
1/2	1/2	1/2	1/2	1/2	1/2	0	0	1/2	0	0	1/2

A list of vertices of $\mathrm{RDMET}_5^\square$ and $\mathrm{DMET}_5^\square$ can be obtained here:

`http://cgm.cs.mcgill.ca/~cmeagh1/DMET5/dmet5.pdf`

References

[1] A. Akaike *Strategic planning of long term production schedule using 4D network relaxation method*, PhD Dissertation Colorado School of Mines, 1999.

[2] N. Alon and A. Noar *Approximating the Cut-Norm via Grothendieck's inequality*, Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 72-80, 2004.

[3] S. Anzai, J. Chun, R. Kasai, M. Korman and T. Tokuyama, *Effect of Corner Information in Simultaneous Placement of k-Rectangles and Tableaux*, In Proceedings of the 16th international conference on Computing and Combinatorics (COCOON'10), LNCS 6196:235-243, 2010.

[4] D. Avis. *lrs Homepage*. `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`.

[5] D. Avis, H. Imai, T. Ito and Y. Sasaki, *Two-party Bell inequalities derived from combinatorics via triangular elimination*, Journal of Physics A: Mathematical and General, 38(50):10971-10987, 2005

[6] D. Avis, H. Imai and T. Ito, *Generating facets for the cut polytope of a graph by triangular elimination*. Mathematical Programming, A 112(2):303-325, 2008.

[7] D. Avis and T. Ito, *New classes of facets of the cut polytope and tightness of Imm22 Bell inequalities*, Discrete Applied Mathematics,155(13):1689-1699, 2007.

[8] D. Avis and J. Umemoto, *Stronger linear programming relaxations for max-cut*. Mathematical Programming, B 97:451-469, 2003.

[9] E. Balas, *Facets of the knapsack polytope*, Mathematical Programming, 8:146-164, 1975.

[10] F. Barahona, *On cuts and matchings in planar graph*, Mathematical Programming, 60:53-58, 1993.

[11] F. Barahona, M. Grötschel, and A. R. Mahjoub. *Facets of the bipartite subgraph polytope* Mathematics of Operations Research, 10(2):340-358, 1985.

[12] F. Barahona and A. R. Mahjoub. *On the Cut Polytope*. Mathematical Programming, 36(2): 157-173, 1986.

[13] F. Barahona, M. Grötschel, M. Jünger and G. Reinelt. *An application of combinatorial optimization to statistical physics and circuit layout design*, Operations Research, 36:493-513, 1988.

[14] F. Barahona, M. Jünger and G. Reinelt. *Experiments in quadratic 0-1 programming*, Mathematical Programming, 44:127-137, 1989.

[15] D. Bienstock and M. Zuckerberg, *Solving LP relaxations of large-scale precedence constrained problems*, Integer Programming and Combinatorial Optimization, LNCS 8060:1-14, 2010.

[16] N. Boland, A. Bley, C. Fricke, G. Froyland and R. Sotirov, *Clique-based facets for the precedence constrained knapsack problem*,Technical report, Tilburg University Repository `http://arno.uvt.nl/oai/wo.uvt.nl.cgi` (Netherlands), 2005.

[17] V. A. Bondarenko and B. V. Uryvaev, *On one problem of integer optimization*. Automation and Remote Control, 68(6):948-953, 2007.

[18] E. A. Boyd *Polyhedral results for the precedence-constrained knapsack problem*. Discrete Applied Mathematics, 41:185-201, 1999.

[19] L. Caccetta and S.P. Hill, *An application of branch and cut to open pit mine scheduling*, Journal of Global Optimization, 27:349-365, 2003.

[20] M. Charikar, K. Makarychev and Y. Makarychev. *Directed metrics and directed graph partitioning problems*. Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm

[21] J. Chun, K. Ryosei, M. Korman and T. Tokuyama, *Algorithms for Computing the Maximum Weight Region Decomposable into Elementary Shapes* In Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC'09), LNCS, 1166-1174, 2009.

[22] K. Dagdelen And T.B. Johnson *Optimum open pit mine production Scheduling by Lagrangian parametrization*, 19th Application of Computers in Operations Research in the Mineral Industry, 127-142, 1986.

[23] C. De Simone, *The Cut Polytope and the Boolean Quadric Polytope*. Discrete Mathematics, 79:71-75, 1990.

[24] C. De Simone, *Lifting facets of the cut polytope*. Operations Research Letters, 9(5):341-344, 1990.

[25] C. De Simone, M. Deza and M. Laurent *Collapsing and lifting for the cut cone*, Discrete Mathematics, 127:105-130, 1994.

[26] M. Deza. *Matrices de formes quadratiques non négatives pour des arguments binaires*. Comptes rendus de l'Académie des Sciences de Paris, 277(1):873-875, 1973.

[27] M. Deza and M. Laurent. *Facets for the cut cone II: Clique-web inequalities.* Mathematical Programming, 56(2):161-188, 1992.

[28] M. Deza and M. Laurent. *Applications of cut polyhedra I* Journal of Computational and Applied Mathematics, 55:191-216, 1994.

[29] M. Deza and M. Laurent, *Applications of cut polyhedra II.* Journal of Computational and Applied Mathematics 55 55:217-247, 1994.

[30] M. Deza and M. Laurent, *Geometry of Cuts and Metrics*, Springer, 1997.

[31] P.A. Dowd, *Risk in minerals projects: Analysis, perception and management*, IMM Transactions (Sect. A: Mining Industry), 106:A9-A18, 1997.

[32] R. Dimitrakopoulos, C. Farrelly and M. Godoy *Moving forward from traditional optimization: Grade uncertainty and risk effects in open pit mine design*, IMM Trans., 111:A82-A89, 2002.

[33] R. Dimitrakopoulos and S. Ramazan, *Stochastic integer programming for optimizing long term production schedules of open pit mines: methods, application and value of stochastic solutions*, Trans. Inst. Min. Metall. A, 117A(4):155-167, 2008.

[34] U. Faigle and W. Kern, *Computational complexity of some maximum average weight problems with precedence constraints*, Operations Research, 42(4):1268-1272, 1994.

[35] U. Feige and M. Goemans, *Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT* , In Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems, IEEE Computer Society Press, 182–189, 1995.

[36] C. Fricke, *Applications of Integer programming in Open Pit Mining*, PhD Thesis, University of Melbourne, 2006.

[37] G Gallo, M.D. Grigoriadis and R.E. Tarjan *A fast parametric maximum flow algorithm and applications*, SIAM Journal on Computing, 18(1):30-55, 1989.

[38] M.R. Garey and D.S. Johnson, *The rectilinear steiner tree problem is NP-complete*, SIAM Journal on Applied Math., 32(4):826-824, 1997.

[39] M. Godoy *The effective management of geological risk in long-term production scheduling of open pit mines*, PhD thesis, The University of Queensland, Brisbane, Qld., 1-354, 2003.

[40] M. Godoy and R. Dimitrakopoulos, *Managing risk and waste mining in long-term production scheduling*, SME Transactions, 316:43-50, 2004.

[41] M. Goemans and D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM 42:1115-1145, 1995.

[42] R.E. Gomory, *Some polyhedra related to combinatorial problems*, Linear algebra and its applications, 2:451-558, 1969.

[43] M. Grötschel, M. Jünger and G. Reinelt. *Calculating exact ground states of spin glasses: A polyhedral approach*, In Proceedings of the Heidelberg colloquium on glassy dynamics, J. L. van Hemmen and I. Morgenstern editors 325-353, 1985.

[44] M. Grötschel, L. Lovász, A. Schrijver, *The Ellipsoid Method and its Consequences in Combinatorial Optimization*, Combinatorica 1:169-197, 1981.

[45] M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, 1988.

[46] Z. Gu, G.L. Nemhauser and M.W.P. Savelsbergh, *Lifted cover inequalities for 0-1 integer programs: Computation.* INFORMS Journal on Computing 10:427437, 1998.

[47] M. Hajiaghayi, K. Jain, K. Konwar and L. Lau, *The minimum k-colored subgraph problem in haplotyping and DNA primer selection* Proc. Int. Workshop on Bioinformatics Research and Applications, Jan 2006.

[48] P.L. Hammer, *Some network flow problems solved with pseudo-boolean programming*, Operations Research, 28:121-155, 1984.

[49] P.L. Hammer, E.L. Johnson and U.M. Peled, *Facets of regular 0-1 polytopes*, Mathematical Programming 8:129-206, 1975.

[50] D. Hartvigsen and E. Zemel, *The complexity of lifted inequalities for the knapsack problem*, Discrete Applied Math. 93:113-123, 1992.

[51] C. Helmberg, F. Rendl and R. Weismantel, *Quadratic Knapsack Relaxations Using Cutting Planes and Semidefinite Programming*, Lecture Notes in Computer Science, Volume 1084, Springer Berlin / Heidelberg, 175-189, 1996.

[52] D. Hochbaum *A new-old algorithm for minimum-cut and maximum-flow in closure graphs*, Networks, 37(4):171-193, 2001.

[53] D. Hochbaum and A. Chen *Performance Analysis and Best Implementations of Old and New Algorithms for the Open-Pit Mining Problem*, Operations Research, 48(6):894-914, 2000.

[54] D.S. Johnson and K.A. Niemi, *On knapsacks, partitions and a new dynamic programming technique for trees*, Mathematics of Operations Research 8: 1-14, 1983.

[55] R.M. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations (Proceedings of a symposium on the Complexity of Computer Computations, R.E. Miller, J.W. Thatcher eds.) Plenum Press, 85-103, 1972.

[56] S.G. Kolliopoulos and G. Steiner, *Partially ordered knapsack and applications to scheduling*, Discrete Applied Mathematics, 155(8):889–897, 2007.

[57] M. Laurent, *Cuts, matrix completions and graph rigidity*, Mathematical Programming, 79:255-283, 1997.

[58] R.L.M.J. Van de Leensel, C.P.M. Van Hoesel, and J.J. Van de Klundert, *Lifting valid inequalities for the precedence constrained knapsack problem.* Mathematical Programming Series A, 86: 161-185, 1999.

[59] H. Lerchs and I. F. Grossmann, *Optimum design Of Open Pit Mines*, Canadian Institute of Mining, 17-24, 1965.

[60] N. Linial, E. London and Y. Rabinovich, *The geometry of graphs and some of its algorithmic applications*, Combinatorica, 15:215-245, 1995.

[61] C. Meagher, R. Dimitrakopoulos and D. Avis, *A New Approach to Constrained Open Pit Pushback Design Using Dynamic Cut-Off Grades*, In proceedings, Orebody Modelling and Strategic Mine Planning 2009, AusIMM, 171-176, 2009.

[62] C. Meagher, S.A. Abdel Sabour and R. Dimitrakopoulos, *Pushback Design of Open Pit Mines Under Geological and Market Uncertainties*, In proceedings, Orebody Modelling and Strategic Mine Planning 2009, AusIMM, 297-304, 2009.

[63] D.C.W. Muir, *Pseudoflow, new life for Lerchs-Grossman pit optimization*, In Proceedings of the international symposium on orebody modelling and strategic mine planning: Uncertainty and risk management, 39-48, 2004.

[64] G.L. Nemhauser and L.E. Trotter, Jr., *Properties of vertex packing and independence systems polyhedra*, Mathematical Programming, 6:48-61, 1974.

[65] Y. Nesterov and A. Nemirovskii, *Self-Concordant Functions and Polynomial Time Methods in Convex Programming* Central Economic and Mathematical Institute, Academy of Science, Moscow, 1990.

[66] M.W. Padberg, *On the facial structure of set packing polyhedra*, Mathematical Programming 5:199-215, 1973.

[67] M.W. Padberg, *Covering, packing, and knapsack polytopes*, Annals of Discrete Mathematics 4, North-Holland, Amsterdam 265-287, 1979.

[68] K. Park and S. Park, *Lifting cover inequalities for the precedence-constrained knapsack problem.* Discrete Applied Mathematics, 72:219-241, 1997.

[69] J.C. Picard, *Maximal closure of a graph and applications to combinatorial problems*, Management Science, 22:1268-1272, 1976.

[70] S. Ramazan and K. Dagdelen, *A New Pushback Design Algorithm In Open Pit Mining*, Mine Planning and Equipment, 119-124, 1998.

[71] S. Ramazan *The new fundamental tree algorithm for production scheduling of open pit mines*, European Journal of Operational Research, 177(2):1153-1166, 2007.

[72] M. Rossi, *Improving the estimates of recoverable reserves*, Mining Engineering, Jan.: 50-54, 1999.

[73] P. Schuurman and G. J. Woeginger, *Polynomial time approximation algorithms for machine scheduling: ten open problems* J. of Scheduling, 2:203-213, 1999.

[74] F. Seymour, *Pit limit parameterization from modified 3D Lerchs-Grossmann algorithm*, Manuscript, 1-11, 1994.

[75] P.D. Seymour, *Matroids and multicommodity flows*, European Journal of Combinatorics 2:257-290, 1981.

[76] A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, 1986.

[77] P. Stone, G. Froyland, M. Menabde, B. Law, R. Pasyar and P. Monkhouse, *Blasor - Blended iron ore optimisation at Yandi, Western Australia* Orebody Modelling and Strategic Mine Planning, AusIMM Spectrum Series 14:117-120, 2005.

[78] B. Tachefine, *Methods d'optimisation pour la planification de la production dans une mine à ciel ouvert*, Ph.D. Dissertation, École Polytechnique Montreal, 1996.

[79] B. Tachefine and F. Soumis, *Maximal closure on a graph with resource constraints*, Computers and Operations Research, 24(10):981-990, 1997.

[80] R. Tamassia and I.G. Tollis, *Planar grid embedding in linear time*, IEEE Transactions on Circuits and Systems, 36(9):1230-1234, 1989.

[81] B. Tolwinski and R. Underwood, *A scheduling algorithm for open pit mines*, IMA Journal of Mathematics Applied in Business & Industry 7:247-270, 1996.

[82] F. Whittle, *A decade of open pit mine planning and optimisation - the craft of turning algorithms into packages*, Proc. 28th APCOM Symposium, 15-24, 1999.

[83] G. Whittle, *Global asset optimisation*, Orebody modelling and Strategic Mine Planning Symposium, Perth, 2004.

[84] G.J. Woeginger, *On the approximability of average completion time scheduling under precedence constraints*, In Proc. 28th ICALP, 887–897, 2001.

[85] K. Wagner, *Über eine eigenschaft der ebenen komplexe*, Mathematische Annalen, 114:570-590, 1937.

[86] L.A. Wolsey, *Faces of linear inequalities in 0-1 variables*, Mathematical Programming, 8:165-178, 1975.

[87] L.A. Wolsey, *Facets and strong valid inequalities for integer programs*, Operations Research, 24:367-372, 1976.

[88] L.A. Wolsey, *Integer Programming*, New York: Wiley, 1998.