From Valid Inequalities to Heuristics: A Unified View of Primal-Dual Approximation Algorithms in Covering Problems

Dimitris Bertsimas and Chung-Piaw Teo

OR 294-94 June 1994

From valid inequalities to heuristics: a unified view of primal-dual approximation algorithms in covering problems

Dimitris Bertsimas *   Chung-Piaw Teo †

June 1994

Abstract

We propose a primal dual approach to design approximation algorithms from stronger integer programming formulations of the covering type. We also quantify the notion of strength of different valid inequalities for discrete optimization problems of the covering type and show that the proposed primal dual algorithm has worst case performance bounded by the strength of the valid inequalities used in the algorithm and the bound is tight. This bound generalizes a large class of results obtained in the literature and produces several new ones. By introducing the notion of reducible formulations, we show that it is relatively easy to compute the strength of various classes of valid inequalities for problems with reducible formulations. We also propose a multiphase extension of the primal dual algorithm and apply it to a variety of problem classes.

*Dimitris Bertsimas, Sloan School of Management and Operations Research Center, MIT, Cambridge, MA 02139. Research partially supported by a Presidential Young Investigator Award DDM-9158118 with matching funds from Draper Laboratory.

†Operations Research Center, MIT. Research supported by a graduate scholarship from the National University of Singapore.

1 Introduction

In the last twenty years, two approaches to discrete optimization problems have emerged: **polyhedral combinatorics and approximation algorithms**. Under the first approach researchers formulate problems as integer programs and solve their linear programming relaxations. By adding strong valid inequalities (preferably facets) of the convex hull of solutions to enhance the formulations, researchers are able to solve large scale discrete optimization problems within a branch and bound or branch and cut framework. Extensive computational experience suggests that the success of this approach critically depends on the choice of the valid inequalities. The principal difficulty with this approach, however, is that it is not a priori clear which class of valid inequalities is better to use at particular instances. Typically the research community depends on computational experience to evaluate the power of different valid inequalities.

The second approach involves the design and analysis of approximation algorithms. The quality of solutions produced is usually judged by the worst case criterion, for which there are two main motivations: a) understanding, from a theoretical point of view the class of problems that can be approximated well, b) designing algorithms for problems that are robust, i.e., work well for all inputs. The area has produced significant insight into our finer understanding of $\mathcal{NP}$ and for some problems it has produced algorithms which have been successfully used in practice. Despite its success, we believe that there some difficulties in the area:

1. Approximation algorithms are designed to produce the best worst case bound, which usually adds considerable complexity into the design and analysis of the algorithm. Most importantly, the success of approximation algorithms in practice has been questionable. For example the Christofides heuristic for the traveling salesman problem under triangle inequality that has the best known worst case bound of $3/2$ from optimum is consistently outperformed by various methods whose worst case behavior is particularly bad.

2. There is a lack of a unified method to construct approximation algorithms. Insights gained from one successful analysis typically do not transfer to another. Moreover, finding the worst case performance of an approximation algorithm is often a nontrivial task involving ingenious but often adhoc arguments.

3. Approximation algorithms are somewhat inflexible as they typically generate a single feasible solution. In part, this is due to the need to facilitate the analysis, but it makes

these algorithms unsuitable in situations when one needs to generate a large number of good feasible solutions, which can be ranked subsequently via other selection criteria. Another advantage of generating many feasible solutions is that the best solution selected from the list of candidate solutions may after all improve upon the worst case bound guaranteed by each individual solution! Balakrishnan et.al. [2] showed that certain network design problems have this characteristic.

In summary, guaranteed worst case bounds for approximation algorithms do indeed provide qualitative insight on their performance, but simplicity, time complexity and flexibility are also essential features for a good approximation algorithm that can be used reliably in applications.

In recent years progress in approximation algorithms has crystalized the idea that to a large extent our ability to design good approximation algorithms depends on tight integer programming formulations, i.e., there is a deeper connection between approximability of discrete optimization problems and strong formulations of these problems as integer programming problems (see Bertsimas and Vohra [3]).

Our goals in this paper is to propose an approach to design approximation algorithms from stronger integer programming formulations and to provide a way to judge the strength of valid inequalities for discrete optimization problems. We address covering problems of the form:

$$(IP)\quad IZ = \min\ cx$$
$$\text{subject to}\quad Ax \geq b$$
$$x \in \{0,1\}^n,$$

where A, c have nonnegative integer entries; entries in b are integral but not restricted to be nonnegative, since rows corresponding to negative b_i are redundant. We denote with Z the value of the linear programming relaxation, in which we relax the integrality constraints $x \in \{0,1\}^n$ with $x \geq 0$. Our contributions in this paper are as follows:

1. Given a valid inequality $\alpha_i x \geq \beta_i$ in a class $\mathcal{F}$ we introduce the notion of **strength** λ_i of this inequality as well as the notion of strength $\lambda_{\mathcal{F}}$ of the class $\mathcal{F}$. We also introduce the notion of **reducible formulations** for covering problems. This class includes a large collection of problems, including general covering problems, all the problems considered in [8, 9, 20], polymatroids, intersections of polymatroids, network design problems, etc. For reducible formulations we show that it is relatively easy to bound the strength of a class of inequalities.

2. Inspired by the primal-dual methods proposed recently in [8, 9, 20] for cut covering problems, we propose a general primal-dual approximation algorithm and a multiphase extension that uses valid inequalities in a class $\mathcal{F}$, and show that the worst case behavior of the primal-dual algorithm is bounded by the strength $\lambda_{\mathcal{F}}$. As a by-product, we also obtain bounds between the optimal integer programming value and its LP relaxation. The algorithm generalizes earlier work of [8, 9, 20] to general covering problems and uses a new (and in our opinion considerably simpler) inductive proof to show the bound. By using geometric arguments we show that the analysis is tight, i.e., the notion of strength is **inherent** in the primal-dual approach and not an artifact of the analysis.

   In addition we propose a multiphase extension of the th primal dual method and prove a bound for its worst case performance.

3. We apply the primal dual algorithm and its multiphase extension to a variety of problem classes, matching or improving upon the best known guarantee for the problem. We also prove the integrality of several polyhedra using the primal-dual algorithm.

We believe that the proposed approach addresses to a large extent some of the difficulties that the areas of polyhedral combinatorics and approximation algorithms have experienced:

1. Regarding the choice of the class of valid inequalities to use in a branch and cut exact algorithm or a primal-dual approximate algorithm for a discrete optimization problem, we propose the notion of strength of the class of valid inequalities, which is easily computable at least for reducible formulations, as the criterion to differentiate valid inequalities.

2. Regarding flexibility in approximation algorithms, by varying the class of valid inequalities we use in the primal-dual approach, we can produce a large collection of feasible solutions, each of which has a guarantee for its suboptimality. In this way we achieve two goals: From a theoretical viewpoint, progress in deriving stronger inequalities translates in potentially better worst-case bounds and from a practical perspective, we can generate a large collection of feasible solutions, which even if they have the same worst-case guarantee, they might have very different average case behavior.

3. Bounding the worst-case performance of a primal-dual algorithm for a problem that has a reducible formulation is now reduced to the considerable easier problem of computing the strength of a valid inequality. In this way we can calculate **a priori** the

bound of a primal-dual method knowing that the bound is tight. This also eliminates the often nontrivial task of providing special examples that show tightness.

The paper is structured as follows. In Section 2 we describe the general primal-dual approximation algorithm, introduce the notion of strength of a set of valid inequalities and prove that the performance of the primal-dual algorithm is bounded by the strength. Furthermore, we show using a geometric argument that the bound is tight. In Section 3, we introduce the notion of reducible formulations and show that a large collection of problem formulations fall into this framework. We further show how to compute the strength of a large collection of problems that have reducible formulations and show that our result encompasses and unifies a large set of results in the literature. In Section 4 we consider extensions of the basic primal dual algorithm to more general problems.

2 A Primal-Dual Approximation Algorithm

In this section we propose and analyze a primal dual approximation algorithm for problem (IP). Before presenting the algorithm formally we first illustrate informally the ideas on which the algorithm is based. At each step the algorithm introduces a valid inequality, updates the dual variables and the costs, fixes one variable to one, thus reducing the size of the problem. An important step of the algorithm is the idea of reverse deletion originated in [8] in the context of cut covering problems, in which we set variables that were previously set to one, equal to zero in order to ensure that the solution produced is minimal. More formally the algorithm is as follows:
Primal-dual Algorithm $\mathcal{PD}$

- Input : $A, b, c,\ (A, c \geq 0)$.
- Output : x feasible for (IP) or conclude that the problem is infeasible.

1. **Initialization** : Let $A^1 = A,\ b^1 = b,\ c^1 = c;\ r = 1;\ \mathcal{IS}_1 = \{x \in \{0,1\}^n : Ax \geq b\}$; let $F_1 = \{1, \ldots, n\}$ be the set of variables that has not yet been fixed.
2. **Addition of valid inequalities** : Construct a valid inequality $\sum_{i \in F_r} \alpha_i^r x_i \geq \beta^r$ for the convex hull of solutions in $\mathcal{IS}_r$;
   Set
$$\begin{aligned} y_r &\leftarrow \min\{\frac{c_i^r}{\alpha_i^r} : \alpha_i^r > 0\}, \\ k(r) &\leftarrow argmin\{\frac{c_i^r}{\alpha_i^r} : \alpha_i^r > 0\}. \end{aligned}$$

3. **Problem modification** : Set $\overline{x}_{k(r)} = 1$; $F_{r+1} = F_r \setminus \{k(r)\}$;
Delete the column $A^r_{k(r)}$ corresponding to $x_{k(r)}$, i.e., set $A^{r+1} = A^r \setminus A^r_{k(r)}$;
set $b^{r+1} = b^r - A^r_{k(r)}$; set $c^{r+1} = c^r - y_r \alpha^r$,
set $\mathcal{IS}_{r+1} = \{x \in \{0,1\}^{n-r} : \; A^{r+1}x \geq b^{r+1}\}$;
let $(\mathcal{IP}_{r+1})$: min $c^{r+1}x$ *st.* $x \in \mathcal{IS}_{r+1}$ be the current problem instance;
Set $r \leftarrow r+1$ and repeat Step 2 until the solution obtained is feasible to the original problem, else conclude that the problem is infeasible.

4. **Reverse deletion** : Consider the variables selected in each step $x_{k(1)}, x_{k(2)}, \ldots, x_{k(t)}$, in that order. Let $\mathcal{C}_t = \{x_{k(t)}\}$. For r from $t-1$ to 1, in reverse order,

   - Set $\mathcal{C}_r \leftarrow \{x_{k(r)}\} \cup \mathcal{C}_{r+1}$.
   - Delete $x_{k(r)}$ if $\mathcal{C}_r \setminus \{x_{k(r)}\}$ corresponds to a minimal feasible solution to problem instance $\mathcal{IP}_r$.

5. Set $x^H_i = 1$ if $x_i \in \mathcal{C}_1$. Return x^H. Let $Z_H = cx^H$.

Remarks:

1. Another way to understand the reverse deletion process is to delete the variables $x_{k(1)}, x_{k(2)}, \ldots, x_{k(t)}$ in reverse order, while maintaining feasibility to the **original** problem instance (IP), which is the same as $\mathcal{IP}_1$. This observation is particularly useful in implementing the above heuristic.

2. If n is the dimension of Problem (IP) the running time of Algorithm $\mathcal{PD}$ is $O(n(n + C(n))$, where $C(n)$ is the time to check feasibility of an instance of (IP) of size n. There are at most n stages for Steps 2 and 3. The work per stage is $O(n)$. In the reverse deletion step we need to check feasibility at most n times in order to ensure minimality.

Note that we have not specified the valid inequalities to be used at each stage of the primal-dual algorithm $\mathcal{PD}$. The performance of the algorithm depends critically on the choice of the inequalities. In order to analyze the algorithm, we introduce the notion of strength of the inequalities used.

For ease of presentation, if x_i does not appear in the current problem instance, we set $\alpha^r_i = 0$. This is to maintain the same dimensionality throughout for all the inequalities used. We also write α for the vector corresponding to $(\alpha_1, \alpha_2, \ldots, \alpha_n)$.

Definition 2.1 *The strength* $\lambda_r = s(\alpha^r, \beta^r)$ *of the inequality* $\sum_i \alpha_i^r x_i \geq \beta^r$ *with respect to instance* $\mathcal{IP}_r$ *is defined to be*

$$\lambda_r = \max\{\frac{\sum_i \alpha_i^r y_i}{\beta^r} : \ y \text{ minimal integral solution for } \mathcal{IP}_r\}.$$

In order to bound the performance of the primal-dual algorithm let

$$(LP_{PD}) \ Z_D = \min\{cx : \alpha^r x \geq \beta^r, r = 1, 2, \ldots, t, x \geq 0\}.$$

and $\lambda = \max_{r=1,\ldots,t} \lambda_r$.

Theorem 2.2 *The solution* x_H *is a feasible minimal solution to Problem* (IP)*, the vector* $y = (y_1, \ldots y_t)$ *is a feasible dual solution to Problem* (LP_{PD}) *and*

$$Z_H = cx^H \leq \lambda \sum_{r=1}^{t} y_r \beta^r \leq \lambda Z_D.$$

In particular,

a. $Z_H \leq \lambda I Z$.

b. Moreover, if all the inequalities $\alpha^r x \geq \beta^r)$ *are redundant inequalities for* $Ax \geq b, x \geq 0$*, then* $Z_H \leq \lambda Z$.

Proof:
Let $x_{k(r)}$ be the variable selected in the rth stage of the algorithm (note that $x_{k(r)}$ need not be the same as $x^H_{k(r)}$, since $x_{k(r)}$ might have been deleted in the reverse deletion step). Let $x^{(r)}$ be obtained from x_H by setting $x_{k(1)}, \ldots, x_{k(r-1)}$ to 0. By construction x^r is a minimal solution to $\mathcal{IP}_r$. We first prove by induction that for every $r = t$ to 1:

$$c^r x^r \leq \lambda \sum_{i \geq r} y_i \beta^i. \tag{1}$$

For $r = t$, since x^t is a minimal solution to $\mathcal{IP}_t$ and by the definition of strength

$$\alpha^t x^t \leq \lambda \beta^t,$$

which implies that

$$c^t x^t = y_t \alpha^t x^t \leq \lambda y_t \beta^t.$$

Assuming that the induction hypothesis holds for all $k \geq r+1$, we obtain (by the way we update the cost vectors) that

$$c^r x^r = [c^{r+1} + y_r \alpha^r] x^r,$$

Since $c^{r+1}_{k(r)} = 0$,

$$c^r x^r = c^{r+1} x^{r+1} + y_r \alpha^r x^r.$$

Applying the induction hypothesis and using $\alpha^r x^r \leq \lambda \beta^r$ by the definition of strength and the minimality of x^r, as well as $y_r \geq 0$ by construction, we obtain (1).

We next prove by induction that for every $r = t$ to 1:

$$\sum_{j \geq r} y_j \ \alpha^j_i \leq c^r_i, \ i = 1, \ldots, m. \tag{2}$$

For $r = t$, $y_t \ \alpha^t_i \leq c^t_i$, which follows since by construction

$$y_t = \min_{i:\ \alpha^t_i > 0} \{\frac{c^t_i}{\alpha^t_i}\}.$$

Assuming (2) holds for $r+1$, then

$$\sum_{j \geq r} y_j \ \alpha^j_i = \sum_{j \geq r+1} y_j \ \alpha^j_i + y_r \ \alpha^r_i \leq c^{r+1}_i + y_r \ \alpha^r_i = c^r_i,$$

where the last equality holds from the way the cost vector is updated, proving (2). As an additional remark, note that for $i = k(r)$ (2) holds as equality since $\alpha^j_{k(r)} = 0$ for each $j > k(r)$, since $x_{k(r)}$ does not appear in the subsequent problems.
Therefore, $\{y_j\}_{j \geq 1}$ forms a dual feasible solution to dual of the relaxation

$$Z_D = \min\{cx : \alpha^i x \geq \beta^i, i = 1, 2, \ldots, t, x \geq 0\}.$$

Letting $Z_D = \sum_{i \geq 1} \ y_i \beta^i$, be the value of this dual feasible solution we obtain

$$Z_H \leq \sum_r y_r \beta^r \leq \lambda Z_D \leq \lambda IZ.$$

If in addition, all the inequalities (α^r, β^r) are redundant to $Ax \geq b, x \geq 0$, then $Z_H \leq \lambda Z$.
□

By the previous theorem, we have reduced the construction of a λ-approximation algorithm to one of finding valid inequalities with strength bounded by λ. Since there are cases where more than one such inequalities exist, each inequality suggest a different primal-dual approximation algorithm, all attaining the same bound λ.

2.1 A geometric view of the primal-dual algorithm

Let us first develop some geometric insight on the strength of an inequality. Let $CH(IP)$ denote the convex hull of all minimal integral solutions to problem (IP). Let $\alpha x \geq \beta$

denote a valid inequality for $CH(IP)$, touching $CH(IP)$ at a vertex x_1 (see Figure 1). It corresponds to a hyperplane with all the vertices of $CH(IP)$ on one side. Let λ denote the strength of this inequality with respect to (IP). By the definition of λ, the vertices of $CH(IP)$ are "sandwiched" between the hyperplane $\alpha x = \beta$ and $\alpha x = \lambda\beta$. A valid inequality that gives us the "thinnest" slab sandwiching the vertices of $CH(IP)$ will thus result in the best bound in terms of strength. This geometric view enables us to show next that the bound of Theorem 2.2 is essentially tight.

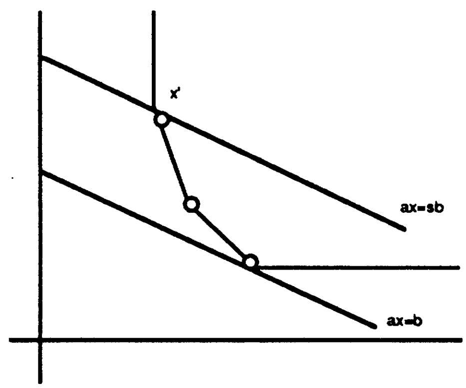

Figure 1: Geometry of strength of an inequality

Theorem 2.3 *Assume that the first valid inequality $\alpha x \geq \beta$ we introduce in Algorithm $\mathcal{PD}$ achieves the maximum strength λ. Then, for all $\epsilon > 0$ there exists a cost vector such that Algorithm $\mathcal{PD}$ outputs a solution x_H with cost*

$$Z_H \geq \lambda(1-\epsilon)IZ.$$

Proof: Let x' be a minimal solution with $\alpha x' = \max\{\alpha x : x \text{ minimal in } (IP)\} = \lambda\beta$. Let C denote the set of indices k with $x'_k = 1$. For each $k \in C$, set $c_k = \alpha_k$. Set $c_k = \alpha_k + \gamma$ for all $k \notin C$, with $\gamma > 0$. By this choice of cost function c, the reduced cost at all x_i, $i \in C$, are 0 after the first step. Thus the algorithm will always return the solution x', with objective value

$$Z_H = \alpha x' + \gamma \sum_{i \notin C} x'_i \geq \lambda\beta.$$

Moreover, $IZ \leq cx_1$, where x_1 is a vertex in $CH(IP)$ with $\alpha x_1 = \beta$. Therefore,

$$IZ \leq \alpha x_1 + \gamma \sum_{i \notin C} x_{1,i} \leq \beta + \gamma n.$$

By choosing $\gamma = \frac{\epsilon\beta}{n}$, we can ensure that under c

$$\frac{Z_H}{IZ} \geq \frac{\lambda\beta}{\beta + \gamma n} = \frac{\lambda}{1+\epsilon} \geq \lambda(1-\epsilon).$$

□

Remarks:

1. The previous theorem illustrates that the notion of strength is inherent in the primal dual approach and not an artifact of the analysis.

2. The fundamental reason the bound is essentially tight is that the cost function is not incorporated in the design of Algorithm $\mathcal{PD}$. In other words, the valid inequalities used in each stage of the algorithm are independent of the cost function. The previous theorem shows that in order to obtain a better bound within a primal-dual framework, we need to take into account the cost function in the choice of valid inequalities.

3. In the next section we apply Algorithm $\mathcal{PD}$ in many problems. In all these applications the maximum strength is attained at the first stage. Therefore, the bounds attained for the respective problems are essentially tight. This eliminates the need to construct problem specific examples that attain the bound.

3 Reducible formulations and approximability

In this section we illustrate the power of Theorem 2.2 by showing that the best known results in approximation algorithms for covering problems are special cases of Theorem 2.2. Theorem 2.2 reduces the construction of good approximation algorithms to the design of valid inequalities of small strength. At first sight it appears difficult to bound the maximum strength of a class of inequalities, since we need to bound the strength of each inequality we add with respect to a new, each time, problem instance. We next illustrate that for a rather rich class of formulations bounding the strength can be greatly simplified.

3.1 Reducible formulations

We consider covering problems of the form:

$$(IP_n) \ \ IZ_n = \min \ cx$$

$$\text{subject to} \ \ Ax \geq b$$

$$x \in \{0,1\}^n,$$

where A is an $m \times n$ matrix and c is an n-vector with nonnegative integer entries; entries in b are integral but not restricted to be nonnegative, since rows corresponding to negative b_i are redundant. Note that we have explicitly stated the dependence on the problem size n. We assume that formulation (IP_n) models problems from a problem class $\mathcal{C}$. By fixing variable x_j to 1, we create the following problem:

$$(IP^j_{n-1}) \quad IZ^j_{n-1} = \min \ \bar{c}\bar{x}$$

$$\text{subject to} \quad \bar{A}\bar{x} \geq b - A_j$$

$$\bar{x} \in \{0,1\}^{n-1},$$

where $\bar{A}$ is an $m \times (n-1)$ matrix.

Definition 3.1 *Formulation (IP_n) is reducible with respect to problem class $\mathcal{C}$ if for all j, formulation (IP^j_{n-1}) belongs to problem class $\mathcal{C}$.*

In other words, reducible formulations of a problem with respect to a problem class $\mathcal{C}$ have the property that the new smaller instance that results by fixing a variable, still belongs in problem class $\mathcal{C}$. The importance of reducible formulations in the context of the primal dual algorithm $\mathcal{PD}$ is that, we can bound the strength of an inequality with respect to the original problem's instance, since by the definition of a reducible formulation even after fixing a variable, the problem instance belongs in the same class. Therefore, given a reducible covering formulation, there is no need to calculate the strength of a given inequality with respect to an instance generated in the course of the primal-dual algorithm. Since by reducibility all the instances belong in the same class, it suffices to calculate the strength with respect to the original instance. This greatly simplifies the calculation of strength as we show next.

3.2 General Covering Problems

Consider the problem

$$(GC) \quad IZ_{GC} = \min \ cx$$

$$\text{subject to} \quad Ax \geq b$$

$$x \in \{0,1\}^n,$$

where a_{ij} and c_j are nonnegative integers. Fixing some variable x_j to 1, results in a new instance that still has the property that the matrix $\bar{A}$ and the vector $\bar{c}$ are nonnegative integers. Thus, formulation (GC) is reducible with respect to the class of general covering problems.

Hall and Hochbaum [11] proposed a dual heuristic for the case when a_{ij} are 0 or 1, with $Z_H(GC) \leq f Z_{GC}$, $f = \max_i \sum_{j=1}^n a_{ij}$. We refer to this bound as the row-sum bound. Bertsimas and Vohra [3] proved that the same bound holds with general values of a_{ij}. We next show that algorithm $\mathcal{PD}$ produces the same bound for problem (GC).

Theorem 3.2 *The strength of the inequalities $a_i x \geq b_i$, $i = 1, \ldots, m$ is at most f, i.e., Algorithm $\mathcal{PD}$ applied to these inequalities produces a solution such that*

$$\frac{Z_H}{Z_{GC}} \leq f. \tag{3}$$

Proof: Consider an inequality $a_i x \geq b_i$. Let x' be a minimal solution to (GC). Clearly $a_i x' \leq f$. Therefore, $\lambda_i \leq \frac{f}{b_i} \leq f$. Since the row sum reduces after each step of the algorithm, the strength of all inequalities is bounded above by f. Therefore, from Theorem 2.2 (3) follows. □

3.3 The Minimum Spanning Tree Problem

Let G denote an undirected graph on the vertex set V and edge set E. The minimum spanning tree (MST) problem asks for a spanning tree that minimizes a given nonnegative objective function c. Since c is nonnegative, we can solve the problem by the following cut-formulation

$$\begin{aligned} (CUT) \quad IZ_{CUT} = \min \quad & cx \\ \text{subject to} \quad & \sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \subset V, \\ & x_e \in \{0, 1\}. \end{aligned}$$

By fixing some x_e to be 1, we obtain the cut formulation for the MST on $|V| - 1$ nodes on the multigraph created by contracting the edge $e = (i, j)$ (combining i, j into a supernode a and adding an edge (a, k) whenever $(i, k) \in E$ or $(j, k) \in E$). Thus formulation (CUT) is reducible with respect to the MST problem.

By adding the multicut constraints, first suggested by Fulkerson [7], we arrive at the multicut formulation:

$$(MCUT) \quad IZ_{MCUT} = \min \quad cx$$

$$\text{subject to} \quad \sum_{e\in\delta(S_1,\ldots,S_k)} x_e \geq k-1, \ \forall (S_1,\ldots,S_k) \text{ partitioning of } V,$$
$$x_e \in \{0,1\}.$$

Fixing $x_e = 1$ in $(MCUT)$ we again arrive at a multicut formulation for G contracted at the edge e. Thus the multicut formulation is reducible. The LP relaxation of this formulation gives the complete characterization of the dominant of the spanning tree polytope (see [4]). By applying Theorem 2.2 we provide a genuinely simple proof of the integrality of the multicut polyhedron, as well as the known tight bound on the duality gap of the IZ_{CUT} and Z_{CUT}.

Theorem 3.3 *The inequality $\sum_{e\in E} x_e \geq n-1$ is valid for the multicut polyhedron and has strength* 1, *i.e.,*

$$IZ_{MCUT} = Z_{MCUT}. \tag{4}$$

The inequality $\sum_{e\in E} x_e \geq \frac{n}{2}$ is valid for the cut polyhedron and has strength $2(1-\frac{1}{n})$, *i.e.,*

$$\frac{Z_H}{Z_{CUT}} \leq 2(1-\frac{1}{n}). \tag{5}$$

Proof: We first consider the multicut formulation $(MCUT)$. Since $\sum_e x_e \geq n-1$ is a valid inequality (consider a partition of V into the nodes) and all minimal solutions, being trees, have at most $n-1$ edges, the strength of this inequality is 1. By using inequalities of this type in each stage of the algorithm, we obtain an optimal integral solution to the spanning tree problem, thus showing (4).

We next consider the cutset formulation (CUT). Since $\sum_e x_e \geq \frac{n}{2}$ is a valid inequality (add all the cut inequalities for singletons), the strength is $2(1-1/n)$, thus showing (5). The bound obtained is again tight. □

Remark: Algorithm $\mathcal{PD}$ applied to the multicut formulation corresponds to the classical Kruskal Algorithm.

3.4 The Shortest Path Problem

Let s,t be two distinct vertices in an undirected graph G. The problem of finding the shortest path from s to t can be modelled as an edge-covering formulation

$$(SP) \quad IZ_{SP} = \min \quad cx$$
$$\text{subject to} \quad \sum_{e\in\delta(S)} x_e \geq 1, \ \forall S : s \in S \textit{ or } t \in S,$$
$$x_e \in \{0,1\}.$$

It is easy to observe that formulation (SP) is again reducible. In this case, the following theorem is immediate

Theorem 3.4 *Inequalities*

1. $x(\delta(s)) \geq 1$,

2. $x(\delta(t)) \geq 1$, *and*

3. $x(\delta(s)) + x(\delta(t)) \geq 2$.

have strength 1, *i.e.,*

$$IZ_{SP} = Z_{SP}.$$

Using any of these inequalities in each stage of our primal-dual approach, we would have obtained an optimal shortest path solution. Each choice of the inequalities gives rise to the (1) forward Dijkstra, (2) backward Dijkstra and (3) bidirectional Dijkstra algorithm respectively. Our analysis indicates that one can in fact use any of the three inequalities at each stage of the algorithm.

3.5 Uncrossable functions

Consider the following edge-covering problem introduced in Goemans and Williamson [8]:

$$\begin{array}{lll} (UC) & IZ_{UC} = \min & \sum_e c_e x_e \\ & \text{subject to} & \sum_S x(\delta(S)) \geq f(S), S \subset V, \\ & & x_e \in \{0,1\}, \end{array}$$

where the function f defined on 2^V is a symmetric 0-1 function, $f(V) = 0$, and f satisfies further the following uncrossability property:

- if S, T are intersecting sets with $f(S) = f(T) = 1$, then either $f(S-T) = 1, f(T-S) = 1$ or $f(S \cap T) = f(S \cup T) = 1$.

A 2-approximation algorithm for this class of problem was first proposed by Williamson et. al. [20]. It generalized an earlier algorithm [8] designed for a more restrictive 0-1 function f such that

$$f(S \cup T) \leq \max\{f(S), f(T)\}, \tag{6}$$

for all disjoint S and T, and f symmetric. Symmetric functions f satisfying (6) are called proper functions. Note that the conditions for properness imply uncrossability. We refer the readers to Goemans and Williamson [8] for a long list of problems that can be modelled as edge-covering problems with 0-1 proper functions f (note that formulations (CUT) and (SP) for the minimum spanning tree and the shortest path belong in this class). The edge-covering formulations are reducible with respect to both 0-1 uncrossable functions and proper functions. By fixing an edge x_e to 1, we see that the cut condition for all S containing e in the cut set is satisfied. Hence the problem reduces to an edge-covering problem on G contracted at e (denoted by $G \circ e$). The corresponding function f on $G \circ e$ inherits the uncrossability (or respectively properness) property.

In this section we exhibit valid inequalities for (UC) of strength at most 2. While a proof of the next theorem can be extracted from [20], we offer a new self-contained proof.

Theorem 3.5 *Let $\{S_1, \ldots, S_l\}$ denote a* maximal *collection of disjoint subsets S_j with $f(S_j) = 1$ for all S_j, and $f(T) = 0$ if $T \subset S_j$ for some j. The strength of the inequality*

$$\sum_{j=1}^{l} \sum_{e \in \delta(S_j)} x_e \geq l \tag{7}$$

is $2(1 - \frac{1}{l})$, i.e.,

$$\frac{Z_H}{Z_{UC}} \leq 2(1 - \frac{1}{l}).$$

Proof: Let F denote a minimal set of edges corresponding to a feasible solution and let $G[F]$ the graph induced by the set of edges F. It suffices to prove that

$$\sum_{j=1}^{l} x_F(\delta(S_j)) \leq 2(l-1) \tag{8}$$

Note that the coefficients of edges in $\delta(S_i, S_j)$ are 2 whereas those between $\delta(S_i, V - \cup_j S_j)$ are 1.

Let $U = V \setminus \{S_1, \ldots, S_j\}$. Let $T_1, \ldots, T_m$ denote the connected components in U under F. Let G' denote the new graph obtained from $G[F]$ by treating all S_j's and T_k's as nodes. Let f' be the function induced on G' by f. Clearly f' is also uncrossable and symmetric, and $F' = F \cap E(G')$ is again a minimal solution with respect to f'. F' consists of all the edges counted in (8). Note that this construction need not necessarily reduce the size of the graph. If none of the nodes T_j has degree 1 in G', then (8) follows immediately from the forest structure of F'. So we may assume that $\deg(T_1) = 1$, and the edge e connect T_1 to the vertex S_1.

We will use induction on the number of nodes in G' to compute (8). To do so, we will contract a suitable subgraph of G' of size at least 2.

Case 1 : If deg(S_1) is also 1, then $f'(\{S_1, T_1\}) = 0$. Contract the graph at the component $\{S_1, T_1\}$. If there is no set S containing the component $\{S_1, T_1\}$ but not S_i for $i \geq 2$, with $f'(S) = 1$, then the number of disjoint minimal sets in the contracted graph reduced to $l - 1$. Using induction on the number of nodes, the contribution by the rest of the edges of F' to (8) is at most $2(l-2)$. Counting e, we have

$$\sum_{j=1}^{l} x_F(\delta(S_j)) \leq 2(l-2) + 1 \leq 2(l-1).$$

If a set S with the above property exists, then the number of disjoint minimal sets for the contracted graph remains at l, but there must be an edge e' incident to S and some S_i, $i \geq 2$. This edge will be counted twice in this contracted instance under the induction hypothesis, whereas its contribution to (8) is 1. So we have

$$\sum_{j=1}^{l} x_F(\delta(S_j)) \leq \{2(l-1) - 1\} + 1 = 2(l-1).$$

Case 2 : Suppose $\deg_{G'}(S_1) \geq 2$. By minimality of F', there exists a set W in the vertex set of G' such that $\delta(W) = \{e\}$, $f'(W) = 1$ and $S_1 \subset W$. By symmetry, $f'(\overline{W}) = 1$. Thus $|W| \geq 2, |\overline{W}| \geq 2$. Let $G_W, G_{\overline{W}}$ denote respectively the graph obtained from G' by contracting $\overline{W}$ and W into a single node. These are minimal solutions with respect to f' restricted to the vertex sets of G_W and $G_{\overline{W}}$. Let $l_W, l_{\overline{W}}$ denote the number of S_i's contained in W and $\overline{W}$ respectively. By our modification, the number of disjoint minimal sets in G_W and $G_{\overline{W}}$ are $l_W + 1$ and $l_{\overline{W}} + 1$ respectively. Using induction on the number of nodes, the contribution of edges in G_W and $G_{\overline{W}}$ to (8) are at most $2l_W$ and $2l_{\overline{W}}$ respectively. Note that the edge $e = (S_1, T_1)$ has been counted thrice, once in $G_{\overline{W}}$ and twice in G_W, whereas its contribution to (8) is 1. Therefore,

$$\begin{aligned}\sum_{j=1}^{l} x_F(\delta(S_j)) &\leq 2l_W + 2l_{\overline{W}} - 2 \\ &= 2(l-1).\end{aligned}$$

Therefore the theorem holds. □

A direct corollary of the analysis in the previous theorem is the observation that the strength of

$$\sum_{j=1, j\neq k}^{l} x(\delta(S_j)) \geq l - 1$$

and

$$\sum_{j=1,j\neq k_1,j\neq k_2}^{l} x(\delta(S_j)) \geq l-2$$

are $2-\frac{1}{l-1}$ and 2 respectively. Using these inequalities in Algorithm $\mathcal{PD}$ leads to an approximation algorithm with bound not worse than 2.

So far we have not indicated how one could find the minimal sets S_i's used in the construction of the inequality. If f is proper, then the sets S_i's are simply all the nodes v with $f(v)=1$, and thus we could implement the primal-dual algorithm in polynomial time. For the case of uncrossable function, the question is still open.

3.6 Constrained Contra-Polymatroids

Consider the problem

$$\begin{aligned}
(CP) \quad IZ_{CP} = \min \quad & \sum_i c_i x_i \\
\text{subject to} \quad & x(S) = \sum_{i\in S} x_i \geq f(S), S \subset N = \{1,...,n\}, \\
& x_i \in \{0,1\}.
\end{aligned}$$

where f satisfies $f(\emptyset)=0$ and

$$\begin{aligned}
f(S)+f(T) &\leq f(S\cap T)+f(S\cup T) \text{ (supermodular);} && (9)\\
f(S) &\leq f(T),\ \forall\ S\subset T. \text{ (nondecreasing),} && (10)
\end{aligned}$$

The function f is called a contra-polymatroid function (see [18]). Notice that we have the additional restriction that $x_i \in \{0,1\}$, giving rise to what we call a constrained contra-polymatroid problem.

If we set $x_i = 1$ and modify the constraints, we have a problem instance on $N\setminus\{i\}$, with $f'(S) = \max(f(S), f(S\cup i)-1)$ for all S in $N\setminus\{i\}$. Clearly $f'(S) \leq f'(T)$ if $S\subset T\subset N\setminus\{i\}$. To show supermodularity, suppose $f'(S)=f(S), f'(T)=f(T\cup i)-1$. Then

$$f'(S)+f'(T) \leq f(S\cap T)+f(S\cup\{T+i\})-1 \leq f'(S\cap T)+f'(S\cup T).$$

The other cases can be handled similarly. Thus f' is a contra-polymatroid function. The formulation is thus reducible.

Theorem 3.6 *The inequality*

$$\sum_i x_i \geq f(N)$$

has strength 1*, thus*

$$IZ_{CP} = Z_{CP}.$$

Proof: Let x' be a minimal solution. By minimality, there exists a set S_i with $f(S_i) = 1$ (called a tight set) containing each $x'_i = 1$. Hence

$$\begin{aligned} x'(S_i) + x'(S_j) &= f(S_i) + f(S_j) \\ &\leq f(S_i \cap S_j) + f(S_i \cup S_j) \\ &\leq x'(S_i \cap S_j) + x'(S_i \cup S_j) \\ &= x'(S_i) + x'(S_j). \end{aligned}$$

Hence $S_i \cup S_j$ is again tight. By repeating this procedure, we obtain $x'(N) = f(N)$. Hence the strength of the inequality is 1. The constrained contra-polymatroid polytope is thus integral. □

This analysis reveals that one can indeed remove the conditions that f is nondecreasing, and the LP formulation will still be tight. This is due to the presence of the inequalities of the form $x_i \leq 1$.

A direct generalization of this argument to the intersection of k constrained contra-polymatroids leads to the following theorem, which is, to the best of our knowledge, new:

Theorem 3.7 *The strength of the inequality* $\sum_i x_i \geq \frac{f_1(N)+\ldots+f_k(N)}{k}$ *for the intersection of* k *contra-polymatroids is* k*, i.e., Algorithm* $\mathcal{PD}$ *has a worst case bound of* k*.*

Remark: Although for $k = 2$ there exists a polynomial algorithm, Algorithm $\mathcal{PD}$ has a faster running time.

3.7 Set covering problems

In this section we consider special cases of the set covering problem:

$$\begin{aligned} (COVER) \quad IZ_{COVER} = \min \quad & cx \\ \text{subject to} \quad & Ax \geq 1 \\ & x \in \{0,1\}^n, \end{aligned}$$

where A is a 0-1 matrix. We show that the application of Theorem 2.2 in the following cases gives rather strong results.

1. Row-inclusion matrices (see [17]):

   A does not contain the submatrix $\begin{pmatrix} 11 \\ 01 \end{pmatrix}$; Row-inclusion matrices play an important role in the study of totally balanced matrices (see [17]). It is easy to verify that the covering formulation is reducible with respect to the row-inclusion property: by removing all redundant constraints after deleting the kth column from A, one obtains another constraint matrix with the row inclusion property. It is well known that the underlying polyhedron is integral. Surprisingly, we can prove it simply by applying Theorem 2.2.

   Theorem 3.8 *The strength of the first inequality* $a_{11}x_1 + \ldots + a_{1n}x_n \geq 1$ *is* 1.

   Proof: Consider a minimal solution x'. We will show that $a_{11}x'_1 + \ldots + a_{1n}x'_n \leq 1$. Assuming otherwise, then there exist i, j with

   $$a_{1i} = a_{1j} = x'_i = x'_j = 1$$

   and $i < j$. By the minimality of x', if we set x'_j to 0, then the solution is no longer feasible. Thus there must exist a row k such that $a_{ki} = 0$ and $a_{kj} = 1$. This however contradicts the fact that A is a row-inclusion matrix. Therefore, the inequality $a_{11}x_1 + \ldots + a_{1n}x_n \geq 1$ has strength 1, proving that Algorithm $\mathcal{PD}$ finds an optimal solution in this case. □

2. Matrices A with consecutive ones in columns.

   This class of matrices belongs to the class of totally unimodular matrices (see [17]) and therefore the underlying polyhedra are integral. There exists an optimal algorithm that transforms the problem to a shortest path problem. We show that Algorithm $\mathcal{PD}$ is a direct optimal algorithm for the problem.

   Theorem 3.9 *The strength of the first inequality* $a_{11}x_1 + \ldots + a_{1n}x_n \geq 1$ *is* 1.

   Proof: Consider a minimal solution x'. We will show that $a_{11}x'_1 + \ldots + a_{1n}x'_n \leq 1$. Assuming otherwise, then there exist i, j with

   $$a_{1i} = a_{1j} = x'_i = x'_j = 1$$

   and $i < j$. By the minimality of x', if we set x'_j to 0, then the solution is no longer feasible. Thus there must exist a row k such that $a_{ki} = 0$ and $a_{kj} = 1$. Symmetrically, there must exist a row l such that $a_{lj} = 0$ and $a_{li} = 1$, where $i < j$. Assuming $k < l$ (otherwise we consider the jth column), we have $a_{1i} = 1, a_{ki} = 0$ and $a_{li} = 1$, violating the consecutive ones property. Therefore, the inequality $a_{11}x_1 + \ldots + a_{1n}x_n \geq 1$ has strength 1, proving that Algorithm $\mathcal{PD}$ finds an optimal solution in this case. □

3. Arbitrary $0 - 1$ matrices A.
   A direct generalization of the previous argument yields:

   Theorem 3.10 *The first inequality $a_{11}x_1 + \ldots + a_{1n}x_n \geq 1$ has strength*

   $$\max_{i=1,2,\ldots,n} \{g_i + 1\},$$

   where g_i is the maximum gap between any 2 ones in the ith column.

   Obviously the previous bound can be optimized by considering permutations of the rows that

   $$\min_{\pi} \max_{i=1,2,\ldots,n} \{g_i(\pi) + 1\}.$$

4. Matrices A with consecutive ones in rows.
   We may assume without loss of generality that there is no redundant inequality in the constraints.

   Theorem 3.11 *Inequality $x_1 + x_2 + \ldots + x_L \geq 1$ has strength* 1.

   Proof: Let x' be a minimal solution. We show that $x'_1 + x'_2 + \ldots + x'_L \leq 1$. Assuming otherwise, suppose $x'_u = x'_v = 1$ for some $u < v \leq L$. Then by minimality, there exists a constraint $\sum_j a_{ij}x_j \geq 1$ with $a_{iu} = 1$ but $a_{iv} = 0$. By the consecutive ones property, this implies that the inequality $x_1 + x_2 + \ldots + x_L \geq 1$ is redundant, a contradiction. Therefore, inequality $x_1 + x_2 + \ldots + x_L \geq 1$ has strength 1. □

5. Matrices A with circular ones in rows. Again we may assume that there is no redundant inequality in the constraints. By similar reasoning as in the previous case, we can show

   Theorem 3.12 *Every constraint in $Ax \geq 1$ has strength at most 2.*

4 Multiphase extension of the primal-dual algorithm

In this section we propose an extension of Algorithm $\mathcal{PD}$ to problem (IP) that uses the primal-dual approach in phases. Let $b_{max} = \max_i b_i$.
Multiphase Primal Dual Algorithm $\mathcal{MFPD}$

- Input : A, b, c, $(A, c \geq 0)$.
- Output : x_H feasible for (IP) or conclude that the problem is infeasible.

1. **Initialization:** $k = 1$.
2. **Phase step k** : Let $h_i = 1$ if $b_i = b_{max}$ and $h_i = 0$ otherwise.
   Delete redundant rows from $Ax \geq h$ (resulting in $A'x \geq 1$) and apply Algorithm $\mathcal{PD}$ (using the same inequalities $A'x \geq 1$) to the problem

$$
\begin{aligned}
IZ_k = \min\ & cx \\
\text{subject to}\ \ & A'x \geq 1 \\
& x \in \{0,1\}^n.
\end{aligned}
$$

   yielding a solution x_k of cost $Z_H^k = cx_k$. Let Z_k denote the LP relaxation where we substitute constraints $x \in \{0,1\}^n$ with $x \in [0,1]^n$.
   $J_k = \{j : \ x_{k,j} = 1\}$; $A := A \setminus \{A_j\}_{j \in J_k}$, i.e., delete the columns of A corresponding to the indices in set J_k; $b := b - \sum_{j \in J_k} A_j$; $c := c \setminus \{c_j\}_{j \in J_k}$; $k := k + 1$.
3. Repeat Step 2 until a feasible solution is found. The feasible solution is $x_j = 1$ for all $j \in \cup_k J_k$. If after $\min(b_{max}, n)$ phases a feasible solution is not found conclude that the problem is infeasible.

Let x_H be the solution of obtained by Algorithm $\mathcal{MFPD}$ and Z_H its cost. In the next theorem we bound the performance of the algorithm.

Theorem 4.1 *1. If at each phase k the worst case bound for Algorithm $\mathcal{PD}$ is $Z_H^k \leq \lambda Z_k$, then*

$$\frac{Z_H}{Z} \leq \lambda\ \mathcal{H}(b_{max}), \tag{11}$$

where $\mathcal{H}(n) = \sum_{i=1}^{n} \frac{1}{i}$.

2. If at each phase k the worst case bound for Algorithm $\mathcal{PD}$ is $Z_H^k \leq \lambda I Z_k$, then

$$\frac{Z_H}{IZ} \leq \lambda\ b_{max}, \tag{12}$$

Proof: We prove the theorem by induction on b_{max}. For $b_{max} = 1$, Algorithm $\mathcal{MFPD}$ reduces to Algorithm $\mathcal{PD}$ and (11) follows from the assumed bound on the performance of Algorithm $\mathcal{PD}$. Assuming (11) is true for $b_{max} - 1$, we prove it for b_{max}. For ease of exposition we introduce the notation:

$$P(b,c) \quad Z(b,c) = \min cx$$

$$\text{subject to } Ax \geq b$$

$$x \in [0,1]^n.$$

We denote the corresponding optimal solution $x^*(b,c)$. We also denote with $IZ(b,c)$ the value of the corresponding $0-1$ problem. After the first phase of Algorithm $\mathcal{MFPD}$ the solution x_1 produced has cost

$$\sum_{j \in J_1} c_j \leq \lambda \ Z_{h,c} \leq \lambda \frac{Z_{b,c}}{b_{\max}},$$

because the solution $\frac{x^*(b,c)}{b_{max}}$ is feasible for the problem $P(h,c)$. The cost function for the next stage is $c'_j = c_j$ for j not in J_1. Although the variables x_j with $j \in J_1$ are not present in the next phase, we prefer to set $c'_j = 0$ for $j \in J_1$. By this slight abuse of notation, we can view c' as the cost function for the second phase of the algorithm. Clearly,

$$Z_{b',c'} \leq Z_{b,c'} \leq Z_{b,c}.$$

Since b'_{max} is at most $b_{max} - 1$ in the next phase we can invoke the induction hypothesis to assert that the solution x'_H (with $J'_H = \{j : x'_{H,j} = 1\}$) that the Algorithm $\mathcal{MFPD}$ returns has cost

$$\sum_{j \in J'_H} c_j \leq \lambda \mathcal{H}(b_{\max} - 1) Z_{b',c'} \leq \lambda \mathcal{H}(b_{\max} - 1) Z_{b,c}.$$

The superposition of the solutions x_1 and x'_H with support $J_1 \cup J'_H$ is the solution produced by Algorithm $\mathcal{MFPD}$ on the original input has cost

$$Z_H = \sum_{j \in J_1 \cup J'_H} c_j \leq \lambda(\mathcal{H}(b_{\max} - 1) + \frac{1}{b_{max}}) Z_{b,c} = \lambda \mathcal{H}(b_{\max}) Z_{b,c},$$

proving (11).

When the value of the heuristic is within λ from the optimal integer solution, the proof is identical except that we can only guarantee

$$\sum_{j \in J_1} c_j \leq \lambda \ IZ_{h,c} \leq \lambda \ IZ_{b,c}.$$

The induction on b_{max} proceeds along the same lines except that

$$Z_H = \sum_{j \in J_1 \cup J'_H} c_j \leq \lambda(b_{\max} - 1) I Z_{b,c} + \lambda I Z_{b,c} = \lambda b_{\max} I Z_{b,c}.$$

$\square$

4.1 Applications

In this section we outline a number of applications of Theorem 4.1. All of these applications are special cases of formulation (IP).

1. Matrix A with consecutive ones in columns (or rows), b arbitrary.
   At each phase of Algorithm $\mathcal{MFPD}$, columns from matrix A and redundant constraints are deleted; therefore we obtain $A'x \geq 1$, where A' has again the consecutive ones property. Therefore, at each phase Theorem 3.9 (respectively 3.11) with $\lambda = 1$. Applying Theorem 4.1 Algorithm $\mathcal{MFPD}$ leads to a solution x_H with

   $$\frac{Z_H}{Z} \leq \mathcal{H}(b_{\max}).$$

   In contrast, the known optimal algorithm for the problem transforms the problem to a min-cost flow algorithm, at the expense of doubling the problem size.

2. Matrix A satisfies the row-inclusion property, b is arbitrary.
   In this case by exactly the same argument leads to

   $$\frac{Z_H}{Z} \leq \mathcal{H}(b_{\max}).$$

3. A, b arbitrary.
   From Theorem 3.10 $\lambda = \min_\pi \max_{i=1,2,\ldots,n}\{g_i + 1\}$, leading to

   $$\frac{Z_H}{Z} \leq \min_\pi \max_{i=1,2,\ldots,n}\{g_i + 1\}\mathcal{H}(b_{\max}).$$

   Notice that to the best of our knowledge this a new result, which can be substantially better that both the max-row sum bound (Theorem 3.2) as well as $\mathcal{H}(\max_j \sum_i A_{ji})$ proposed in [6].

4. Cut covering problems with weakly supermodular functions.
   For functions f taking values over integers, the notion of an uncrossable function considered in Section 2 has been generalized in [9] to the notion of a weakly supermodular

function, defined as a symmetric function f with

$$f(S) + f(T) \leq \max\{f(S-T) + f(T-S), f(S \cup T) + f(S \cap T)\}. \tag{13}$$

If f satisfies the stronger property (6), then f is called proper. Again weakly supermodular functions encompass the class of proper functions. Moreover, the edge-covering formulation is reducible with respect to weakly-supermodular functions : Let F be the set of edges fixed to 1, then $f(S) - \sum_{e \in F \cap \delta(S)} x_e$ is weakly supermodular. However, the formulation is not reducible with respect to arbitrary proper functions (although it is for $0-1$ proper functions). These observations underscore an important advantage of the notion of reducible formulations: By considering a wider class of problems (weakly supermodular functions), we simplify the analysis for a more restrictive class of problems (proper functions).

Theorem 4.1 immediately applies to derive approximation algorithm for cut covering problems with weakly supermodular function f, first obtained in [9] using considerably more complicated proof methods.

Theorem 4.2 *([9]) Algorithm* **MFPD** *is a* $2\mathcal{H}\ (f_{\max})$ *approximation algorithm for cut covering problems with weakly supermodular functions, where* $f_{max} = \max_S f(S)$.

Proof: : Letting $h(S) = 1$ if $f(S) = f_{\max}$, $h(S) = 0$ otherwise, implies $h(S)$ is a symmetric uncrossable function, since $f(S)$ is weakly supermodular. Since $f(S) - \sum_{e \in F \cap \delta(S)} x_e$ is still weakly supermodular, the formulation is reducible with respect to weakly supermodular functions and therefore, Theorem 4.1 applies with $\lambda = 2$ (for uncrossable functions, Theorem 2.2) leading to

$$\frac{Z_H}{Z} \leq 2\mathcal{H}\ (f_{\max}).$$

□

Remarks:

(a) In comparison with the proof methods used in [9], we believe that the inductive proof method in Theorem 4.1 is considerably simpler.

(b) When f is proper, there is a polynomial time procedure to construct the minimal sets used in the construction of the valid inequalities (see [20]). The case for weakly supermodular function is again open.

5. More general cut covering problems

We consider next an extension of the edge covering problem (also considered in [9]), in which a_e copies of the edge e are to be used if we decide to include the edge e in the solution. We assume $a_e > 0$. This leads to the following formulation :

$$
\begin{aligned}
(MU) \min \sum_e c_e\, x_e & \\
st \sum_{e \in \delta(S)} a_e x_e \;&\geq\; f(S), \quad S \subset V \\
x_e \in \{0,1\}, &
\end{aligned}
$$

where f is again weakly supermodular.

Note that the LP relaxation of the above formulation could be arbitrarily bad, due to the presence of a_e in the constraint matrix. In the case when f is 0-1, then the set of integral solution remains the same even if we set all a_e to 1, corresponding to the cut covering problem described in the previous section. Thus there is an approximation algorithm, which returns a solution not worse than 2 time of the optimal **integral** solution. The reason that the result does not hold for the optimal LP solution is because the valid inequalities used are not redundant. Given that the formulation is still reducible, we use (12) and obtain a bound of

$$\frac{Z_H}{IZ_{MU}} \leq 2 f_{max},$$

which is also the bound obtained in [9].

5 Conclusions

By showing a general max-min bound (strength) provided by the greedy type primal dual algorithm, we have unified a large part of combinatorial optimization under a single framework and reduced the *analysis* of approximation algorithms to computing the strength of inequalities. This approach also offers insights as to why certain algorithms achieve the stipulated performance bounds, and reduces the *design* of greedy type algorithms to the construction of valid inequalities with small strength. A direction for further research is to incorporate other nongreedy type approximation algorithms into a single framework with the goal of offering insights into the design of robust algorithms.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. Proof Verification and hardness of approximation problems. In *Proceedings of the 33rd Annual Symposium on Foundation of Computer Science,* page 14-23, 1992.

[2] A. Balakrishnan, T. Magnanti and P. Mirchandani. Modelling and Heuristic worst-case performance analysis of the two level network design problem. To appear in *Management Science.*

[3] D. Bertsimas and R. Vohra. Linear programming relaxations, approximation algorithms and randomization : a unified view of covering problems. Operations Research Center, MIT working paper, 1994.

[4] S. Chopra. On the spanning tree polyhedron. Graduate School of Business Administration, New York University, 1988.

[5] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem, technical report GSIA, Carnegie-Mellon Univ., 1976.

[6] G. Dobson, Worst-Case Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data, *Mathematics of Operations Research*, 7, 515 - 531, 1982.

[7] D. Fulkerson. Blocking and anti-blocking pairs of polyhedra. *Mathematical Programming,* 1, 168-194, 1971.

[8] M.X. Goemans and D.P. Williamson. A general approximation technique for constrained forest problems. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms,* page 307-316, 1992.

[9] M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos and D. Williamson. Improved approximation algorithms for network design problems. *5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1993.

[10] M. Grötschel, L. Lovász, A. Schrijver. *Geometric algorithm and combinatorial optimization.* Springer-Verlag, Berlin, 1988.

[11] N. Hall and D. Hochbaum. A fast approximation algorithm for the multicovering problem. *Discrete Applied Mathematics,* 15, 35-40, 1989.

[12] D.S. Johnson. The NP-completeness column: An on-going guide, *Journal of Algorithms,* 3, page 288-300, 1982.

[13] M. Jünger, G. Reinelt, G. Rinaldi. The travelling Salesman Problem. To appear.

[14] R. Karp and H. Papadimitriou. On linear characterization of combinatorial optimization problems. *SIAM J. of Computing* 11,4,page 620-632,1982.

[15] P. Klein and R. Ravi. When cycles collapse: A general approximation technique for constrained two-connectivity problems. In *proceedings of the 3rd MPS Conference on Int. Prog. and Comb. Opt.,* page 33-55, 1993.

[16] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing,* page 286-293, 1993.

[17] G. Nemhauser and L. Wolsey. Integer and Combinatorial Optimization. John Wiley and Sons, New York, NY, 1988.

[18] L. Shapley. Cores of convex games. *International J. Game Theory,* 1, 11-26, 1971.

[19] A. Schrijver. Polyhedral proof methods in combinatorial optimization. *Discrete Applied Mathematics*, 14, 111-133, 1986.

[20] D. Williamson, M. Goemans, M. Mihail and V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing,* page 708-717, 1993.

[21] D. Williamson. On the design of approximation algorithms for a class of graph problems, MIT, Ph.D thesis.