FROM VALID INEQUALITIES TO HEURISTICS: A UNIFIED VIEW OF PRIMAL-DUAL APPROXIMATION ALGORITHMS IN COVERING PROBLEMS

DIMITRIS BERTSIMAS
CHUNG-PIAW TEO

Massachusetts Institute of Technology, Cambridge, Massachusetts

(Received May 1994; revisions received July 1995, February 1996; accepted February 1996)

In recent years approximation algorithms based on primal-dual methods have been successfully applied to a broad class of discrete optimization problems. In this paper, we propose a generic primal-dual framework to design and analyze approximation algorithms for integer programming problems of the covering type that uses valid inequalities in its design. The worst-case bound of the proposed algorithm is related to a fundamental relationship (called strength) between the set of valid inequalities and the set of minimal solutions to the covering problems. In this way, we can construct an approximation algorithm simply by constructing the required valid inequalities. We apply the proposed algorithm to several problems, such as covering problems related to totally balanced matrices, cyclic scheduling, vertex cover, general set covering, intersections of polymatroids, and several network design problems attaining (in most cases) the best worst-case bound known in the literature.

In the last 20 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 at particular instances. The research community typically relies on computational experience to evaluate the valid inequalities.

The second approach involves design and analysis of approximation algorithms. The quality of solutions produced is 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, and (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 $\mathcal{NP}$ (see Arora et al. 1992), and for some problems it has produced algorithms that have been successfully used in practice (see, for instance, Williamson 1994).

In recent years, it has been recognized that tight LP relaxations can often be used as a basis for analyzing and deriving performance bounds for heuristics for NP-hard problems. By comparing the heuristic value with the value of the **dual** of the proposed LP relaxations, the **primal-dual** method has been successfully applied to analyze a variety of exact and approximation algorithms. Some of the best known exact algorithms for problems in combinatorial optimization, including the matching, spanning tree, shortest path, and network flow problems, are based on this approach. (See Nemhauser and Wolsey 1988.) The method has also been applied to analyze approximation algorithms for covering problems (see Chvatal 1979, Dobson 1982, Hochbaum 1982, Bar-Yehuda and Even 1981, and Hall and Hochbaum 1989), vertex feedback set problems (Bar-Yehudar et al. 1994), and network survivability problems (Klein and Ravi 1993, Goemans and Williamson 1992, Goemans et al. 1993, Williamson et al. 1993, Williamson 1994). These analyses, however, appear to be problem specific. Moreover, the analyses do not usually offer insight into the **design** of such algorithms.

This paper is motivated by the authors' desire to find an algorithmic technique to design approximation algorithms when better formulations are available. Based on the work of Goemans and Williamson, we present a generic framework to design approximation algorithms for covering-type problems. We adopt the conventional approach of measuring the tightness of a relaxation by the ratio of the optimum fractional solution (denoted Z) and the IP optimum (denoted IZ). In particular, we say an LP relaxation is a λ-approximation if $Z \leqslant IZ \leqslant \lambda Z$. We also say that a proposed heuristic H that produces a solution with value Z_H is a λ-approximation algorithm if $Z_H \leqslant \lambda IZ$. Throughout this paper we consider only minimization problems, and we assume that the objective function is nonnegative. The key property exploited in this paper is the notion of a *minimal* integral solution. A feasible integral solution x is called minimal if $x - e^i$ is not feasible for all i (e^i is the i-th

Subject classifications: Programming: integer, heuristic.
Area of review: OPTIMIZATION.

0030-364X/98/4604-0503 $05.00

unit vector with all coordinates zero, except for a one in the ith coordinate).

For a covering-type problem with a nonnegative objective function, it is easy to see that the set of minimal integral solution contains an optimal solution to the problem.

The contributions of the paper are as follows.

1. We present a generic primal-dual framework to **design** and **analyze** approximation algorithms for integer programming problems of the covering type, utilizing valid inequalities of the problems to design the algorithm. Furthermore, we show that the worst-case approximation bound corresponds to a fundamental relationship (called strength) between the valid inequalities used and the set of minimal integral solutions. In this way we can construct a λ-approximation algorithm simply by constructing the required valid inequalities.
2. We apply the framework to several classical problems, attaining in most cases the best worst-case bound known in the literature. Though the algorithms constructed by this framework are not as efficient as the known ones, the analysis of the algorithms is considerably simpler.

The paper is structured as follows. In Section 1, we describe the generic primal-dual framework for 0–1 covering problems. We also introduce the notion of *strength* for a set of valid inequalities, and we prove that the performance of the primal-dual algorithm is bounded by the strength. Furthermore, utilizing the underlying geometry of the algorithm, we show that the bound is tight. In Section 2, we introduce the notion of reducible formulations and show that a large collection of problems fall into this framework. We further show how to compute the strength of several problems that have reducible formulations. In Section 3, using an idea from Williamson et al., we consider variations of the basic primal-dual algorithm for more general problems.

1. PRIMAL-DUAL FRAMEWORK FOR COVERING PROBLEMS

In this section we propose a framework to design and analyze primal-dual approximation algorithms for problems of the following type:

$$(IP)\ \min \sum_{j=1}^{n} c_j x_j,$$

$$\text{subject to} \sum_{j=1}^{n} a_{ij} x_j \geqslant b_i, \quad \forall\, i \in \{1, \ldots, m\},$$

$$x_j \geqslant 0,$$

$$x_j \in \{0, 1\},$$

where all parameters a_{ij}, c_j are nonnegative. Problems of this type are called 0–1 **covering** problems. We let A denote the constraint matrix (a_{ij}), and b the right-hand-side vector. We also denote by Z(cover) (resp. IZ(cover)) the value of the optimal LP (resp. IP) solution. Z(cover) is obtained from (IP) by removing the constraint that x is a 0–1 vector. We do not include the constraints $x_j \leqslant 1$ explicitly in Z(cover). Note that, in general, we do not need b_i to be nonnegative, since the ith constraint is trivially redundant in this case.

We first describe informally the ideas on which the algorithm is based. The primal LP we consider at first is the set $\mathcal{S}$ of all valid inequalities to IZ(cover). It contains in particular the set of inequalities in Z(cover). At each step of the algorithm a valid inequality is chosen, and this will be used to decide which one of the primal variables will be fixed at 1. We then repeat the procedure on the modified problem instance. Note that any valid inequality for the modified problem instance is automatically in $\mathcal{S}$. After a feasible primal solution is found in this way, we use a reverse deletion step to ensure that the solution obtained is a minimal solution. More formally, the algorithm proceeds as follows.

Primal-dual Algorithm $\mathcal{PD}$

- Input: A, b, c, $(A, c \geqslant 0)$.
- Output: x_H feasible for (IP) if the problem is feasible.

Step 1. **Initialization:** Let $r = 1$; $A^1 = A$, $b^1 = b$, $c^1 = c$; $IS_1 = \{x \in \{0, 1\}^n\colon Ax \geqslant b\}$; $IP_1 = \min\{c^1 x : x \in IS_1\}$, $F_1 = \{1, \ldots, n\}$.

Step 2. **Addition of valid inequalities:** Construct a valid inequality $\sum_{i \in F_r} \alpha_i^r x_i \geqslant \beta^r$ for the convex hull of solutions in IS_r:
Set

$$y_r \leftarrow \min_{i \in F_r} \left\{ \frac{c_i^r}{\alpha_i^r} : \alpha_i^r > 0 \right\},$$

$$k(r) \leftarrow \operatorname*{argmin}_{i \in F_r} \left\{ \frac{c_i^r}{\alpha_i^r} : \alpha_i^r > 0 \right\}.$$

Step 3. **Problem modification:**
Set $\bar{x}_{k(r)} = 1$; $F_{r+1} = F_r \backslash \{k(r)\}$.
Set $A^{r+1} = A^r \backslash A^r_{k(r)}$, $b^{r+1} = b^r - A^r_{k(r)}$, $c^{r+1} = c^r - y_r \alpha^r$.
Set $IS_{r+1} = \{x \in \{0, 1\}^{n-r}\colon A^{r+1} x \geqslant b^{r+1}\}$.
Let (IP_{r+1}) denote the instance $\min\{c^{r+1} x\colon x \in IS_{r+1}\}$;
$r \leftarrow r + 1$. Repeat Step 2 until the solution obtained is feasible to the original problem, else conclude that the problem is infeasible.

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

- Set $C_r \leftarrow \{x_{k(r)}\} \cup C_{r+1}$.
- Set $\bar{x}_{k(r)} = 0$ if $C_r \backslash \{x_{k(r)}\}$ corresponds to a minimal feasible solution to problem instance IP_r.

Step 5. Set $x_i^H = 1$ if $x_i \in C_1$. Return x^H. Let $Z_H = cx^H$.

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

In order to analyze the performance of the algorithm we introduce the following definition.

Definition 1. The strength $\lambda_r = s(\alpha^r, \beta^r)$ of the inequality $\sum_i \alpha_i^r x_i > \beta^r$ with respect to instance IP_r is defined to be:

$$\lambda_r = \max\left\{\frac{\sum_i \alpha_i^r w_i}{\beta^r} : w \text{ minimal integral solution for } IP_r\right\}.$$

A different notion of strength, has been used earlier by Goemans (1995) to compare several valid inequalities for the Traveling Salesman Problem. The reverse deletion step that ensures the minimality of the solution was first used on a class of network design problems in Goemans and Williamson, Williamson et al., and Williamson. Algorithm $\mathcal{PD}$ can be seen as a generalization of these algorithms to general covering problems.

Let t be the number of valid inequalities added by the primal-dual algorithm. In order to bound the performance of the primal-dual algorithm let:

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

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

Note that by our construction all constraints in (LP_D) are feasible for the original covering problem IP_1. Thus, $Z_D \leqslant IZ(\text{cover})$.

Theorem 1. *Let x^H, y denote the primal and dual solution returned by the algorithm $\mathcal{PD}$. Let λ denote the maximum strength of all the inequalities used. Then*

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

In particular:

(*a*) $Z_H \leqslant \lambda IZ(cover)$.

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

Proof. By construction, x^H and y are primal and dual feasible, respectively. Let $x_{k(r)}$ be the variable selected in the rth stage of the algorithm. Let x^r be obtained from x^H by setting $x^H_{k(1)}, \ldots, x^H_{k(r-1)}$ to 0. By construction, x^r is a minimal solution to (IP_r). We first prove by induction that for every $r = t$ to 1:

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

For $r = t$, since x^t is a minimal solution to IP_t, and by the definition of strength

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

which implies that

$$c^t x^t = c_{k(t)} = y_t \alpha^t_{k(t)} = y_t \alpha^t x^t \leqslant \lambda y_t \beta^t.$$

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

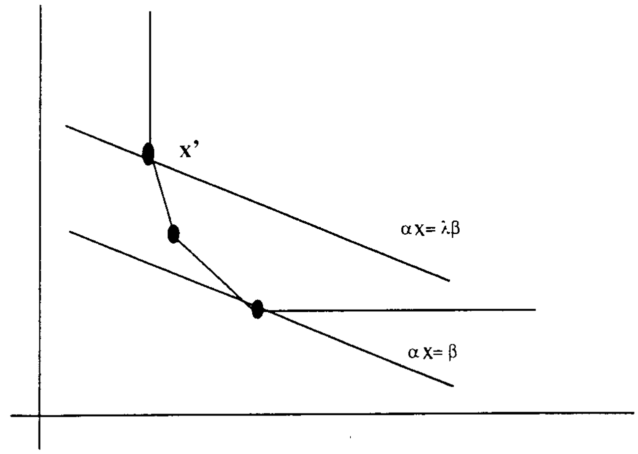

Figure 1. Geometry of the strength of an inequality.

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

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 $\alpha^r x^r \leqslant \lambda \beta^r$ (by the definition of strength and the minimality of x^r), we obtain (1), since

$$c^r x^r \leqslant \lambda \sum_{i \leqslant r+1} y_i \beta^i + \lambda y_r \beta^r.$$

Since $\{y_j\}_{j \geqslant 1}$ forms a feasible solution to the dual of the relaxation (LP_D),

$$Z_H = cx^H \leqslant \lambda \sum_r y_r \beta^r \leqslant \lambda Z_D \leqslant \lambda IZ(\text{cover}).$$

If in addition, all the inequalities $\alpha^r x \geqslant \beta^r$ are redundant to $Ax \geqslant b$, $x \geqslant 0$, then $Z_H \leqslant \lambda Z(\text{cover})$. □

By this theorem, we can construct a λ-approximation algorithm if we can find valid inequalities with strength bounded by λ at each stage of the algorithm. For cases where several of these inequalities exist, each inequality gives rise to a different primal-dual approximation algorithm. The running time of the algorithm is determined by the time to construct a valid inequality, update the objective function and the right-hand side and delete redundant rows and construct a minimal solution at each stage of the algorithm. Note that the LP formulation need not be explicitly given. Since the number of stages is clearly upper bounded by the number of variables, the algorithm runs in polynomial time if the previous three steps can be implemented in polynomial time.

1.1. A Geometric View of the Primal-Dual Algorithm

Let us first develop some geometric insight on the strength of an inequality. Let $conv(IP)$ denote the convex hull of all minimal integral solutions to problem (IP). Let $\alpha x \geqslant \beta$ denote a valid inequality for $conv(IP)$, touching $conv(IP)$ at a vertex x^1. (See Figure 1.) It corresponds to a hyperplane with all the vertices of $conv(IP)$ on one side. Let λ denote the strength of this inequality with respect to (IP).

By the definition of λ, the vertices of *conv*(*IP*) are "sandwiched" between the hyperplane $\alpha x = \beta$ and $\alpha x = \lambda\beta$. A valid inequality that gives the "thinnest" slab sandwiching the vertices of *conv*(*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 1 is essentially tight.

Theorem 2. *Assume that the first valid inequality $\alpha x \geqslant \beta$ 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*

$$Z_H \geqslant \lambda(1 - \epsilon)IZ(cover).$$

Proof. Let x' be a minimal solution with $\alpha x' = \max\{\alpha x : x$ 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, after updating the cost function with the valid inequality $\alpha x \geqslant \beta$ in the algorithm, the new costs for all $i \in C$, are 0. Thus, the algorithm always returns the solution x', with objective value

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

Moreover, $IZ(\text{cover}) \leqslant cx^1$, where x^1 is a vertex in conv(IP) with $\alpha x^1 = \beta$. Therefore,

$$IZ(\text{cover}) \leqslant \alpha x^1 + \gamma \sum_{i \notin C} x^1_i \leqslant \beta + \gamma n.$$

By choosing $\gamma = \epsilon\beta/n$, we can ensure that under the cost c

$$\frac{Z_H}{IZ(\text{cover})} \geqslant \frac{\lambda\beta}{\beta + \gamma n} = \frac{\lambda}{1 + \epsilon} \geqslant \lambda(1 - \epsilon). \quad \square$$

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

Remark 2. For the network design problem, it was also clear from the work of Goemans and Williamson and from Williamson et al. that a valid inequality with improved strength would improve the approximation bound of their algorithms.

Remark 3. In the next section we apply Algorithm $\mathcal{PD}$ to several 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.

2. REDUCIBLE FORMULATIONS AND APPROXIMABILITY

In this section we illustrate the power of Theorem 1 by showing that some of the best known results in approximation algorithms for covering problems are special cases of Theorem 1. Theorem 1 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 at each stage we need to bound the strength of the inequality we add with respect to a new problem instance. We next illustrate that for a rather rich class of formulation, bounding the strength can be greatly simplified.

2.1. Reducible Formulations

We consider covering problems of the form:

$$\begin{aligned} (IP_n)\ \min\ & cx \\ \text{subject to}\ & Ax \geqslant b, \\ & x \in \{0, 1\}^n, \end{aligned}$$

where A is an $m \times n$ matrix, and c is an n-vector with nonnegative integer entries. Entries in b are integral, but are 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 $\mathscr{C}$. By fixing variable x_j to 1, we create the following problem:

$$\begin{aligned} (IP^j_{n-1})\ \min\ & \bar{c}\bar{x}, \\ \text{subject to}\ & \bar{A}\bar{x} \geqslant b - A_j \\ & \bar{x} \in \{0, 1\}^{n-1}, \end{aligned}$$

where $\bar{c}$, $\bar{x}$ are obtained from c, x by deleting the coordinate corresponding to variable x_j, and $\bar{A}$ is an $m \times (n - 1)$ matrix obtained from A by deleting the column corresponding to the variable x_j.

Definition 2. Formulation (IP_n) is reducible with respect to problem class $\mathscr{C}$ if for all j formulation (IP^j_{n-1}) belongs to problem class $\mathscr{C}$.

In other words, reducible formulations of a problem with respect to a problem class $\mathscr{C}$ have the property that the new smaller instance that results by fixing a variable still belongs in problem class $\mathscr{C}$. 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.

2.2. The Minimum Spanning Tree Problem

Let G denote an undirected graph on the vertex set V and edge set E. Let $\delta(S)$ denote the set of edges in $[S, \bar{S}]$. More generally, if $(S_1, \ldots, S_k)$ is a partition of the node set V, then $\delta(S_1, \ldots, S_k)$ denote the set of edges in $[S_i, S_j]$ for all $i \neq j$. 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:

$$(CUT)\ IZ_{CUT} = \min cx,$$

$$\text{subject to} \quad \sum_{e\in\delta(S)} x_e \geqslant 1, \ \forall S \subset V,$$

$$x_e \in \{0, 1\}.$$

By fixing some x_e to be 1, we obtain a new cut formulation for the MST on $|V| - 1$ nodes. The new problem instance is obtained from G 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 (1971), we arrive at the multicut formulation:

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

$$\text{subject to} \quad \sum_{e\in\delta(S_1,\ldots,S_k)} x_e \geqslant k-1, \ \forall (S_1, \ldots, S_k)$$

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. By applying Theorem 1 we provide a simple proof of the integrality of the multicut polyhedron, as well as the known tight bound on the duality gap of IZ_{CUT} and Z_{CUT}.

Theorem 3. *The inequality $\sum_{e\in E} x_e \geqslant n - 1$ is valid for the multicut polyhedron and has strength* 1, *i.e.*,

$$IZ_{MCUT} = Z_{MCUT}. \tag{2}$$

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

$$\frac{Z_H}{Z_{CUT}} \leqslant 2\left(1 - \frac{1}{n}\right). \tag{3}$$

Proof. We first consider the multicut formulation $(MCUT)$. Since $\sum_e x_e \geqslant n - 1$ 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 (2).

We next consider the cut set formulation (CUT). Since $\sum_e x_e \geqslant n/2$ is a valid inequality (add all the cut inequalities for singletons), the strength is $2(1 - 1/n)$, thus showing (3). □

Remark 4. Using inequalities $\sum_e x_e \geqslant n-1$ in Algorithm $\mathcal{PD}$, we obtain the classical Kruskal Algorithm.

2.3. 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 modeled as an edge-covering formulation

$$(SP)\ IZ_{SP} = \min cx,$$

$$\text{subject to} \quad \sum_{e\in\delta(S)} x_e \geqslant 1, \ \forall S:$$

$$s \in S,\ t \notin S \text{ or } t \in S,\ s \notin S,$$

$$x_e \in \{0, 1\}.$$

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

Theorem 4. *Inequalities*:

1. $x(\delta(s)) \geqslant 1$;
2. $x(\delta(t)) \geqslant 1$; *and*
3. $\alpha x(\delta(s)) + \beta x(\delta(t)) \geqslant \alpha + \beta$,

have strength 1, *i.e.*,

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

Using any of these inequalities in each stage of our primal-dual framework, 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.

2.4. Special Cases of the General Covering Problem

Consider the problem:

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

$$\text{subject to} \quad Ax \geqslant b,$$

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

where a_{ij} and c_j are nonnegative integers. Fixing some variable x_j to 1, we have 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 proposed a dual heuristic for the case when a_{ij} are 0 or 1, with $Z_H(GC) \leqslant fZ_{GC}$, $f = \max_i \sum_{j=1}^{n} a_{ij}$. We refer to this bound as the row-sum bound. Bertsimas and Vohra (1998) 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 5. *The strength of the inequalities $a_i x \geqslant b_i$, $i = 1, \ldots, m$ is at most f, i.e., Algorithm $\mathcal{PD}$ applied to these inequalities produces a solution such that*

$$\frac{Z_H}{Z_{GC}} \leqslant f. \tag{4}$$

Proof. Consider an inequality $a_i x \geqslant b_i$. Let x' be a minimal solution to (GC). Clearly, $a_i x' \leqslant f$; therefore, $\lambda_i \leqslant f/b_i \leqslant f$. Since the row sum reduces after each step of the algorithm, the strength of all inequalities is bounded above by f. Therefore from Theorem 1, (4) follows. □

2.5. Vertex Feedback Set

Let G be a graph with weights defined on its nodes. The Vertex Feedback Set (VFS) problem asks for a minimum weighted set of nodes that intersects all cycles of the graph G. A natural set covering formulation for this problem has the form

$$\sum_{v \in C} x(v) \geqslant 1 \quad \forall \text{ cycle } C \text{ in } G.$$

Recently, Bar-Yehuda et al. constructed a 4 log($|V|$) approximation algorithm for VFS, using the following results of Erdös and Pósa 1962.

Theorem 6. *If G has minimum degree at least* 3, *then G contains a cycle of length not more than* 2 *log*($|V|$).

We next show that this theorem leads immediately to the proposed bound for the (VFS). The analysis uses several ideas from Bar-Yehuda et al. In general, all degree 1 vertices in the graph G can be deleted, since they will not be contained in any minimal VFS. Moreover, by repeatedly contracting the degree 2 vertices with their neighbor, we obtained a new graph G' with minimum degree at least 3. By the above theorem, there is a cycle C in G' with at most 2 log($|V|$) nodes. We can then construct a cycle C' in G such that C' consists of at most 2 log($|V|$) nodes of degree at least 3 and at most 2 log($|V|$) disjoint chains consisting of degree 2 nodes. Since all minimal VFS intersect these chains at most once, we have Theorem 7.

Theorem 7. *The inequality $\sum_{v \in C'} x(v) \geqslant 1$ has strength at most* 4 *log*($|V|$).

By deleting all cycles that contain a particular node, say v, we reduced the problem to another VFS on $G - v$. Hence at each stage of the algorithm, there exists an inequality of strength at most 4 log($|V|$). The algorithm $\mathcal{PD}$ in this case corresponds to the 4 log($|V|$) approximation algorithm proposed in Bar-Yehuda et al.

2.6. Proper and Uncrossable Functions

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

$$(UC)\ IZ_{UC} = \min \sum_{e} c_e x_e,$$
$$\text{subject to} \sum_{e \in \delta(S)} x_e \geqslant f(S),\ S \subset V,$$
$$x_e \in \{0, 1\},$$

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 $f(S \cap T) = f(S \cup T) = 1$.
f is called *uncrossable* if the above property holds. A two-approximation algorithm for this class of problem was first proposed by Williamson et al. It generalized an earlier algorithm in Goemans and Williamson designed for a more restrictive 0–1 function f, such that

$$f(S \cup T) \leqslant \max\{f(S), f(T)\}, \tag{5}$$

for all disjoint S and T, and f symmetric. Symmetric functions f satisfying (5) are called *proper* functions. Note that the conditions for properness imply uncrossability. We refer the readers to Goemans and Williamson for a long list of problems that can be modeled 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 Williamson et al., we offer a new self-contained proof.

We say S is a *minimal* set if $f(S) = 1$ and $f(T) = 0$ for all $T \subset S$.

Theorem 8. *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*

$$\sum_{j=1}^{l} \sum_{e \in \delta(S_j)} x_e \geqslant l, \tag{6}$$

is $2(1 - 1/l)$, *i.e.*,

$$\frac{Z_H}{Z_{UC}} \leqslant 2\left(1 - \frac{1}{l}\right).$$

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

$$\sum_{j=1}^{l} x_F(\delta(S_j)) \leqslant 2(l - 1). \tag{7}$$

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. We use also the following fact:
Contract $G[F]$ at the sets $S_1, \ldots, S_l$. *The contracted graph* $G'[F]$ is again a forest.

Let $U = V \backslash \{S_1, \ldots, S_j\}$. 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_js and T_ks 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 (7). 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 (7) follows immediately from the forest structure of F'. For each i, delete T_i and its incident edges. Connect all neighbors of T_i by a path. This transformation does not increase the count for (7). In this way, we obtain a forest defined only on the nodes of $S_1, \ldots, S_j$. Since the forest has at most $l - 1$ edges,

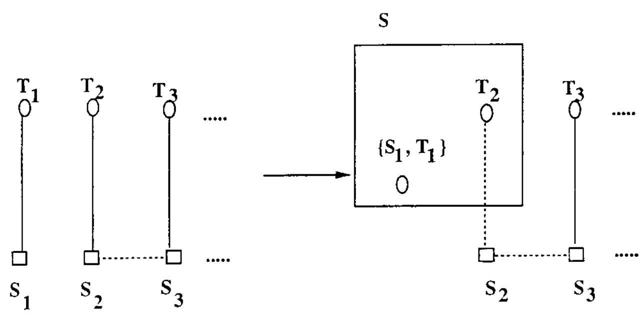

Figure 2. Case 1 with deg $S_1 = 1$.

we have the result. So we may assume that $\deg(T_1) = 1$, and the edge e connects T_1 to the vertex S_1.

We will use induction on the number of nodes in G' to compute (7). To do so, we will contract a suitable subgraph of G' with 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, with $f'(S) = 1$ that contains the component $\{S_1, T_1\}$ and does not contain S_i for any $i \geq 2$, 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 (7) 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, $i \geq 2$. (See Figure 2.)

This edge is counted twice in this contracted instance under the induction hypothesis, whereas its contribution to (7) 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 $\delta(W) = \{e\}$, where $e = (S_1, T_1)$, $f'(W) = 1$, and $S_1 \subset W$. By symmetry, $f'(\bar{W}) = 1$. Thus, $|W| \geq 2$, $|\bar{W}| \geq 2$. Let G_W, $G_{\bar{W}}$ denote respectively the graph obtained from G' by contracting $\bar{W}$ and W into a single node. (See Figure 3.)

These are minimal solutions with respect to f' restricted to the vertex sets of G_W and $G_{\bar{W}}$. Let l_W, $l_{\bar{W}}$ denote the number of S_i's contained in W and $\bar{W}$, respectively. By our modification, the number of disjoint minimal sets in G_W and $G_{\bar{W}}$ are $l_W + 1$ and $l_{\bar{W}} + 1$, respectively. Using induction on the number of nodes, the contribution of edges in G_W and $G_{\bar{W}}$ to (7) are, at most, $2l_W$ and $2l_{\bar{W}}$, respectively. Note that the edge $e = (S_1, T_1)$ has been counted three times, once in $G_{\bar{W}}$ and twice in G_W, whereas its contribution to (7) is 1. Therefore,

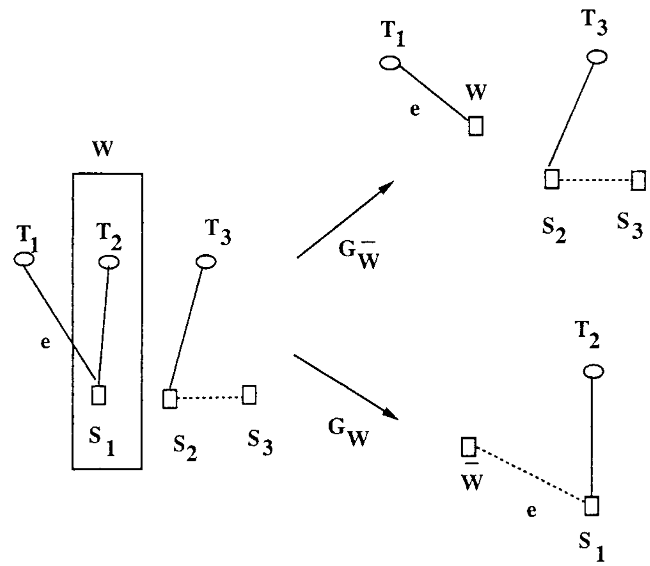

Figure 3. Case 2 with deg (S_1) greater than 1.

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

Hence 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 $2 - 1/(l-1)$ 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 $f(v) = 1$; thus, we could implement the primal-dual algorithm in polynomial time. For the case of uncrossable functions, the question is open.

2.7. Constrained Contra-polymatroids

Consider the problem

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

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

$$f(S) + f(T) \leq f(S \cap T) + f(S \cup T) \text{ (supermodular);} \quad (8)$$
$$f(S) \leq f(T),\ \forall S \subset T \text{ (nondecreasing).} \quad (9)$$

The function f is called a contra-polymatroid function (see Shapley 1971). 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 $N \backslash \{i\}$, with $f'(S) = \max(f(S), f(S \cup i) - 1)$ for all S in $N \backslash \{i\}$. Clearly, $f'(S) \leqslant f'(T)$ if $S \subset T \subset N \backslash \{i\}$. To show supermodularity, suppose $f'(S) = f(S)$, $f'(T) = f(T \cup i) - 1$. Then

$$\begin{aligned} f'(S) + f'(T) &\leqslant f(S \cap T) + f(S \cup \{T + i\}) - 1 \\ &\leqslant f'(S \cap T) + f'(S \cup T). \end{aligned}$$

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

Theorem 9. *The inequality*

$$\sum_{i \in N} x_i \geqslant f(N),$$

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 $x'_1 = 1$. Hence

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

Hence $S_i \cup S_j$ is again tight. By repeating this procedure, we obtain $x'(T) = f(T)$ for some T with $x'_i = 0$ for $i \notin T$. Since $f(T) \leqslant f(N)$, $x'(N) = x'(T) \geqslant f(N)$, 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, provided we use the valid inequality $\sum_{i \in S} x_i \geqslant f(S)$ at each stage of the algorithm, where S is chosen to be the set that maximizes $f(\cdot)$ and has the maximum cardinality among such sets.

A direct generalization of this argument to the intersection of k constrained contra-polymatroids leads to the following theorem:

Theorem 10. *The strength of the inequality* $\sum_i x_i \geqslant f_1(N) + \cdots + f_k(N)/k$ *for the intersection of* k *contra-polymatroids is* k, *i.e., Algorithm* $\mathcal{PD}$ *has a worst-case bound of* k.

Remark 5. Although for $k = 2$ there exists a polynomial algorithm (see Nemhauser and Wolsey 1978), Algorithm $\mathcal{PD}$ has a faster running time. Note that Fisher et al. studied a related problem on packing of k matroids, attaining also the same bound k.

2.8. Set Covering Problems

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

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

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

1. A is called a row-inclusion matrix if A does not contain the submatrix $\binom{11}{01}$. This class of matrices plays an important role in the study of totally balanced matrices. (See Nemhauser and Wolsey.) 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.

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

Proof. Consider a minimal solution x'. We show that $a_{11}x'_1 + \cdots + a_{1n}x'_n \leqslant 1$. 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 exists 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 $a_{11}x_1 + \cdots + a_{1n}x_n \geqslant 1$ has strength 1, proving that Algorithm $\mathcal{PD}$ finds an optimal solution in this case. □

2. Suppose A has consecutive ones in all its columns. This class of matrices belongs to the class of totally unimodular matrices, and therefore the underlying polyhedra are integral. There exists an optimal algorithm that first transforms the problem to a shortest path problem. (See, for instance, Ahuja et al. 1993.) We show that Algorithm $\mathcal{PD}$ is a direct optimal algorithm for the problem.

Theorem 12. *The strength of the first inequality* $a_{11}x_1 + \cdots + a_{1n}x_n \geqslant 1$ *is* 1.

Proof. Consider a minimal solution x'. We will show that $a_{11}x'_1 + \cdots + a_{1n}x'_n \leqslant 1$. 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 $a_{11}x_1 + \cdots + a_{1n}x_n \geqslant 1$ has strength 1, proving that Algorithm $\mathcal{PD}$ finds an optimal solution in this case. □

3. For arbitrary 0–1 matrix A, a direct generalization of the previous argument yields Theorem 13.

Theorem 13. *The first inequality* $a_{11}x_1 + \cdots + a_{1n}x_n \geqslant 1$ *has strength*

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

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

Obviously, the previous bound can be optimized by considering permutations of the rows to attain the bound

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

4. Suppose A has consecutive ones in each row. We may assume without loss of generality that there is no redundant inequality in the constraints.

Theorem 14. *Inequality* $x_1 + x_2 + \cdots + x_L \geqslant 1$ *has strength* 1, *where for some* L, $a_{ij} = 1$ *for* $j \leqslant L$, *and* $a_{ij} = 0$ *for* $j > L$.

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

5. Matrices A with circular ones in rows. (See Bartholdi et al. 1980.) Again we may assume that there is no redundant inequality in the constraints. By similar reasoning we can easily show Theorem 15.

Theorem 15. *Every constraint in* $Ax \geqslant 1$ *has strength at most* 2.

3. MULTIPHASE EXTENSION OF THE PRIMAL-DUAL ALGORITHM

In this section we propose an extension of Algorithm $\mathcal{PD}$ to 0–1 covering problems, in which the right-hand-side vector b need not be all 1, unlike most of those problems considered in the previous section. In this case, the strength of the valid inequalities tends to be weak. To achieve a better approximation bound, we use an idea from Williamson et al. to apply Algorithm $\mathcal{PD}$ in phases.

Let $b_{\max} = \max_i b_i$. Note that we will delete columns from the constraints as the algorithm progresses, but by slight abuse of notation, we will maintain the dimension of the x-vector. The coordinates of x corresponding to deleted columns should be treated as zero.

Multiphase primal-dual Algorithm $\mathcal{MPD}$

Step 1. **Initialization:** $k = 1$, $A^1 = A$, $b^1 = b$, $c^1 = c$.

Step 2. Phase k: Let $h_i = 1$ if $b_i = b_{\max}$ and $h_i = 0$ otherwise.

Apply Algorithm $\mathcal{PD}$ to the problem:

$$IZ_k = \min c^k x,$$
$$\text{subject to} \quad A^k x \geqslant h^k$$
$$x \in \{0, 1\}^n,$$

yielding a solution x^k with cost $Z_H^k = c^k x^k$.

Let $J_K = \{j : x_j^k = 1\}$.
$A^{k+1} = A^k \backslash \{A_j^k\}_{j \in J_k}$ (delete the columns of A^k corresponding to the indices in set J_k).
$b^{k+1} = b^k - A^k x^k$, $c^{k+1} = c^k \backslash \{c_j\}_{j \in J_k}$, and $k = k + 1$.

Step 3. Repeat Step 2 until the solution x with $x_j = 1$ for all $j \in \cup_k J_k$ is feasible. If after $\min(b_{\max}, n)$ phases a feasible solution is not found, conclude that the problem is infeasible.

Lemma 1. *Any valid inequality for phase* $k + 1$ *is valid for phase* k.

Proof. Suppose $\alpha x \geqslant \beta$ is valid for $\{A^{k+1}x \geqslant b^{k+1}, x \in \{0, 1\}^n\}$. Note that $\alpha_j = 0$ for j in J_k, since the corresponding columns have been deleted before phase $k + 1$. We will show that all x^* such that $A^k x^* \geqslant b^k$, $x^* \in \{0, 1\}^n$ satisfy $\alpha x^* \geqslant \beta$.

Let $x^* = (x^*_{J_k}, x^*_{\bar{J}_k})$. Then

$$A^k x^* = A^{k+1} x^*_{\bar{J}_k} + A^k x^*_{J_k} \geqslant b^k,$$

hence

$$A^{k+1} x^*_{\bar{J}_k} \geqslant b^k - A^k x^*_{J_k} \geqslant b^{k+1}.$$

The last inequality follows from the fact that $b^{k+1} = b^k - \sum_{j \in J_k} A_j^k$ and x^* is 0–1. Hence, $\alpha x^* = \alpha x^*_{\bar{J}_k} \geqslant \beta$. □

Let x^H be the solution obtained by Algorithm $\mathcal{MPD}$ and $Z_H = cx^H$. In the next theorem we bound the performance of the algorithm.

Theorem 16. *There are the following cases.*

1. *If at each phase k the worst bound for the strength of inequalities used in Algorithm* $\mathcal{PD}$ *is* λ, *and all inequalities used by Algorithm* $\mathcal{PD}$ *are redundant for* $\{A^k x \geqslant h^k, x \geqslant 0\}$ *for each phase k, then*

$$\frac{Z_H}{Z} \leqslant \lambda \mathcal{H}(b_{max}), \tag{10}$$

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

2. *If at each phase k the worst bound for the strength of inequalities used in Algorithm* $\mathcal{PD}$ *is* λ, *but some inequalities are not redundant to* $\{A^k x \geqslant h^k, x \geqslant 0\}$ *for each phase k, then*

$$\frac{Z_H}{IZ} \leqslant \lambda b_{max}. \tag{11}$$

Proof. We prove the theorem by induction on $b_{\max}$. For $b_{\max} = 1$, Algorithm $\mathcal{MPD}$ reduces to Algorithm $\mathcal{PD}$ and (10) follows from Theorem 1. Assuming (10) is true for $b_{\max} - 1$, we prove it for $b_{\max}$. For ease of exposition we introduce the notation:

$$P(b, c)\colon Z_{b,c} = \min cx,$$
$$\text{subject to} \quad Ax \geqslant b,$$
$$x \in [0, 1]^n.$$

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 $\mathcal{MPD}$, the solution x_1 produced has cost

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

because the solution $x^*(b, c)/b_{\max}$ 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. Let b' denote the right-hand-side parameters for the next stage. Clearly,

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

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

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

The superposition of the solutions x_1 and x'_H with support $J_1 \cup J'_H$ is the solution produced by Algorithm $\mathcal{MPD}$ on the original input. It has cost $Z_H = \sum_{j \in J_1 \cup J'_H} c_j \leqslant \lambda(\mathcal{H}(b_{\max} - 1) + 1/b_{\max}) Z_{b,c} = \lambda \mathcal{H}(b_{\max}) Z_{b,c}$, proving (10).

When the value of the heuristic is within λ from the optimal integer solution, the proof is identical, except that we can only guarantee $\sum_{j \in J_1} c_j \leqslant \lambda IZ_{h,c} \leqslant \lambda IZ_{b,c}$. The induction on $b_{\max}$ proceeds along the same lines, except that $Z_H = \sum_{j \in J_1 \cup J'_H} c_j \leqslant \lambda(b_{\max} - 1) IZ_{b,c} + \lambda IZ_{b,c} = \lambda b_{\max} IZ_{b,c}$. □

3.1. Applications

In this section we outline a number of applications of Theorem 16. 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 $\mathcal{MPD}$, columns from matrix A and redundant constraints are deleted, and the new problem instance reduced to $A'x \geqslant 1$, where A' again has the consecutive ones property. Hence the formulation is reducible; therefore, at each phase of $\mathcal{MPD}$ we can use Theorem 12 (respectively, 14) with $\lambda = 1$. Applying Theorem 16, Algorithm $\mathcal{MPD}$ leads to a solution x_H with

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

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. Cut covering problems with weakly supermodular functions. Goemans et al. generalized the notion of uncrossable function considered in Section 1 to arbitrary nonnegative function f. A function f is called weakly supermodular function, if it is symmetric and

$$f(S) + f(T) \leqslant \max\{f(S - T) + f(T - S), f(S \cup T) + f(S \cap T)\}. \tag{12}$$

If f satisfies the stronger property (5), 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. Note that, however, the formulation is not reducible with respect to arbitrary proper functions (although it is reducible 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 16 immediately applies to derive an approximation algorithm for cut-covering problems with weakly supermodular function f, first obtained in Goemans et al. using considerably more complicated proof methods.

Theorem 17. *Algorithm $\mathcal{MPD}$ 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. Define $h(S) = 1$ if $f(S) = f_{max}$, $h(S) = 0$ otherwise. It can be easily verified that $h(S)$ is a symmetric uncrossable function whenever $f(S)$ 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; therefore, Theorem 16 applies with $\lambda = 2$ (for uncrossable functions, Theorem 1) leading to

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

Remark 6. When f is proper, there is a polynomial time procedure to construct the minimal sets used in the construction of the valid inequalities. (See Williamson et al.). The case for weakly supermodular function is open.

3. More general cut-covering problems.

We consider next an extension of the edge-covering problem (also considered in Goemans et al., 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:

$$(MU) \ \min \sum_e c_e x_e,$$
$$\text{subject to} \sum_{e \in \delta(S)} a_e x_e \geqslant f(S), \quad S \subset V,$$
$$x_e \in \{0, 1\},$$

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 that returns a solution with cost not more than twice the cost of the optimal **integral** solution. The reason that the result does not hold for the optimal LP solution is that the valid inequalities used are not redundant. Given that the formulation is still reducible, we use (11) and obtain a bound of

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

which is also the bound obtained in Goemans et al.

4. CONCLUDING REMARKS AND OPEN PROBLEMS

In this paper we have used the strength of valid inequalities to design approximation algorithms for a broad class of covering problems. An advantage of this approach is that we have decomposed the design of the algorithm into stages, and in each stage we can use the strength of inequalities to guide us to the next stage. This framework allows us to look at many classical and recent primal-dual algorithms in a new way, leading, we believe, to unified and simpler proofs of performance bounds. This framework can also be used to design primal-dual algorithms for large fractional covering problems that arise in several stochastic scheduling problems. We refer the readers to Teo (1996) for details.

The approach also raises several issues in the design of such algorithms:

1. Given a reducible covering problem, can one construct a valid inequality of the least strength? This will be an intrinsic bound for the primal-dual approach. Furthermore, for general set covering problems, is there a routine to generate such valid inequalities?

2. The construction of the worst-case example shows that one should use the information on the objective function to guide the choice of valid inequalities used at each stage of the algorithm. More generally, the strength as a bound arises because there is no coordinated effort across stages during the construction of the algorithm: Can one impose a dynamic programming-like structure into the choice of valid inequality at each stage?

3. The notion of strength depends also on problem instances. By imposing additional structure on the problem, like restricting the graphs to be planar or of bounded-tree-width in the network design problem, will the strength improve?

ACKNOWLEDGMENT

Research partially supported by a Presidential Young Investigator Award DDM-9158118 with matching funds from Draper Laboratory. Research supported by a graduate scholarship from the National University of Singapore.

REFERENCES

AHUJA, R., T. MAGNANTI, AND J. ORLIN. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New York.

ARORA, S., C. LUND, R. MOTWANI, M. SUDAN, AND M. SZEGEDY. 1992. Proof Verification and Hardness of Approximation Problems. In *Proceedings of the 33rd Annual Symposium on Foundation of Computer Science*. 14–23.

BARTHOLDI, J., J. ORLIN, AND H. RATLIFF. 1980. Cyclic Scheduling via Integer Programs with Circular Ones. *Opns. Res.* **28,** 1074–1085.

BAR-YEHUDA, R., AND S. EVEN. 1981. A Linear Time Approximation Algorithm for the Weighted Vertex Cover Problem. *J. Algorithms* **2,** 198–203.

BAR-YEHUDA, R., D. GEIGER, J. NAOR, AND R. ROTH. 1994. Approximation Algorithms for the Vertex Feedback Set Problem with Applications to Constraint Satisfaction and Bayesian Inference. *Proceedings of the 5th Annual ACM-SIAM Symposium of Discrete Algorithms*.

BERTSIMAS, D. AND R. VOHRA. 1998. Rounding Algorithms for Covering Problems. *Math. Programming* **80,** 63–89.

CHVATAL, V. 1979. A Greedy Heuristic for the Set-Covering Problem. *Math. O. R.* **4,** 233–235.

DOBSON, G. 1982. Worst-Case Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data. *Math. O.R.* **7,** 515–531.

ERDÖS, P. AND L. PÓSA. 1962. On the Maximal Number of Disjoint Circuits of a Graph. *Publ. Math. Debrecen* **9,** 3–12.

FISHER, M., G. NEMHAUSER, AND L. WOLSEY. 1978. An Analysis of Approximations for Maximizing Submodular Set Functions-II. *Math. Prog. Study* **8,** 73–87.

FULKERSON, D. 1971. Blocking and Anti-Blocking Pairs of Polyhedra. *Math. Prog.* **1,** 168–194.

GOEMANS, M. X. 1995. Worst-Case Comparison of Valid Inequalities for the TSP. *Math. Prog.*, **69,** 335–349.

GOEMANS, M., A. GOLDBERG, S. PLOTKIN, D. SHMOYS, E. TARDOS, AND D. WILLIAMSON. 1993. Improved Approximation Algorithms for Network Design Problems. *5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 223–232.

GOEMANS, M. X., AND D. P. WILLIAMSON. 1992 A General Approximation Technique for Constrained Forest Problems. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, 307–316.

HALL, N. AND D. HOCHBAUM. 1989. A Fast Approximation Algorithm for the Multicovering Problem. *Discrete Appl. Math.* **15,** 35–40.

HOCHBAUM, D. 1982. Approximation Algorithms for Set Covering and Vertex Cover Problems. *SIAM J. Comput.* **11,** 555–556.

KLEIN, P. AND R. RAVI. 1993. 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.* 33–55.

NEMHAUSER, G. AND L. WOLSEY. 1988. Integer and Combinatorial Optimization. John Wiley and Sons, New York.

SHAPLEY, L. 1971. Cores of Convex Games. *Int. J. Game Theory,* **1,** 11–26.

TEO, C. P. 1971. Cores of Convex Games. *Internat. J. Game Theory,* **1,** 11–26.

TEO, C. P. 1996. Constructing Approximation Algorithms Via Linear Programming Relaxations. Ph.D. Dissertation. Operations Research Center, MIT.

WILLIAMSON, D. 1994. On the Design of Approximation Algorithms for a Class of Graph Problems. Ph.D. Thesis.

WILLIAMSON, D., M. GOEMANS, M. MIHAIL, AND V. VAZIRANI. 1993. A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, 708–717.