JOURNAL OF COMPUTER AND SYSTEM SCIENCES 43, 441–466 (1991)

Expressing Combinatorial Optimization Problems by Linear Programs

MIHALIS YANNAKAKIS

AT&T Bell Laboratories,
Murray Hill, New Jersey 07974

Received December 28, 1988; revised April 1, 1990

Many combinatorial optimization problems call for the optimization of a linear function over a certain polytope. Typically, these polytopes have an exponential number of facets. We explore the problem of finding small linear programming formulations when one may use any new variables and constraints. We show that expressing the matching and the Traveling Salesman Problem by a symmetric linear program requires exponential size. We relate the minimum size needed by a LP to express a polytope to a combinatorial parameter, point out some connections with communication complexity theory, and examine the vertex packing polytope for some classes of graphs. © 1991 Academic Press, Inc.

1. INTRODUCTION

Many combinatorial optimization problems call for the optimization of a linear function $c'x$ over a discrete set S of solution vectors. For example, in the case of the Traveling Salesman Problem (TSP), $x = (x_{ij})$ is an $\binom{n}{2}$-dimensional variable vector whose coordinates corespond to the edges of the complete graph K_n on n nodes, c is the vector of inter-city distances, and $S \subseteq \{0, 1\}^{\binom{n}{2}}$ is the set of characteristic vectors of the tours of n cities (considered as subsets of the edges of K_n). In the case of the weighted (perfect) matching problem, S is the set of characteristic vectors of the perfect matchings of K_n (n even). These problems are equivalent to: $\min(\max) c'x$ subject to $x \in \text{convex hull}(S)$. The convex hull of the solutions is a polytope, which takes its name from the corresponding problem: the TSP (resp. *matching*) *polytope*. Analogous polytopes have been defined and studied extensively for other common problems: (weighted) bipartite perfect matching (*assignment* polytope), maximum independent set and clique problem (*vertex-packing* and *clique* polytopes), etc.

Optimizing a linear function over a polytope is a linear programming problem. Typically, however, polytopes associated with most combinatorial problems (the assignment polytope is one of the exceptions) have an exponential number of facets. Thus any linear programming formulation in the variables x that defines the polytope has exponential size, and one cannot apply an LP algorithm directly. However, it may be possible (and in some cases true) that the size can be drastically reduced if extra variables and constraints are used.

0022-0000/91 $3.00

Let P be a polytope in the set of variables (coordinates) x. We say that a set of linear constraints $C(x, y)$ in the variables x plus new variables y *expresses* P if the projection of the feasible space of $C(x, y)$ on x is equal to P; i.e., $P = \{x : \text{there is a } y \text{ such that } (x, y) \text{ satisfies } C\}$. Equivalently, C expresses the polytope P iff optimizing any linear function $c'x$ over P is equivalent to optimizing $c'x$ subject to C. We are interested in the question of whether particular polytopes can be expressed by small LP's.

Since linear programming is in P, if one could construct a small (polynomial size LP expressing the polytope of an NP-complete problem, such as the TSP, then it would follow that P = NP. Actually, if anybody believes that P = NP, it is a reasonable approach to try to prove it using linear programming, given the fact that the polynomial algorithms for LP are quite hard and that for a long time linear programming was thought to be outside P. In fact, a recent report was claiming a proof of P = NP this way [Sw]. The proposed LP for the TSP polytope had n^8 (n^{10} in a revision) variables and constraints. With LP's of this size, it is hard to tell what they do or do not express, and clearly, some methodology is needed.

Besides ruling out a possible approach to P = NP, there are several reasons for examining the question of the LP size. There are some problems which we know now to be theoretically in P, but the only method known uses Khachian's ellipsoid algorithm. The ellipsoid algorithm for LP has the advantage that it does not require a complete listing of all the (often exponentially many) constraints. It suffices to have a *separation* algorithm: a polynomial time algorithm which, given a point, decides whether it is feasible and, if it is not, produces a violated constraint. Given the impracticality of the ellipsoid algorithm, it would be desirable to replace it with simplex or Karmakar's algorithm; however, these algorithms need a complete listing of the constraints.

Another use of separation algorithms is in procedures for the TSP based on the polyhedral approach [GP]. Recent progress in this area has increased dramaticaly the size of instances that can be solved optimally in reasonable time [PRi]. The programs are based on a sophisticated combination of branch-and-bound and cutting planes. For cutting planes they use some known simple and "well-behaved" classes of facets. Expressing these facts with a small LP would not imply anything unexpected in theory (such as P = NP), but would permit one to shorcut the repeated generation of cutting planes and solution of the resulting linear programs by solving a single LP.

There may be reason also to look into problems that already have good algorithms: It is reported that implementations of Karmakar's algorithm outperform standard procedures for the assignment problem that are based on the Hungarian method, already at about 100 nodes [J]. In view of this, it would be especially interesting to know whether we can express succinctly also the general (nonbipartite) matching polytope.

After giving some background in Section 2, we examine in Section 3 the matching and the TSP polytopes. We show that these polytopes cannot be expressed by polynomial size *symmetric* LP's. Informally, "symmetric" means that the nodes of the complete graph are treated the same way; see Section 3 for a formal definition. It is not clear what can be gained by treating one node differently than another, but of course this still requires a proof. In Section 4 we reduce the minimum size necessary to express a polytope to a concrete combinatorial problem and point out a relation to communication complexity theory. In Section 5 we examine the vertex packing polytopes of classes of graphs on which the optimization problem can be solved via the ellipsoid algorithm. For one class of graphs we observe that the vertex packing polytope can be expressed by polynomial size LP's; for the class of perfect graphs we use a result from communication complexity to show that their polytopes can be expressed by subexponential linear programs.

2. Preliminaries

We assume familiarity with the basic notions and results from the theory of polyhedra and linear programming [S]. We will be concerned with *rational* polytopes, i.e., polytopes whose vertices have rational coordinates. (In fact, polytopes associated with combinatorial optimization problems have usually vertices with 0–1 coordinates.) The *size* of a rational number p/q (p, q relatively prime) is $\log(|p|+1)+\log(|q|+1)+1$, the size of a (rational) vector is the sum of the sizes of its coordinates. The size of a linear constraint (equation or inequality) is the sum of the sizes of its coefficients, and the size of a set of linear constraints (a linear program) is the sum of the sizes of the constraints; i.e., the size of a LP is roughly the amount of space (in bits) needed to write it down.

If the feasible space of a LP in n variables is a polytope P, then the size of every vertex of P is within a factor $O(n^2)$ of the maximum size of a constraint of the LP. Conversely, if P is a polytope in n-dimensional space, then there is a LP whose feasible space is P and each of whose constraints has size within a factor $O(n^2)$ of the maximum size of a vertex of P (see [S]). Of course, even if the vertices of P have polynomial size (as in the case of polytopes associated with combinatorial optimization problems), the LP itself may not because it needs too many constraints.

Let P be a polytope in variables (coordinates) x. If one wants to express the polytope P by a Linear Program in the variables x, there is very little flexibility. If P is full-dimensional, a nonredundant LP (one in which no constraint can be thrown away) must contain exactly one inequality per facet of the polytope; furthermore, the inequality for each facet is unique up to scalar multiplication. In the case of a lower dimensional polytope, a minimal LP must contain as many equations as the deficit from full dimension (the equations describe the affine hull of P) and exactly one inequality per facet; inequalities that define the same facet may differ more substantially though in this case.

These facts do not exclude the possibility of finding smaller linear programs expressing a polytope by introducing new variables and constraints. We give few examples where this is the case.

EXAMPLES. *Parity polytope.* Let PP be the convex hull of the n-dimensional 0–1 vectors with an odd number of 1's. Optimizing over this polytope is a trivial problem. This polytope has an exponential number of facets (just cutting off the 0–1 vectors with an even number of 1's requires an exponential number of constraints [Je]): for every subset A of $\{1, ..., n\}$ with even cardinality, $\sum_{i \in A} x_i - \sum_{i \notin A} x_i \leqslant |A| - 1$, and $0 \leqslant x_i \leqslant 1$ for all i.

Using new variables, we can express this polytope with a small LP. A vector x is in PP if it can be written as a convex combination $\sum_{k \text{ odd}} \alpha_k y_k$, where each vector y_k is in the convex hull of the 0–1 vectors with k 1's (k odd). It is easy to see that the convex hull of the 0–1 vectors with k 1's is described by the LP: $\sum_i x_i = k$, $0 \leqslant x_i \leqslant 1$ for all i. Thus, the parity polytope is expressed by the LP:

$$\sum_{k \text{ odd}} \alpha_k = 1$$

$$x_i = \sum_{k \text{ odd}} z_{ik} \qquad \text{for all} \quad i = 1, ..., n$$

$$\sum_i z_{ik} = k\alpha_k \qquad \text{for all (odd) } k$$

$$0 \leqslant z_{ik} \leqslant \alpha_k \qquad \text{for all } i, k.$$

The first two sets of constraints say that x can be written as a convex combination $\sum_{k \text{ odd}} \alpha_k y_k$ of vectors $y_k = \langle z_{1k}/\alpha_k, z_{2k}/\alpha_k, ... \rangle$, and the last two sets of constraints say that y_k is in the convex hull of the 0–1 vectors with k 1's. Clearly, for any symmetric Boolean function f, one can construct similarly a small LP that expresses the convex hull of the 0–1 vectors that make f true.

Spanning tree polytope. This is the convex hull of the characteristic vectors of the spanning trees of the complete graph K_n. It is described by the LP:

$$\sum_{i,j} x_{ij} = n - 1$$

$$\sum_{i,j \in S} x_{ij} \leqslant |S| - 1 \qquad \text{for all subsets } S \text{ of nodes}$$

$$0 \leqslant x_{ij} \leqslant 1 \qquad \text{for all } i, j.$$

The spanning tree polytope can be expressed by the following polynomial size LP from [M] (after some obvious simplifications). Introduce an auxiliary variable λ_{kij} for every ordered triple of nodes $k, i, j = 1, ..., n$ with $i \neq j$. The constraints are

$$\sum_{i,j} x_{ij} = n - 1$$

$$\lambda_{kij} + \lambda_{kji} \geqslant x_{ij} \qquad \text{for all} \quad 1 \leqslant i, j, k \leqslant n \quad \text{with} \quad i \neq j$$

$$\sum_j \lambda_{kij} \leqslant 1 \qquad \text{for all} \quad 1 \leqslant i, k \leqslant n \quad \text{with} \quad i \neq k$$

$$0 \leqslant x_{ij} \leqslant 1 \qquad \text{for all } i, j$$

$$\lambda_{kkj} = 0 \quad \text{and} \quad \lambda_{kij} \geqslant 0 \qquad \text{for all } k, i, j.$$

Martin obtained this LP by applying duality to a linear program for the separation problem for the facets of the spanning tree polytope. He presents in [M] a technique by which an LP (of a certain type) for the separation problem of a polytope can be transformed to an LP that expresses the polytope, and he gives some more examples.

There are many other problems that can be expressed by polynomial size LP's [MRC] shows how to do this for problems that can be solved by dynamic programming. For some problems it is very easy to find simple, small LP's if one uses auxiliary variables, but it is much harder (and requires exponential size) to describe them by an LP that uses only the original variables [BP].

It is not known if there are small LP's for matching, and of course for NP-complete problems such as the TSP. Can it be the case that for all polynomial optimization problems (e.g., matching) one can find a small LP, and that, therefore, proving that the TSP polytope cannot be expressed by a small LP would entail $P \neq NP$? After all, we know that linear programming is P-complete [DLR, V]; that is, any problem in P can be written in some sense as a LP. We do not know the answer, but we suspect that it is negative. Linear programming is complete for *decision* problems in P; the $P = NP$? question is equivalent to a weaker requirement of the LP (than that expressing the TSP polytope), in some sense reflecting the difference between decision and optimization problems. Say that a polytope Q in variables z_{ij} $(1 \leqslant i,\ j \leqslant n)$ is a *Hamilton circuit* (HC) polytope if it includes the characteristic vectors of Hamiltonian graphs and excludes non-Hamiltonian (considered again as subsets of the edges of the complete graph). For example, if an LP expresses the TSP polytope in variables x_{ij}, then the polytope obtained by adding constraints $x_{ij} \leqslant z_{ij}$ and then projecting the feasible space on z is a HC polytope.

PROPOSITION. *NP has polynomial size circuits (resp. $P = NP$) if and only if (for every n) there is a polynomial size LP (resp., that can be constructed efficiently) which expresses a HC polytope.*

Proof. The one direction follows from the fact that linear programming is in P. The other direction follows Valiant's proof that linear programming is complete for P under p-projections [V]. Suppose that NP has polynomial size circuits and take such a circuit for the Hamilton circuit problem with inputs z_{ij}, $1 \leqslant i,\ j \leqslant n$. Introduce a variable for every gate. For a gate $g = \neg u$, include constraints $0 \leqslant g = 1 - u \leqslant 1$; for a gate $g = u \wedge v$, include constraints $0 \leqslant g \leqslant u \leqslant 1$, $0 \leqslant g \leqslant v \leqslant 1$, $g \geqslant u + v - 1$; for a gate $g = u \vee v$, include constraints $0 \leqslant u \leqslant g \leqslant 1$, $0 \leqslant v \leqslant g \leqslant 1$, $g \leqslant u + v$. Finally, if g is the output gate, include the equation $g = 1$. As in [V], it is easy to see that if the variables z_{ij} have values 0 and 1, then all other variables are also forced to have value 0 or 1 equal to the truth value of the corresponding gate; the final equation is satisfied iff the graph is Hamiltonian. Thus, the projection of the LP on the variables z includes the Hamiltonian graphs and excludes the nonHamiltonian. ∎

Of course, the same observation applies to all decision problems. If $\Pi \subseteq \{0, 1\}^*$ is a decision problem (language) in P, then for every n we can construct a polynomial size LP that expresses a polytope which includes the 0–1 vectors of length n that are in Π, and excludes those that are not in Π. Note the importance of allowing new variables here. Without new variables, linear programming is extremely weak for decision problems; even such trivial problems as parity require exponential size. With new variables, linear programming achieves the full power of P-time decision algorithms.

This may not be the case for optimization problems. That is, solving an optimization problem by expressing it as a small LP may be a restriction in the model of computation, as, for example, monotone Boolean circuits form a restriction for (monotone) decision problems. Even so, it is a natural restriction (many optimization problems are solved by formulating them as linear programs), and it has some power. For example, in the case of the bipartite perfect matching problem, the corresponding decision problem requires superpolynomial size for monotone circuits [R], while the optimization problem is very easy for linear programming, even without using new variables: the convex hull of the perfect matchings of the complete bipartite graph $K_{n,n}$ with n nodes on each side is described by the LP $\sum_j x_{ij} = 1$ for all $i = 1, ..., n$; $\sum_i x_{ij} = 1$ for all $j = 1, ..., n$; $x_{ij} \geqslant 0$ for all i, j.

3. The Matching and the TSP Polytopes

Although we usually speak of *the* TSP (or matching) polytope, there is one for every size n. Thus also, when we say that an LP expresses such a polytope, we mean again one LP for every n.

A complete description of the matching polytope (in terms of the standard variables x_{ij}) was found by Edmonds [E]: $\sum_j x_{ij} = 1$ for all i; $\sum_{i \in S, j \notin S} x_{ij} \geqslant 1$ for all odd subsets S of nodes; and $x_{ij} \geqslant 0$.

A complete description of the TSP polytope is not known (and probably will never be). This polytope has many complex facets [GP, PW, PY]. However, several simple and useful classes of facets have been identified and are used in procedures for the TSP. First are the obvious constraints: $0 \leqslant x_{ij} \leqslant 1$, $\sum_j x_{ij} = 2$ for all i. Another easy class are the *subtour elimination constraints* (SECs): $\sum_{i \in S, j \notin S} x_{ij} \geqslant 2$ for all nonempty proper subsets S of nodes. Next come the *2-matching constraints*: If F is a set of $2k+1$ edges and S a set of nodes that contains exactly one node from every edge of F, then $\sum_{i,j \in S} x_{ij} + \sum_{[i,j] \in F} x_{ij} \leqslant |S| + k$. These are followed by further generalizations, *comb constraints* and *clique-tree* inequalities. We will not define these constraints here; see [GP] for a comprehensive treatment.

The subtour elimination constraints can be easily expressed by a polynomial size LP. One way is based on the separation algorithm for these constraints. Viewing the x_{ij}'s as capacities on the edges, the SECs state that the minimum cut in the graph has capacity at least 2. From the max flow–min cut theorem, this can be expressed in the obvious way introducing appropriate flow variables and constraints. A different and less obvious way uses a small portion of the LP described in [Sw]. Introduce variables v_{ij} and y_{rijk} with the following intended meaning for a Hamilton tour: Orient the tour in one of the two ways. Variable v_{ij} (where i, j is now an *ordered* pair of nodes) has value 1 if the tour traverses the directed edge (i, j), and 0 otherwise; y_{rijk} is 1 if the directed edge (i, j) is the kth edge of the tour starting from node r, and 0 otherwise. Some constraints that are clearly consistent with this interpretation are: $x_{ij} = v_{ij} + v_{ji}$ for all i, j; $\sum_j v_{ji} = \sum_j v_{ij} = 1$ for all i; $\sum_r y_{rijk} = v_{ij}$ for all i, j, k; $\sum_k y_{rijk} = v_{ij}$ for all i, j, r; $\sum_i y_{rijk} = \sum_i y_{rji,k+1} = \sum_i y_{jri,n-k}$ for all r, j, k; $y_{rijk} \geqslant 0$ for all r, i, j, k. It is not clear what these variables and constraints accomplish exactly, but it can be shown that they too imply the SEC's (and in fact make deeper cuts). Although both of these LP's have polynomial size, they are rather large. It would be of interest if there are alternative smaller LP's as the SCs provide usually good lower bounds, within a few percentage points of the true integer optimum [J].

The constraints of the matchig polytope look similar to the SECs, apart from the parity of S. In fact, their separation algorithm is a minor modification of the one for the SECs (the proof is nontrivial—see [PRa]). The same is true of the 2-matching constraints, the next set of facets of the TSP. Despite this similarity, these constraints are apparently harder to express.

A permutation (relabelling) π of the nodes of the complete graph also induces a permutation of the edges and their corresponding variables: x_{ij} is mapped to $x_{\pi(i)\pi(j)}$. It also maps one perfect matching (or tour) into another, and induces a rotation of the coordinates that leaves the matching and the TSP polytopes invariant. We say that a polytope $P(x, y)$ over variables $x = (x_{ij})$ and new vaiables y is *symmetric* if every permutation π of the nodes can be also extended to the new variables y so that P remains invariant. A LP (set of linear constraints) is called symmetric if its feasible space is. Clearly, if a set of constaints "looks" symmetric (permutation of the variables gives the same LP) then so is its feasible space, but not conversely; a LP that does not look symmetric may describe a symmetric polytope.

The assumption of symmetry is a natural one when one tries to construct a linear program. First, as we observed in the previous section, one can construct a small LP for any polytope with 0–1 vertices which is invariant under *all* permutations of variables (i.e., whose vertices are the 0–1 vectors that make a symmetric Boolean function f true); clearly, this LP is "symmetric," in the sense that every permutation of the original variables can be extended to the new variables. With a small symmetric LP we can distinguish those *bipartite* graphs that have a perfect matching from those that do not: simply add constraints $x_{ij} \leqslant z_{ij}$ to the LP expressing the bipartite perfect matching polytope (see end of Section 2) and consider the projection on z. The linear program of the previous section for the spanning tree polytope, and the LP's for the subtour elimination constraints that we described earlier, are obviously symmetric and so is the full linear program for the TSP proposed in [Sw].

THEOREM 1. *The matching polytope cannot be expressed by a symmetric LP of subexponential size.*

Proof. The lower bound of the theorem applies to the number of variables and constraints of the LP; that is, we will ignore the sizes of the coefficients. Before going into the details of the proof, we give first a brief, informal outline. The proof consists of the following four steps.

1. Transform to a symmetric LP in standard form (equality constraints plus nonnegativity of variables).
2. Show that every variable of the LP "depends" on "few" nodes. (We will define formally the terms later, as we go along.)
3. Reduce to an LP with a specific set of variables, which is at least as powerful.
4. Show that this LP does not work.

Step 1. Let P be a symmetric polytope in the variables $x=(x_{ij})$ and new variables y. We will use the shorthand z for the vector $\left[\begin{smallmatrix}x\\y\end{smallmatrix}\right]$ of all the variables. A minimal LP description of P consists of a set of equality constraints $Az=b$ describing the affine hull of P, and a set of inequality constraints $c_iz\leqslant d_i$; $i=1, ..., r$, one for every facet of P. Take such a minimal LP description where the c_i's are in the linear space parallel to the affine hull of P (i.e., they are orthogonal to the rows of A) and are normalized in the L_∞ metric, $\|c_i\|=1$. Note that these conditions determine uniquely the inequality constraints (the c_i's and d_i's) [S]. Add now slack variables $u_1, ..., u_r$, and consider the Linear Program L': $Az=b$, $c_iz+u_i=d_i$, $u_i\geqslant 0$ for $i=1, ..., r$. We claim that L' is also symmetric. To see this, it suffices to show that every permutation g of the variables z which leaves P invariant can be extended to the new variables u so that it leaves also the feasible space of L' invariant. Let g be a permutation of z such that $g(P)=P$. Then, (1) g maps the affine hull of P onto itself, and (2) permutes the facets of P. Extend g to the slack variables u_i by letting it permute them the same way as it permutes the corresponding facets. Note that g maps the hyperplane $c_iz=d_i$ into the hyperplane $g(c_i)z=d_i$, where $g(c_i)$ is the vector obtained from c_i by permuting its coordinates according to g. Clearly, $g(c_i)$ is also normalized and in the linear space parallel to the affine hull of $g(P)=P$. From the uniqueness of the inequalities in the LP description we conclude that, if g maps the ith facet of P into the jth facet, then we must have $g(c_i)=c_j$ and $d_i=d_j$. It follows that the extension of g leaves also the feasible space of L' invariant.

Clearly, the size of L' is linear in the size of any LP description of P, and the projection of L' on x is equal to the projection of P. The linear program L' is not quite in standard form because not all of the variables are constrained to be nonnegative. Although it is not important for the rest of the proof, it will simplify the notation to have all the variables constrained. Regarding the variables x, we may assume that they are nonnegative in any feasible solution of L', because otherwise the projection on x would not equal the matching polytope. Thus, we may add the constraints $x \geqslant 0$ to L' without changing the feasible space. We can take care of the variables y in the standard way: use variables y^+ and y^-, replace y in all the constraints with $y^+ - y^-$, and add the constraints $y^+ \geqslant 0$, $y^- \geqslant 0$. It is straightforward to verify that the new LP is also symmetric and has the same projection on x.

Step 2. We assume from now on that our symmetric LP is in standard form: $Az = b$; $z \geqslant 0$, where $z = [{}^x_y]$. We will argue in the rest of the proof that the LP cannot express the matching polytope unless it has an exponential number of variables. More precisely, we will suppose from now on that the LP has at most $\binom{n}{k}$ variables, for an integer $k < n/4$, and prove that they are not enough.

We may assume that for every perfect matching M of the complete graph, there is a feasible solution whose projection on x is the characteristic vector of M. Suppose that $\{z^*(M)\}$ is a family of such solutions, one for each perfect matching M. We say that a variable z_i *depends* on a set S of nodes with respect to the family $\{z^*(M)\}$ of solutions if for all permutations π of the nodes that fix the nodes of S and for all perfect matchings M, we have $z_i^*(M) = z_i^*(\pi(M))$, where $\pi(M)$ is the perfect matching $\{[\pi(a), \pi(b) \mid [a, b] \in M\}$. In other words, if we take any perfect matching M and permute the labels of the nodes outside S to obtain another matching $\pi(M)$, the value of z_i is not affected. For example, the variable x_{ij} depends on the nodes i and j. We shall use the symmetry assumption in this step to show that we can choose a family $\{z^*(M)\}$ of solutions, one for each perfect matching, so that every variable depends on at most k nodes (with respect to these solutions).

Let G be the set of permutations g of the variables x, y which (1) leave the feasible space invariant, and (2) extend some permutation π of the nodes (i.e., $g(x_{ij}) = x_{\pi(i)\pi(j)}$). It is easy to see that G is a group, every permutation g in G extends a unique permutation π of the nodes, and there is a natural homeomorphism from G to S_n, the symmetric group of permutations of the n nodes. By our hypothesis that the LP is symmetric, the homeomorphism maps G *onto* S_n. We choose the family of solutions corresponding to perfect matchings as follows. First, take any particular perfect matching M_0, and let $z(M_0)$ be any feasible solution whose projection on x is the characteristic vector of M_0. Let $G_0 = \{g \in G \mid g$ extends a permutation π such that $\pi(M_0) = M_0\}$. Define $z^*(M_0) = \sum_{g \in G_0} g(z(M_0))/|G_0|$. Note that for every $g \in G_0$, the vector $g(z(M_0))$ is a feasible solution (since $g \in G$), and its x-projection is also equal to the characteristic vector of M_0 (since g extends a π with $\pi(M_0) = M_0$). Therefore, the same properties are true of their convex combination $z^*(M_0)$. For any other perfect matching M, take a permutation τ such that $\tau(M_0) = M$ (obviously, there is such a τ), take a $f \in G$ that extends τ (there is one by the symmetry of the LP), and define $z^*(M) = f(z^*(M_0))$.

CLAIM 1. *For any perfect matching M, any permutation π of the nodes, and any extension g in G of π, we have $z^*(\pi(M)) = g(z^*(M))$.*

Proof. We will prove first the claim for the case that M is M_0 (the perfect matching we picked initially), and π maps M_0 to itself. In this case, the right-hand side is $g(z^*(M_0)) = \sum_{h \in G_0} g(h(z(M_0))/|G_0|$. Since π maps M_0 to itself, g is in G_0. Also, it is easy to see that G_0 is a group. Thus, as h ranges over all member of G_0, so does $g \cdot h$. Therefore, the right-hand side is equal to $\sum_{h \in G_0} h(z(M_0))/|G_0| = z^*(M_0)$, the left-hand side in this case.

Consider now the general case. If $M \neq M_0$, then let τ be the permutation of the nodes and f its extension in G that were used in defining $z^*(M)$; otherwise $(M = M_0)$ let τ and f be the identity permutations. In either case, $z^*(M) = f(z^*(M_0))$ and $M = \tau(M_0)$. Defining similarly the permutations σ and h for the perfect matching $\pi(M)$, we have: $z^*(\pi(M)) = h(z^*(M_0))$ and $\pi(M) = \sigma(M_0)$. The permutation $h^{-1}gf$ of the variables extends the permutation $\sigma^{-1}\pi\tau$ of the nodes that maps M_0 to itself. From the first case we have: $z^*(M_0) = h^{-1}gf(z^*(M_0))$. Therefore, $z^*(\pi(M)) = h(z^*(M_0)) = g(f(z^*(M_0))) = g(z^*(M))$. ∎

For a variable z_i, define $H(z_i)$ to be the group of permutations π of the nodes such that for all perfect matchings M, $z_i^*(\pi(M)) = z_i^*(M)$. In words, a permutation π of the nodes is in $H(z_i)$ if when we take any perfect matching M and relabel the nodes according to π to obtain another matching $\pi(M)$, the value of the variable z_i does not change (with respect to our chosen family of perfect matching solutions.) It is easy to see that $H(z_i)$ is indeed a group. Note that according to the definitions, z_i depends on a set S of nodes iff $H(z_i)$ contains all permutations that fix the nodes of S.

We shall show now that the index of $H(z_i)$ in the symmetric group S_n is no more than the number of variables. Clearly, the number of variables is at least as large as the orbit of z_i under G, which is equal to the index of the stabilizer of z_i in G. If a member g of G fixes z_i and extends a permutation π of the nodes, then from Claim 1 we have that for any perfect matching M, $z_i^*(\pi(M))$ is equal to the ith component of $g(z^*(M))$, which is simply $z_i^*(M)$, since g fixes z_i. That is, if $g \in G$ fixes z_i and g extends π, then $\pi \in H(z_i)$. In the homeomorphism from G to S_n, the stabilizer of z_i is mapped into (a possibly proper subgroup of) $H(z_i)$. Therefore, the index of $H(z_i)$ in S_n is no larger than the index of the stabilizer of z_i in G, which in turn is no larger than the number of variables.

The following claim must be known (and is true for all values of k).

Claim 2. *Let H be a group of permutations on a set N of n nodes and suppose that its index in S_n is at most $\binom{n}{k}$, where $k < n/4$. Then there is a set S of at most k nodes such that H contains all even permutations that fix the nodes of S.*

Proof. Let B_1 be the largest orbit of H (if H is transitive then $B_1 = N$) and B_2 the rest of the nodes. Since $|H| \geqslant k!(n-k)!$, the number m of nodes of B_1 is at least $n-k$. The group H is the subdirect product of a group H_1 acting (transitively) on B_1 with a group H_2 acting on B_2. Suppose that H_1 is not primitive and take any complete system of r blocks of imprimitivity with s nodes each, where $sr = m = |B_1|$ and $s, r \geqslant 2$. Then $|H| \leqslant r!(s!)^r\,|H_2|$. It is easy to see that the right-hand side of this inequality is at most $2((n/2)!)^2$ (achieved when $r=2$, $s=n/2$, and B_2 is empty). Therefore, H_1 is primitive.

Let F be the group of the permutations on B_1 that combine (in H) with the identity on B_2. F is a normal subgroup of H_1, it is transitive (because H_1 is primitive), and $|H| = |F|\,|H_2|$. As in the case of H_1, F must be primitive. The index of a primitive subgroup of S_m is at least $(m+\frac{1}{2})!$, unless it is the symmetric group S_m itself or the alternating group A_m (see [W, Theorem 14.2]). It follows that F is S_m or A_m. ∎

Suppose that $H(z_i)$ contains all even permutations that fix the nodes of S; then we claim that it must also contain all odd permutations. For, suppose there is an odd permutation π which fixes the nodes of S and is not in $H(z_i)$. This means that there is a perfect matching M such that $z_i^*(\pi(M)) \neq z_i^*(M)$. Let τ be the transposition of two nodes in $N-S$ that are matched in M; clearly, they exist because $|S| < n/4$. Then $\sigma = \pi \cdot \tau$ is an even permutation which also fixes S, and $\sigma(M) = \pi(\tau(M)) = \pi(M)$. Since $z_i^*(\sigma(M)) \neq z_i^*(M)$, σ should not be in $H(z_i)$.

We conclude that if there are less than $\binom{n}{k}$ variables, then every variable depends on at most k nodes; in particular, if there is a polynomial number of variables then every variable depends on a bounded number of nodes.

Step 3. We assume from now on that every variable z_i depends on (at most) $k < n/4$ nodes (with respect to our chosen set of perfect matching solutions $z^*(M)$). We will define a new set of variables w_J, and show that if the matching polytope is expressed by our standard form LP $Az = b$; $z \geqslant 0$, then it is aso expressed by a standard form LP in the variables w_J.

For every set J of at most k independent edges (i.e., a partial matching) we have a variable w_J. The value of the variable for a perfect matching M is defined as follows: $w_J^*(M) = 1$ if $J \subseteq M$, and 0 otherwise. The standard variables x_{ij} are identified with the variables w_J, where J is a singleton. We show first that, as far as the perfect matching solutions $\{z^*(M)\}$ and $\{w^*(M)\}$ are concerned, every z variable can be written as a positive combination of some w variables.

CLAIM 3. *There is a nonnegative matrix B such that for all perfect matchings M, $z^*(M) = Bw^*(M)$.*

Proof. Let V_i be the set of (at most k) nodes on which variable z_i depends. Suppose that M_1 and M_2 are two perfect matchings which agree on the edges that cover V_i (i.e., the nodes of V_i have the same mates in M_1 and M_2). Then, clearly, there is a permutation π which fixes the nodes of V_i and maps M_1 to M_2. Therefore, $z_i^*(M_2) = z_i^*(\pi(M_1)) = z_i^*(M_1)$.

A row of B corresponds to a variable z_i, and a column corresponds to a variable w_J. The row of B that corresponds to a standard variable x_{ij} has 0 everywhere, except for the column that corresponds to the w_J variable identified with x_{ij} (i.e., with $J = \{[i\,j]\}$), where it has a 1. In general, the entry B_{iJ} corresponding to variables z_i and w_J is defined as follows: If J covers V_i and is minimal with respect to this property (i.e., every edge of J is incident to a node of V_i), then we take any perfect matching M containing J and we let $B_{iJ} = z_i^*(M)$; by our previous observation, this value does not depend on the choice of M. In the contrary case, i.e., J does not cover minimally V_i, we let $B_{iJ}=0$.

Let M be any perfect matching. Consider a variable z_i, and let J be the set of edges of M that are incident to V_i. Since $|V_i| \leqslant k$, also $|J| \leqslant k$, and there is a variable w_J. Note that for any other set I of edges, either I does not cover minimally V_i, in which case $B_{iI}=0$, or I is not contained in M, in which case $w_I^*(M)=0$. Therefore, the inner product of the row B_i of B (corresponding to variable z_i) and $w^*(M)$ has at most one nonzero term: the term $B_{iJ}w_J^*(M)$. Thus, $z_i^*(M)=B_i w^*(M)$ for every variable z_i, proving the claim. ∎

Let (L1) be our LP $Az=b$; $z \geqslant 0$, and consider the LP (L2): $ABw=b$; $w \geqslant 0$. For every perfect matching M, since $z^*(M)$ is feasible in (L1), $w^*(M)$ is feasible in (L2) by Claim 3. Thus, the projection of (L2) on x (i.e., the w-variables identified with the standard variables x_{ij}) contains the matching polytope. On the other hand, if a vector w is feasible in (L2), then Bw is feasible in (L1) because B is a nonnegative matrix. Clearly, the vectors w and Bw have the same projection on x. Thus, the projection of (L2) on x is contained in the projection of (L1). We conclude that, if (L1) expresses the matching polytope, then so does (L2).

Step 4. Consider now a standard form LP in the variables w_J with $|J| \leqslant k < n/4$, and suppose that for every perfect matching M the vector $w^*(M)$ is feasible. Then, any vector $w \geqslant 0$ which can be written as an affine combination of the $w^*(M)$'s is a feasible solution as well.

Partition the nodes into two sets S_1 and S_2, where $|S_1|=2k+1$ and $|S_2|=n-(2k+1)$; since $n>4k$ and n is even, also S_2 has odd cardinality at least $2k+1$. We shall construct an affine combination $\tilde{w}$ of the $w^*(M)$'s such that (1) $\tilde{w} \geqslant 0$ and (2) for every edge $[i,j]$ that goes from S_1 to S_2, the value of the corresponding variable x_{ij} in $\tilde{w}$ is 0. This means that $\tilde{w}$ is feasible, and its projection on x is not in the matching polytope, since it violates the constraint $\sum_{i \in S_1, j \notin S_1} x_{ij} \geqslant 1$.

For each odd $i=1,3,...,2k+1$, let $\bar{w}(i)$ be the average of the vectors $w^*(M)$, where M ranges over all perfect matchings with exactly i edges from S_1 to S_2. Note that $\bar{w}(i)$ is feasible, because it is a convex combination of feasible solutions. Let $\tilde{w}$ be the vector $\tilde{w}=c_1\bar{w}(1)+c_3\bar{w}(3)+\cdots+c_{2k+1}\bar{w}(2k+1)$, where the c_i's satisfy the following system (E) of linear equations:

$$c_1+c_3+\cdots+c_{2k+1}=1. \tag{0}$$

For $j=1$ to k:

$$\sum_{i \geqslant j}\binom{i}{j}c_i=0. \tag{j}$$

CLAIM 4. *The matrix of the linear system* (E) *is nonsingular, and therefore,* (E) *has a solution.*

Proof. For each $j=0$ to k, the jth row of the matrix consists of the values of a degree j polynomial $P_j(u)$ on the $k+1$ points $u=1,3,...,2k+1$; namely, $P_0 \equiv 1$, and for $j \geq 1$, $P_j(u) = u(u-1) \cdots (u-j+1)/j!$. It follows that the matrix is nonsingular, and (E) has a (unique) solution. ∎

Thus, $\tilde{w}$ is well defined. Observe that, because of Eq. (0), the vector $\tilde{w}$ is an affine combination of feasible solutions and therefore satisfies all equality constraints.

CLAIM 5. *$\tilde{w}_J = 0$ for every set J of (at most k independent) edges that go from S_1 to S_2.*

Proof. Because of the obvious symmetry in the construction of $\tilde{w}$, if I and J are two sets of edges that go from S_1 to S_2 and have the same cardinality, then $\tilde{w}_I = \tilde{w}_J$. (In general, if π is a permutation of the nodes that maps S_1 to itself and S_2 to itself, then for every set J of edges, $\tilde{w}_J = \tilde{w}_{\pi(J)}$.) For $j \leq k$, let s_j be the sum of the components $\tilde{w}_J$, where J ranges over all sets of j independent edges from S_1 to S_2. It suffices to show that $s_j = 0$.

If M is a matching with exactly i edges from S_1 to S_2, then the corresponding sum for $w^*(M)$ is $\binom{i}{j}$ if $i \geq j$, and 0 if $i < j$. Therefore, the same is true for $\tilde{w}(i)$, a convex combination of such matchings. Thus, $s_j = \sum_{i \geq j} \binom{i}{j} c_i = 0$ by Eq. (j). ∎

CLAIM 6. $\tilde{w} \geq 0$.

Proof. We shall show that for every set J of at most k independent edges, $\tilde{w}_J \geq 0$, and furthermore, if J has at least one edge from S_1 to S_2, then $\tilde{w}_J = 0$. We use induction on the number t of edges of J that do not go from S_1 to S_2. The basis, $t = 0$, follows from Claim 5. For the induction step, let $[a, b]$ be an edge of J that does not go from S_1 to S_2, say both a and b are in S_1, and let I be the rest of the edges of J. Suppose first that $I = \emptyset$; that is, $J = \{[a, b]\}$, and w_J is actually the variable x_{ab}. All solutions $w^*(M)$, corresponding to the perfect matchings, satisfy the equation $\sum_{c \neq a} x_{ac} = 1$; therefore, also their affine combination $\tilde{w}$ satisfies the same equation. By Claim 5, for $c \in S_2$, the value of x_{ac} in $\tilde{w}$ is 0. By the symmetry in the construction of $\tilde{w}$, for all $c \in S_1$ with $c \neq a$, x_{ac} has the same value, which therefore must be $\tilde{w}_J = 1/(|S_1| - 1) > 0$.

Suppose now that $I \neq \emptyset$. For every perfect matching M, the solution $w^*(M)$ satisfies the equation $w^*_I(M) = \sum_c w^*_{I \cup \{[a,c]\}}(M)$, where c ranges in the sum over all nodes for which the variable is defined (i.e., c is a node different than a and the nodes of I). To see this, observe that, either M does not contain I, in which case all terms on both sides of the equation are 0, of M contains I, in which case the left-hand side is 1, and on the right-hand side all terms are 0 except for the one where c is the mate of node a in M. Since $\tilde{w}$ is an affine combination of the $w^*(M)$'s, it satisfies te same equation. For $c \in S_2$, the set $I \cup \{[a, c]\}$ has fewer edges (than J) that do not go from S_1 to S_2, and has at least one edge (namely, $[a, c]$) from S_1 to S_2; thus, by the induction hypothesis, the corresponding component of $\tilde{w}$ is 0. Therefore, $\sum_{c \in S_1} \tilde{w}_{I \cup \{[a,c]\}} = \tilde{w}_I$. By the symmetry in the construction of $\tilde{w}$, all terms in the sum are equal to each other, and in particular to $\tilde{w}_J$. By the induction hypothesis, $\tilde{w}_I \geq 0$, and therefore $\tilde{w}_J \geq 0$. If J has an edge from S_1 to S_2, then this edge is also in I; from the induction hypothesis $\tilde{w}_I = 0$, and thus also $\tilde{w}_J = 0$. ∎

This completes Step 4 and the proof of Theorem 1. ∎

Consider a LP in the standard varibles x and new variables y. The *support* of a feasible solution is the graph consisting of the edges $[i, j]$ such that $x_{ij} > 0$ in the solution. If the LP expresses the matching polytope, then for every feasible solution, its x-projection is a convex combination of the characteristic vectors of perfect matchings. Thus, certainly its support must have a perfect matching. In the proof of Theorem 1 we have actually shown the following.

COROLLARY 1. *A symmetric LP of subexponential size, whose projection contains the matching polytope, has a feasible solution whose support is a graph that does not have a perfect matching.*

We can argue as in the proof of Theorem 1 to show a similar result for the TSP polytope. Although a direct proof is possible, it is somewhat more complicated; it is much easier to use a reduction from matching.

THEOREM 2. *The TSP polytope cannot be expressed by a symmetric LP of subexponential size.*

Proof. Consider a graph G with $6n$ nodes which are partitioned into three equal size sets $L = \{l_1, ..., l_{2n}\}$, $M = \{m_1, ..., m_{2n}\}$, $R = \{r_1, ..., r_{2n}\}$. The subsets L, R induce complete subgraphs and, in addition, each node m_i is connected to l_i and r_i. Think of the complete graph induced by L as an instance for the matching polytope, and suppose we have a symmetric LP C for the TSP on $6n$ nodes.

A permutation of L induces an automorphism of G. Therefore, setting in C all variables x_{ij} to 0 for the missing edges $[i, j]$, we get another LP C' which is symmetric with respect to L. Since C expresses the TSP polytope on $6n$ nodes, the LP C' expresses the convex hull of the characteristic vectors of the Hamilton circuits of G, denoted TSP(G). Since the nodes m_i have degree 2, a Hamilton circuit of G consists of their incident edges and perfect matchings from L and R. Also, every perfect matching of L can be extended to a Hamilton circuit of G. Therefore, the matching polytope for L is a projection of TSP(G). It follows that C' expresses the matching polytope on $2n$ nodes. ∎

COROLLARY 2. *A symmetric LP of subexponential size whose projection contains the TSP polytope has a feasible solution* (1) *whose support is a nonHamiltonian graph, and* (2) *which violates a 2-matching constraint.*

Proof. Let C and C' be the linear programs as in the proof of Theorem 2. We may assume without loss of generality that every solution of C satisfies the constraints $0 \leqslant x_{ij} \leqslant 1$ for every edge $[i, j]$, and $\sum_j x_{ij} = 2$ for every node i; if not, then we can just add these constraints. From the proof of Theorem 1, we know that there is a feasible solution z of C' (and thus, also of C) and an odd subset S of L such that the support of z does not contain any edge from S to $L-S$. It follows that the support of z is a nonHamiltonian subgraph of G.

Consider the 2-matching constraint that corresponds to the set S of nodes and the set of edges $F=\{[l_i, m_i] : l_i \in S\}$. Since m_i has degree 2 in G, for every edge e in F the corresponding variable x_e has value 1 in z (a solution to C'). Thus, the left-hand side of the 2-matching constraint is $\sum_{e \subseteq S} x_e + \sum_{e \in F} x_e = \frac{1}{2}[\sum_{s \in S} \sum_t x_{st} + \sum_{e \in F} x_e] = 3\,|S|/2$. ∎

The proof of Theorem 1 gives a lower bound for a class of LP's somewhat larger than the symmetric class. Suppose that we have an LP $Az=b$; $z \geqslant 0$ with a family of solutions $\{z^*(M)\}$ for the perfect matchings and that, with respect to these perfect matching solutions, every variable can be written as a positive combination of some variables that depend on few nodes (or of variables w_J) in the sense of Claim 3 in the proof. Then the rest of the proof goes through as before. For example, let us say that a variable z_i *concerns* a set S of nodes (with respect to the family $\{z^*(M)\}$ of perfect matching solutions), if the variable has the same value for any two matchings M_1 and M_2 that agree on the edges that cover S (i.e., $z_i^*(M_1)=z_i^*(M_2)$). From steps 3 and 4 of the proof of Theorem 1, if every variable concerns $k<n/4$ nodes, then the LP does not express the matching polytope and has a feasible solution whose support does not have a perfect matching. Note that, if a variable depends on a set S of nodes, then it also concerns S, but the converse is not true; for example, a variable z_i whose interpretation is "the label of the mate of node i" (i.e., $z_i^*(M)=j$, where $[i,j] \in M$) is sensitive to the labelling of all the nodes, but concerns only node i. As an illustration, we shall show the lower bound can be transferred through a reduction that is not symmetric (does not preserve the symmetries of the complete graph as the reduction of Theorem 2 does).

COROLLARY 3. *A polynomial size symmetric LP, whose projection contains the matching polytope, has a feasible solution whose support is a graph of maximum degree* 3 *that does not have a perfect matching.*

Proof. Consider the following reduction of the matching problem from general graphs to degree 3 graphs. Given a graph G with n nodes, construct a graph $\bar{G}$ as follows. For every node i of G take a complete binary tree with at least n leaves and insert a node in the middle of every edge; let T_i be the resulting tree. The graph $\bar{G}$ has a tree T_i for every node i of G, and for every edge $[i,j]$ it has an edge connecting the jth leaf of T_i to the ith leaf of T_j.

Suppose that M is a perfect matching of G. Construct a perfect matching $\bar{M}$ of $\bar{G}$ as follows. If M contains the edge $[i,j]$, then $\bar{M}$ contains the corresponding edge connecting the jth leaf of T_i to the ith leaf of T_j. Within the tree T_i, an internal degree 2 node is matched to its father if it lies on the path from the root to the jth leaf (the one that is matched to another tree) and is matched to its child otherwise. Conversely, suppose that $\bar{M}$ is a perfect matching of $\bar{G}$. We claim that there is exactly one matched edge coming out of every T_i, and thus, $\bar{M}$ corresponds to a perfect matching M of G. Note that in each tree T_i there is a unique alternating path that starts at the root with a matched edge (because of the degree 2 nodes); this path must lead to a leaf that is matched to a node outside T_i. It is easy to see that every internal degree 2 node that is not on this path must be matched to its child (use induction on the level of the node). Therefore, every other leaf is matched within T_i.

Let $\bar{K}_n$ be the degree 3 graph that results when we apply this transformation on the complete graph K_n of n nodes. Let N be the set of nodes of K_n and $\bar{N}$ of $\bar{K}_n$, and let $m = |\bar{N}| = O(n^2)$. Note that the reduction is not symmetric, it depends on how the nodes of K_n are numbered. Suppose that we have a symmetric LP (without loss of generality in standard form) $L: Az = b$; $z \geqslant 0$ with at most $\binom{m}{k}$ variables, $k < n/4$, and that its projection on the x variables contains the matching polytope on m nodes. We know that we can pick a family $\{z^*(\cdot)\}$ of solutions for the perfect matchings on m nodes so that every variable z_l concerns at most k nodes of $\bar{N}$. Let L' be the LP obtained from L by setting to 0 the x variables corresponding to the edges missing from $\bar{K}_n$. Regard L' as an LP for the matching polytope on n nodes by identifying the x_{ij} variable for the edge $[i, j]$ of K_n with the x variable of L' for the corresponding edge of $\bar{K}_n$, i.e., the edge connecting the jth leaf of T_i with the ith leaf of T_j. For every perfect matching M of K_n, choose $z^*(\bar{M})$ as its feasible solution in L', where $\bar{M}$ is the matching of $\bar{K}_n$ that corresponds to M. We claim that every variable concerns at most k nodes of K_n. In proof, suppose that a variable z_l concerns a set $\bar{S}$ of nodes from $\bar{N}$. Every node of $\bar{S}$ belongs to some tree T_i corresponding to a node i of K_n. Let S be the set of nodes of K_n that correspond to the trees that contains the nodes of $\bar{S}$; since $|\bar{S}| \leqslant k$, also $|S| \leqslant k$. Suppose that two perfect matchings M_1, M_2 of K_n agree on the edges that cover the nodes of S. Then the corresponding matchings $\bar{M}_1$, $\bar{M}_2$ of $\bar{K}_n$ agree on the edges that cover the nodes of $\bar{S}$, and therefore the variable z_l has the same value in the solutions for the two matchings.

Since every variable concerns at most k nodes of K_n, there is a feasible solution $\tilde{z}$ of L' whose support G in K_n does not have a perfect matching. The support of $\tilde{z}$ in K_m is a subgraph $\bar{G}$ of $\bar{K}_n$ that does not contain the edges that correspond to edges missing from G, and thus $\bar{G}$ does not have a perfect matching either. ∎

Along the same lines, one can show an analogous result for the TSP using the reduction in [GJS] of the Hamilton circuit problem to degree 3 graphs.

4. A Combinatorial Parameter

Recall that, if we want to describe a polytope by a LP in the standard variables, there is very little flexibility. However, if one may use any new variables and constraints one wishes, there is an unlimited number of possibilities. We will provide a combinatorial characterization of the number of variables and constraints needed, which may help to get some handle on this problem.

Let P be a polytope in n-dimensional space with f facets and v vertices. Define a matrix SM (for *slack matrix*) for P whose rows correspond to the facets, and the columns correspond to the vertices. Pick an inequality (anyone) for each facet. The ijth entry of SM is the slack of the jth vertex in the inequality corresponding to the ith facet, That is, if the inequality is $c_i x \leqslant d_i$ and the jth vertex is x^j, then $SM[i, j] = d_i - c_i x^j$. Note that SM is a nonnegative matrix.

THEOREM 3. *Let m be the smallest number such that SM can be written as the product of two nonnegative matrices of dimensions $f \times m$ and $m \times v$. The minimum of the number of variables plus number of constraints over all LP's expressing P is $\Theta(m+n)$.*

Proof. Suppose that the slack matrix SM can be written as the product of nonnegative matrices F and V of dimensions $f \times m$ and $m \times v$, respectively. Let $(\mathrm{L1}): Ax = b$; $Cx \leqslant d$ be a complete description of the polytope P in the standard variables x. Introduce a vector y of m new variables, and consider the LP $(\mathrm{L2}): Ax = b$; $Cx + Fy = d$; $y \geqslant 0$. Since F is a nonnegative matrix, the x-projection of any feasible solution of L2 satisfies L1, and thus is in P. Conversely, for every vertex x^j of P, if the corresponding column of V is y^j, then the vector $\langle x^j, y^j \rangle$ is a feasible solution to L2. Therefore, L2 expresses P. The linear program L2 has $m+n$ variables; it may have many more equality constraints. However, at most $m+n$ of them are linearly independent, and the rest can be removed. In general, a LP consisting of the nonnegativity constraints $y \geqslant 0$ and a set of equality constraints describing the affine hull of the vectors $\langle x^j, y^j \rangle$ that correspond to the vertices, expresses P.

Consider now a LP that expresses P. At the cost of at most doubling the number of constraints and variables, we may assume without loss of generality that the LP has the form (L3): $Rx + Sy = t$; $y \geqslant 0$. Let $\langle x^j, y^j \rangle$ be a feasible solution for each vertex x^j of P. Since (L3) expresses P, it must imply every facet $c_i x \leqslant d_i$ of P. From linear programming theory, this means that there is a vector μ_i of multipliers for the equalities of (L3), such that $\mu_i[R, S] = \langle c_i, f_i \rangle$ with $f_i \geqslant 0$, and $\mu_i t = d_i$. Thus, for the solution $\langle x^j, y^j \rangle$ corresponding to the jth vertex, we have: $c_i x^j + f_i y^j = \mu_i[R, S][\begin{smallmatrix} x^j \\ y^j \end{smallmatrix}] = \mu_i t = d_i$. That is, the slack of the jth vertex in the ith facet is $f_i y^j$, and $SM = F \cdot V$, where the matrix F has the f_i's as its rows and V has the y^j as its columns. ∎

Expressing the polytope by a LP in the original variables corresponds to the trivial factorization $SM = I \cdot SM$ (I the $f \times f$ identity matrix). The theorem remains true if the matrix is augmented with additional rows corresponding to any valid constraints, and additional columns corresponding to any feasible points. In particular, to get a lower bound we may use any valid constraints and do not need to know a full description of the polytope.

The theorem concerns only the number of variables and constraints of the LP, and not the sizes of its coefficients. An analogous result holds for the total LP size if we take into account also the sizes of the entries of the factor matrices. Consider a polytope P whose vertices have size polynomial in n. If the slack matrix SM of P can be factored into two nonnegative matrices F, V with dimensions $f \times m$, $m \times v$, and whose entries have size at most l, then P can be expressed by a LP of size polynomial in n, m, l. And conversely, if P can be expressed by a LP of size s, then SM can be factored as above so that l and m are polynomial in s. The proof is the same as for Theorem 3 using standard arguments to bound the sizes of the numbers.

EXAMPLE. Consider the spanning tree polytope and its exponential family of facets $\sum_{i,j \in S} x_{ij} \leqslant |S| - 1$ for every subset S of nodes. (The rest of the constraints of the polytope are few in number, and we may include them explicitly in the LP.) Let T be a spanning tree and S a subset of nodes. If we root the tree at an arbitrary node k of S, then the slack $SM(S, T)$ of the vertex T in the facet corresponding to S is simply the number of nodes of S whose parent is not in S. Thus, if we introduce a new variable λ_{kij} for every triple of nodes k, i, j with $k \neq i \neq j$, and its value for a spanning tree T is: $\lambda_{kij} = 1$ if j is the parent of i when we root the tree at k, and 0 otherwise, then

$$SM(S, T) = \sum_{i \in S, j \notin S} \lambda_{kij}. \tag{1}$$

This equation describes a factorization of SM into two nonnegative matrices F and V, where the columns of F and rows of V correspond to the variables λ_{kij}; the column of V corresponding to a spanning tree T consists of the values of the variables λ_{kij} for T, and the row of F corresponding to a subset S of nodes consists of the coefficients (0 or 1) of Eq. (1) for any particular choice of a node k of S.

Thus, we can express the spanning tree polytope by a LP consisting of the constraints we left out (some of these are actually redundant), the nonnegativity constraints, and equality constraints satisfied by all the spanning tree solutions. The resulting LP is basically the same as the one of Section 2, except that the second and third inequality there are replaced by equalities. The constraint $\sum_j \lambda_{kij} = 1$ (with $i \neq k$) says that if we root a spanning tree at k, then every other node i has one parent, and the constraint $x_{ij} = \lambda_{kij} + \lambda_{kji}$ says that an edge $[i, j]$ is in the spanning tree iff either i is the parent of j or j is the parent of i. Of course, the root k has no parent; thus, $\lambda_{kkj} = 0$ and can be omitted.

Observe that if $\langle x, \lambda \rangle$ is a solution to this LP, S is any subset of nodes and $k \in S$, then $\sum_{i,j \in S} x_{ij} = \sum_{i,j \in S} \lambda_{kij} + \lambda_{kji} = \sum_{j \in S, j \neq k} [\sum_{i \in S} \lambda_{kij}] = \sum_{j \in S, j \neq k} [1 - \sum_{i \notin S} \lambda_{kij}] = |S| - 1 - \sum_{j \in S, i \notin S} \lambda_{kij} \leqslant |S| - 1$.

Let us call the smallest number m of the theorem, the *positive rank* of the matrix SM. We do not know of any techniques for estimating or deriving bounds for the positive rank of a matrix. There are two parts in the definition of this parameter: the linear algebra part and the nonnegativity restriction. If we ignore the second restriction we get simply the rank of the matrix SM. Although typically SM has an exponential number of rows and columns, its rank is always small, less than n: if $Cx \leqslant d$ is the inequality system describing the facets of the polytope, then $SM = de - CX$, where X is a matrix with the vertices as columns and e is a row vector of 1's. If we ignore the linear algebra part and just look at the zero-nonzero structure of the matrix, we can view the problem as one of communication complexity.

The setting in a communication problem is as follows. There are two sides, R and C, and each of them gets part (r and c, respectively) of the input. The problem is to design a protocol, which uses the minimum number of bits of communication and allows the two sides to compute a given function or predicate of the distributed input r, c. There is also a notion of a nondeterministic protocol for a predicate: here, the two sides are allowed to make guesses in their communication, and the requirement is that if the input r, c satisfies the predicate, then for at least one guess the two sides must determine that this is the case. (See [AUY, MS, PS, Y] for more information and background.) One can associate a *communication matrix* CM with a communication problem. The rows of the matrix correspond to the possible inputs r to the side R, the columns correspond to the possible inputs c to the side C, and for any r, c, the entry $CM(r, c)$ is the value of the function or predicate that has to be computed for this input. In the case of a predicate, CM is a 0–1 matrix. A *monochromatic rectangle* of CM is a submatrix defined by a subset of (not necessarily consecutive) rows and columns whose entries are equal. It is known that the nondeterministic communication complexity of a predicate is equal to the logarithm of the minimum number, call it s, of monochromatic rectangles needed to cover the 1's of the communication matrix of the predicate. Equivalently, the nondeterministic complexity of a predicate is the logarithm of the smallest number s such that the communication matrix of the predicate can be written as the *Boolean* product ($+$ is OR and $\times$ is AND) of two (0–1) matrices with s as the intermediate dimension (number of columns of the first matrix and rows of the second).

Let FV be the predicate which is 1 (true) of a facet and a vertex if the vertex does not lie on the facet, and is 0 (false) otherwise. Consider the following communication problem: there are two sides, one knows a facet f_i and the other a vertex v_j and want to compute FV. The communication matrix of the predicate FV is just the zero-nonzero structure of the slack matrix SM. From Theorem 3 we have:

COROLLARY 4. *The nondeterministic communication complexity of the predicate FV is a lower bound on the logarithm of the minimum size of an LP expressing the polytope.*

For the matching polytope there is an obvious $4 \log n$ nondeterministic protocol: just guess two edges of the matching that cross the partition of the facet. Thus, in the case of the matching polytope, the corollary cannot give a lower bound better than n^4, which is probably (we believe) far from tight. However, even such a lower bound would be nontrivial and would be enough to imply that the direct application of a LP algorithm to general matching could not compete with present combinatorial algorithms.

For the known classes of facets of the TSP polytope that we mentioned in the previous section (including the clique-tree inequalities) it turns out that there are also nondeterministic protocols of complexity $O(\log n)$, and thus Corollary 4 cannot give a superpolynomial bound for the usual TSP facets. As in the case of matching, it can be shown that for each of these facets f_i and for every tour C that does not lie on f_i, there is a bounded number of edges of C such that every tour that contains these edges does not lie on f_i. In the next section we will see some constraints derived from the independent set problem which do not have this property, and therefore may possibly give rise to a superpolynomial lower bound.

5. Vertex Packing Polytopes

The *vertex packing polytope* $VP(G)$ of a graph G is the convex hull of the characteristic vectors of its independent sets of nodes. Note: there is one polytope for every graph. As expected, we do not know full descriptions of these polytopes. However, for some classes of graphs certain simple and natural constraints suffice.

First there are the obvious constraints: (1) $0 \leqslant x_i \leqslant 1$ for every node i, and $x_i + x_j \leqslant 1$ for every edge (i, j) of the graph. These constraints describe the polytope $VP(G)$ iff G is bipartite. Another set of constraints follows from the fact that a cycle C with an odd number $2k+1$ of nodes can contain at most k nodes of an independent set: (2) $\sum_{i \in C} x_i \leqslant (|C|-1)/2$ for all odd cycles C of the graph. The constraints (1) and (2) describe the vertex packing polytope for a class of graphs called *t-perfect*. Although there is in general an exponential number of constraints of type (2), there is a good separation algorithm for them, and thus the optimization problem over the polytope defined by (1) and (2) can be solved in polynomial time using the ellipsoid algorithm (see [L]). It is not too hard to show also:

THEOREM 4. *The polytope defined by constraints* (1) *and* (2) *can be expressed by a polynomial size LP.*

Proof. The LP follows the separation algorithm for the constraints (2). Given a vector x that satisfies constraints (1), consider the graph as having lengths on the edges, where the length l_{ij} of the edge $[i, j]$ is $1 - x_i - x_j$ (thus, $l_{ij} \geqslant 0$ by (1)). Constraints (2) say that for every odd cycle C, its length $\sum_{[i,j] \in C} l_{ij} = |C| - 2\sum_{i \in C} x_i$ is at least 1. The separation algorithm computes the shortest odd cycle and tests if its length is less than 1.

For every pair of nodes i, j, introduce variables e_{ij} and o_{ij}, which stand for the even and odd distances, respectively, between i and j; we do not need the variables e_{ii} (the even distance from i to itself is 0) but we do have variables o_{ii}. The constraints are: $0 \leqslant x_i \leqslant 1$ for all nodes i; $0 \leqslant o_{ij} \leqslant 1 - x_i - x_j$ for every edge $[i, j]$; $o_{ij} \leqslant o_{ik} + e_{kj}$ and $e_{ij} \leqslant o_{ik} + o_{kj}$ for every edge $[i, k]$ and node j; $o_{ii} \geqslant 1$ for all i.

It is easy to see that in any feasible solution to this LP, the values of e_{ij} and o_{ij} are bounded from above by the length of the shortest even and odd path respectively from i to j (if there are paths with these parities); thus, no odd cycle has length less than 1, because of the constraints $o_{ii} \geqslant 1$. Conversely, given a solution x to constraints (1) and (2), we can extend it to the new variables by letting e_{ij} (resp. o_{ij}) be the length of the shortest, not necessarily simple, path from i to j with an even (resp. odd) number of edges; if there is no even or odd $i-j$ path then give the corresponding variable a large value, for example, n. It is easy to see that this is a feasible solution to the LP. ∎

Another set of valid constraints for $VP(G)$ follows from the fact that an independent set can contain at most one node from a clique: (3) $\sum_{i \in K} x_i \leqslant 1$ for every clique K of the graph. Together with the nonnegativity constraints $x_i \geqslant 0$, these constraints are sufficient to describe the vertex packing polytope of *perfect* graphs. This is a well-studied, rich class of graphs; it includes several natural subclasses (for example, chordal and comparability graphs and their complements). Some basic properties are: the chromatic number is equal to the maximum clique size; every induced subgraph of a perfect graph is also perfect; the complement of a perfect graph is also perfect (see [BC, G] for more information). The maximum (weight) independent set problem can be solved on perfect graphs through a very complex and deep application of an extension of the Ellipsoid algorithm [GLS]. It is an important open problem in computational graph theory to find a better algorithm for this problem.

The slack matrix for the constaints (3) is 0–1, has one row for every clique and one column for every independent set, and the entry corresponding to a clique K and an independent set I is 1 if $K \cap I = \varnothing$, and 0 otherwise. The constraints (3) for nonmaximal cliques are clearly redundant; however, it is convenient to include them in the slack matrix. Note then that the transpose of the slack matrix for a graph G is simply the slack matrix for the complementary graph $\bar{G}$. Since any factorization of a matrix into two nonnegative matrices implies obviously a factorization for its transpose, it follows that the number of variables and constraints needed to express the vertex packing polytope of a perfect graph and its complement are linearly related (and the LP sizes are polynomially related).

For a graph G, let Q be the predicate which is true of a clique K and independent set I if K and I are disjoint and false otherwise. We do not see any obvious protocol for Q, but there is an obvious non-deterministic protocol with complexity $\log n$ for the *complementary* predicate $\bar{Q}$: guess the node in the intersection of K and I. This protocol is *unambiguous*: if $\bar{Q}$ is true, then exactly one guess is successful because a clique and an independent set cannot have more than one node in common. In terms of the communication matrix of a predicate Π, the unambiguous complexity of Π is equal to the logarithm of the smallest number d of *disjoint* monochromatic rectangles that cover the 1's of the matrix $CM(\Pi)$; this number d is also the smallest number such that $CM(\Pi)$ can be written as the product (real multiplication) of two 0–1 matrices with d as the intermediate dimension [MS]. Thus, an unambigous protocol of complexity $\log d$ for the predicate Q of a perfect graph G (and of course, a deterministic protocol, as well) gives a linear program expressing $VP(G)$ with $O(d)$ variables and constraints (and 0–1 coefficients).

EXAMPLE. Suppose that G is a comparability graph, and let D be its underlying partial order. A clique of G corresponds to a path in D. Let K be a clique with nodes $v_1, ..., v_k$ in the order they appear in D, and let I be a disjoint independent set. Either (a) every node of K precedes in the partial order D some node of I (equivalently, v_k precedes some node of I), or (b) no node of K precedes a node of I (equivalently, v_1 does not precede a node of I), or (c) there is a unique i such that v_i precedes some node of I (and thus, so do also its predecessors $v_1, ..., v_{i-1}$), whereas v_{i+1} does not (and neither do the successors $v_{i+2}, ..., v_k$). This observation implies the following unambiguous protocol with complexity $2\log n$ for the predicate Q. The clique side guesses which one of the three cases applies; in the first case it sends v_k, in the second it sends v_1, and in the third it guesses an i and sends v_i and v_{i+1}. Thus, there is an LP with $O(n^2)$ variables and constraints.

Comparability graphs (and their complements) have polynomial size LP's. The same is true of chordal graphs (and their complements), since a chordal graph has at most n maximal cliques. How about the rest of the perfect graphs? The best we can do is $n^{O(\log n)}$ using a result from communication complexity.

LEMMA 1. *If the unambiguous communication complexity of a predicate is g, then its deterministic complexity is at most $O(g^2)$.*

Proof. The proof is very similar to the one in [AUY], that if both a predicate and its complement have nondeterministic complexity g, then the deterministic complexity is at most g^2. Let Π be a predicate with unambiguous complexity g, CM be its communication matrix, and let D be a set of 2^g disjoint monochromatic rectangles that cover the 1's of CM. Let G be a graph whose nodes are the rectangles of D and which has an edge connecting two rectangles if they share a row of CM. For every row r, the rectangles that contain r form a clique K_r, and for every column c, the rectangles that contain c form an independent set I_c, because the rectangles are disjoint, so no two of them can share both a row and a column. Since the rectangles cover the 1's of the matrix, for any row r and column c the corresponding entry $CM(r, c)$ is 1 iff $K_r \cap I_c \neq \emptyset$. Some cliques of G may correspond to no row, and some independent sets may correspond to no column. However, this construction shows that any predicate with unambiguous complexity g can be reduced to the Q predicate on a graph with 2^g nodes.

The protocol proceeds in stages which reduce the graph until the answer is determined. In each stage, the clique side sends a node, say u, of the clique K that is adjacent to at most half of the nodes of the current graph or notifies the other side that it has no such node. In the first case, the side with the independent set I communicates whether (i) $u \in I$ or (ii) u is not adjacent to any node of I. If (i) occurs then $K \cap I \neq \emptyset$; if (ii) then $K \cap I = \emptyset$ and the protocol finishes. If neither occurs then the nodes that are not adjacent to u are removed from the graph (they cannot be in the clique, and therefore, neither in $K \cap I$), and the stage finishes. In the second case (all nodes of K are adjacent to more than half of the nodes), the independent side sends a node v of I that is adjacent to at least half of the nodes of the current graph or communicates that it has no such node. In the latter case, $K \cap I = \emptyset$ because of the degrees, and the protocol finishes. Otherwise, the clique side communicates in an analogous fashion whether (i) $v \in K$ or (ii) v is adjacent to all nodes of K. If (i) or (ii) occurs, then the protocol finishes; otherwise the nodes adjacent to v are removed from the graph (they cannot be in I) and the stage finishes. Since every stage removes half of the nodes, there are at most g stages, and the communication per stage is obviously $O(g)$. ∎

As the numbers involved are small (0–1), it follows from the lemma and Theorem 3 that:

THEOREM 5. *The Vertex packing polytopes of perfect graphs can be expressed by LP's of size* $n^{O(\log n)}$.

Note that, unlike Theorem 4, this does not mean that the polytope defined by the constraints (3) is expressible by such an LP for general (nonperfect) graphs. The reason is that the polytope defined by (3) also has fractional vertices for nonperfect graphs. In fact, optimizing over (3) is in general NP-hard [GLS], and it is unlikely that (3) can be expressed by an LP of subexponential size.

Deriving lower bounds on the size of LP's expressing vertex packing polytopes in general would imply similar bounds for the TSP:

THEOREM 6. *If the TSP polytope can be expressed by a polynomial size LP, then so can the vertex packing polytopes of all graphs.*

Proof Sketch. Let G be a graph with n nodes. We can construct another graph H with $O(n^2)$ nodes such that $VP(G)$ is a projection of the polytope TSP(H), the convex hull of the Hamilton circuits of H. Thus, a LP for $VP(G)$ can be derived from a LP for the TSP polytope on $O(n^2)$ nodes by setting to 0 the variables x_{ij} for the edges $[i, j]$ missing from H.

The construction of H is similar to the reduction from the vertex cover to the Hamilton circuit probem (see, e.g., [AHU]). First we construct a directed graph D as follows. For each node i of G, we have a node u_i and a path p_i in D. For every edge $[i, j]$ of G, the path p_i has two consecutive nodes $\langle i, j, 0\rangle$ and $\langle i, j, 1\rangle$. There are arcs in both directions connecting the odes $\langle i, j, 0\rangle$ and $\langle j, i, 0\rangle$ (of the paths p_i, p_j, respectively) and, also, arcs connecting the nodes $\langle i, j, 1\rangle$ and $\langle j, i, 1\rangle$. For each i, the graph D has arcs from u_i to the first node of the path p_i and to node u_{i+1} (addition mod n) and from the last node of p_i to u_{i+1}. Consider a Hamilton cycle C of the graph D. It is easy to see that the set of nodes i of G such that C contains the arc from u_i to u_{i+1} is an independent set. And conversely, for every independent set I of G, there is a Hamilton cycle C of D such that $I = \{i \mid (u_i, u_{i+1}) \in C\}$.

From D construct an undeirected graph H in the usual way. Replace every node v of D by a path of three nodes $\langle v, 1\rangle$, $\langle v, 2\rangle$, $\langle v, 3\rangle$, and every arc $v \to w$ of D by an edge joining $\langle v, 3\rangle$ and $\langle w, 1\rangle$. For each node i of G, identify the corresponding variable (coordinate) x_i of $VP(G)$ with the variable of the TSP corresponding to the edge joining $\langle u_i, 3\rangle$ and $\langle u_{i+1}, 1\rangle$. Then the polytope $VP(G)$ is a projection of $TSP(H)$. ∎

As we mentioned earlier, it is not clear whether there is a nondeterministic protocol with logarithmic complexity for the predicate Q of a graph. We can formulate the nondeterministic complexity of Q as the following graph theoretic problem. Let us say that a family of t partitions of the nodes N into to sets, say $(S_1, N - S_1), ..., (S_t, N - S_t)$, is a *splitting* family if every clique K and disjoint independent set I are split by some partition in the family (i.e., for some i, $K \subseteq S_i$ and $I \subseteq N - S_i$). Let $q(G)$ be the smallest size t of a splitting family for G.

LEMMA 2. *For every graph G, the nondeterministic complexity of its predicate Q is* $\log q(G)$.

Proof. A splitting family with $q(G)$ partitions implies an obvious nondeterministic protocol: for every clique and independent set guess the index of a partition in the family that splits them. Conversely, consider a nondeterministic protocol for Q with complexity $\log t$, i.e., with at most t possible message exchanges. For the ith message exchange ($i = 1, ..., t$), let S_i be the union of the cliques K for which the clique side concludes that Q is 1 (i.e., that K is disjoint from the independent set), and let T_i be the union of the independent sets for which the side with the independent set concludes that Q is 1. We claim that $S_i \cap T_i = \emptyset$. For, if the intersection contains some node v, then let K and I be a clique and an independent set that caused v to be included in S_i and T_i, respectively; $K \cap I \neq \emptyset$, contradicting the correctness of the protocol. Thus, $T_i \subseteq N - S_i$. For every clique K and disjoint independent set I there is a message exchange for which both sides conclude that Q is 1. Thus, there is an i such that $K \subseteq S_i$ and $I \subseteq T_i \subseteq N - S_i$. ∎

The parameter $q(G)$ is not affected significantly if we just require the family of partitions to split *maximal* cliques from (disjoint) maximal independent sets; at most it decreases by $2n$. Let $q(n) = \max\{q(G) : G \text{ a graph with } n \text{ nodes}\}$. Is $q(n)$ superpolynomial? We have not been able to resolve this question. This is equivalent to the question whether the nondeterministic complexity of a predicate is linear in the unambiguous complexity of its complement. (This follows from the fact that any predicate with unambiguous complexity g can be reduced to the Q predicate on a graph with 2^g nodes—see the proof of Lemma 1). From Corollary 4, Theorem 6, and Lemma 2, we have:

COROLLARY 5. *If $q(n)$ is superpolynomial, then the TSP polytope cannot be expressed by a polynomial size LP.*

6. Open Problems

This work is a step towards a sysematic study of expressing combinatorial optimization problems with small linear programs; there are clearly many open problems. Some of the more immediate ones are:

(1) Find techniques for computing or bounding the positive rank of a matrix.

(2) We do not think that asymmetry helps much. Thus, prove that the matching and TSP polytopes cannot be expressed by polynomial size LP's without the symmetry assumption.

(3) Can we express the vertex packing polytpes of perfect graphs with polynomial size LP's How small?

References

[AHU] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "The Design and Analysis of Computer Algorithms," Addison-Wesley, Reading, MA, 1974.

[AUY] A. V. Aho, J. D. Ullman, and M. Yannakakis, On notions of information transfer in VLSI circuits, *in* "Proceedings, 15th Annual ACM Symp. on Theory of Computing," 1983, pp. 133–139.

[BC] C. Berge and V. Chvatal, "Topics on Perfect Graphs," Annals of Discrete Math. Vol. 21, North-Holland, Amsterdam, 1984.

[BP] E. Balas and W. Pulleyblank, The perfectly matchable subgraph polytope of a bipartite graph, *Networks* **13** (1983), 486–516.

[DLR] D. Dobkin, R. J. Lipton, and S. Reiss, Linear programming is log-space hard for P, *Inform. Process. Lett.* **8** (1979), 96–97.

[E] J. Edmonds, Maximum matching and a polyhedron with 0, 1 vertices, *J. Res. Nat. Bur. Standards B* **69** (1965), 125–130.

[GJS] M. R. Garey, D. S. Johson, and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoret. Comput. Sci.* **1** (1976), 237–267.

[G] M. Golumbic, "Algorithmic Graph Theory and Perfect Graphs," Academic Press, New York/London, 1980.

[GLS] M. Grotschel, L. Lovasz, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* **1** (1981), 169–197.

[GP] M. Grotchel and M. W. Padberg, Polyhedral theory, and polyhedral computations, *in* "The Traveling Salesman Problem" (E. L. Lawler *et al.*, Eds.), pp. 251–360, Wiley, New York, 1985.

[Je] R. G. Jeroslow, On defining sets of vertices of the hypercube by linear inequalities, *Discrete Math.* **11** (1975), 119–124.

[J] D. S. Johnson, personal communication.

[Ka] N. Karmakar, A new polynomial time algorithm for linear programming, *Combinatorica* **4** (1984), 373–395.

[Kh] L. G. Khachian, A polynomial algorithm in linear programming, *Dokl. Akad. Nauk SSSR* **244** (1979) 1093–1096; English translation, *Soviet Math. Dokl.* **20** (1979), 191–194.

[L] L. Lovasz, Vertex packing algorithms, *in* "Proceedings, Int. Colloq. Automata, Languages and Programming," pp. 1–14, Springer-Verlag, New York/Berlin, 1985.

[M] R. K. Martin, "Using Separation Algorithms to Generate Mixed Integer Model Reformulations," working paper, Graduate School of Business, University of Chicago, 1987.

[MRC] R. K. Martin, R. L. Rardin, and B. A. Campbell, "Polyhedral Characterization of Discrete Dynamic Programming," Technical Report CC-87-24, Purdue University, 1987.

[MS] K. Melhorn and E. M. Schmidt, Las Vegas is better than determinism in VLSI and distributed computing, *in* "Proceedings, 14th Annual ACM Symp. on Theory of Computing, 1982," pp. 330–337.

[PRa] M. Padberg and M. R. Rao, Odd minimum cut-sets and b-matchings, *Math. Oper. Res.* **7** (1982), 67–80.

[PRi] M. Padberg and G. Rinaldi, Optimization of a 532-city symmetric traveling salesman problem by Branch and cut, *Oper. Res. Lett.* **6** (1987), 1–7.

[PS] C. H. Papadimitriou and M. Sipser, Communication complexity, *in* "Proceedings, 14th ACM Symp. on Theory of Computing, 1982," pp. 196–200.

[PW] C. H. Papadimitriou and D. Wolfe, The complexity of facets resolved, *in* "Proceedings, 26th Annual IEEE Symp. on Foundations of Computer Science, 1985," pp. 74–78.

[PY] C. H. Papadimitriou and M. Yannakakis, The complexity of facets (and some facets of complexity), *J. Comput. System Sci.* **28** (1984), 144–259.

[R] A. A. Razborov, A lower bound on the monotone network complexity of the logical permanent, *Math. Notes* **37** (1985), 485–493.

[S] A. Schrijver, "Theory of Linear and Integer Programming," New York, 1986.

[Sw] E. R. Swart, "P = NP," Technical Report, University of Guelph, 1986; revision 1987.

[V] L. G. Valiant, Reducibility by algebraic projections, *Enseign. Math.* **28** (1982), 253–268.

[W] H. Wielandt, "Finite Permutation Groups," Academic Press, New York/London, 1964.

[Y] A. C. Yao, Some complexity questions related to distributive computing, *in* "Proceedings, 11th Annual ACM Symp. on Theory of Computing, 1979," pp. 209–213.