CUTTING PLANE ALGORITHMS FOR INTEGER PROGRAMMING, *Cutting plane algorithms*

Cutting plane methods are exact algorithms for *integer programming problems*. They have proven to be very useful computationally in the last few years, especially when combined with a **branch and bound algorithm** in a **branch and cut** framework. These methods work by solving a sequence of linear programming relaxations of the integer programming problem. The relaxations are gradually improved to give better approximations to the integer programming problem, at least in the neighborhood of the optimal solution. For hard instances that cannot be solved to optimality, cutting plane algorithms can produce approximations to the optimal solution in moderate computation times, with guarantees on the distance to optimality.

Cutting plane algorithms have been used to solve many different integer programming problems, including the *traveling salesman problem* [16, 34, 1], the *linear ordering* problem [17, 30, 31], *maximum cut* problems [4, 10, 27], and *packing* problems [19, 32]. `M. Jünger` *et al.* [23] contains a survey of applications of cutting plane methods, as well as a guide to the successful implementation of a cutting plane algorithm. The book [33] by `G.L. Nemhauser` and `L. Wolsey` provides an excellent and detailed description of cutting plane algorithms and the other material in this entry, as well as other aspects of integer programming. The book [35] by `A. Schrijver` and also the more recent article [36] are excellent sources of additional material.

Cutting plane algorithms for general integer programming problems were first proposed by `R.E. Gomory` in [13, 14]. Unfortunately, the cutting planes proposed by Gomory did not appear to be very strong, leading to slow convergence of these algorithms, so the algorithms were neglected for many years. The development of polyhedral theory and the consequent introduction of strong, problem specific cutting planes led to a resurgence of cutting plane methods in the eighties, and cutting plane methods are now the method of choice for a variety of problems, including the traveling salesman problem. Recently, there has also been some research showing that the original cutting planes proposed by Gomory can actually be useful. There has also been research on other types of cutting planes for general integer programming problems. Current research is focused on developing cutting plane algorithms for a variety of hard combinatorial optimization problems, and on solving large instances of integer programming problems using these methods. All of these issues are discussed below.

A simple example.

Consider, for example, the integer programming problem

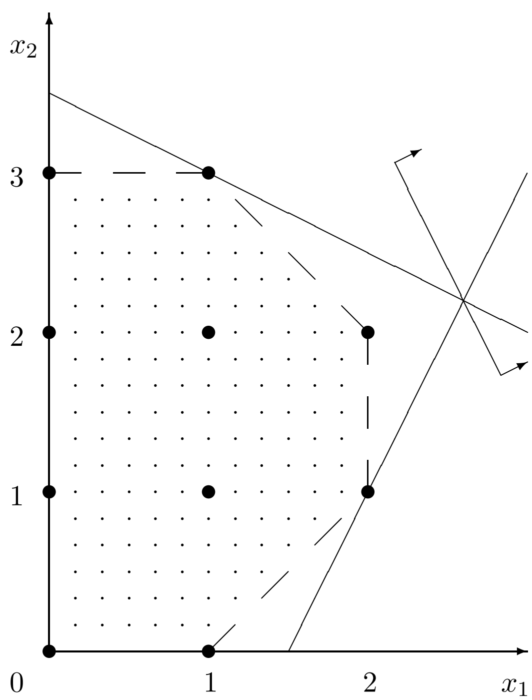
$$\begin{array}{lrcrcl} \min & -2x_1 & - & x_2 & & \\ \text{s.t.} & x_1 & + & 2x_2 & \leq & 7 \\ & 2x_1 & - & x_2 & \leq & 3 \\ & & & x_1, x_2 & \geq & 0, \text{ integer.} \end{array}$$

This problem is illustrated in the figure. The feasible integer points are indicated. The *linear programming relaxation* (or *LP relaxation*) is obtained by ignoring the integrality restrictions; this is given by the polyhedron contained

integer programming problems
branch and bound algorithm
branch and cut
traveling salesman problem
linear ordering
maximum cut
packing
`M. Jünger`
`G.L. Nemhauser`
`L. Wolsey`
`A. Schrijver`
`R.E. Gomory`
linear programming relaxation
LP relaxation

in the solid lines. The boundary of the *convex hull* of the feasible integer points is indicated by dashed lines.

If a cutting plane algorithm were used to solve this problem, the linear programming relaxation would first be solved, giving the point $x_1 = 2.6$, $x_2 = 2.2$, which has value -7.4. The inequalities $x_1 + x_2 \leq 4$ and $x_1 \leq 2$ are satisfied by all the feasible integer points but they are violated by the point $(2.6, 2.2)$. Thus, these two inequalities are valid *cutting planes*. These two constraints can then be added to the relaxation, and when the relaxation is solved again, the point $x_1 = 2$, $x_2 = 2$ results, with value -6. Notice that this point is feasible in the original integer program, so it must actually be optimal for that problem, since it is optimal for a relaxation of the integer program.



A cutting plane example

If, instead of adding both inequalities, just the inequality $x_1 \leq 2$ had been added, the optimal solution to the new relaxation would have been $x_1 = 2$, $x_2 = 2.5$, with value -6.5. The relaxation could then have been modified by adding a cutting plane that *separates* this point from the convex hull, for example $x_1 + x_2 \leq 4$. Solving this new relaxation will again result in the optimal solution to the integer program. This illustrates the basic structure of a cutting plane algorithm:

- Solve the linear programming relaxation.
- If the solution to the relaxation is feasible in the integer programming problem, STOP with optimality.
- Otherwise, find one or more cutting planes that separate the optimal solution to the relaxation from the convex hull of feasible integral points, and add a subset of these constraints to the relaxation.
- Return to the first step.

Typically, the first relaxation is solved using the primal simplex algorithm. After the addition of cutting planes, the current primal iterate is no longer feasible. However, the dual problem is only modified by the addition of some variables. If these extra dual variables are given the value 0 then the current dual solution is still dual feasible. Therefore, subsequent relaxations are solved using the dual simplex method.

Notice that the values of the relaxations provide lower bounds on the optimal value of the integer program. These lower bounds can be used to measure progress towards optimality, and to give performance guarantees on integral solutions.

Totally unimodular matrices.

Consider the integer program $\min\{c^T x : Ax = b, 0 \leq x \leq u, x \text{ integer}\}$, where A is an $m \times n$ matrix, c, x, and u are n-vectors, and b is an m-vector. A cutting plane method attempts to refine a linear programming relaxation until it gives a good approximation of the convex hull of feasible integer points, at least in the region of the optimal solution. In some settings, the solution to the initial linear programming relaxation $\min\{c^T x : Ax = b, 0 \leq x \leq u\}$ may give the optimal solution to the integer program. This is guaranteed to happen if the constraint matrix

convex hull
cutting planes
separates $\rightarrow$ separating hyperplane
totally unimodular

A is *totally unimodular*, that is, the determinant of every square submatrix of A is either 0 or ± 1. Examples of totally unimodular matrices include the node-arc incidence matrix of a directed graph, the node-edge incidence matrix of a bipartite undirected graph, and interval matrices (where each row of A consists of a possibly empty set of zeroes followed by a set of ones followed by another possibly empty set of zeroes). It therefore suffices to solve the linear programming relaxation of *maximum flow problems* and *shortest path* problems on directed graphs, the *assignment problem*, and some problems that involve assigning workers to shifts, among others.

Chvátal-Gomory cutting planes.

One method of generating cutting planes involves combining together inequalities from the current description of the linear programming relaxation. This process is known as *integer rounding*, and the cutting planes generated are known as *Chvátal-Gomory cutting planes*. Integer rounding was implicitly described by `R.E. Gomory` in [13, 14], and described explicitly by `V. Chvátal` in [7].

Consider again the example problem given earlier. The first step is to take a weighted combination of the inequalities. For example,

$$0.2(x_1 + 2x_2 \leq 7) + 0.4(2x_1 - x_2 \leq 3)$$

gives the valid inequality for the relaxation:

$$x_1 \leq 2.6.$$

In any feasible solution to the integer programming problem, the left hand side of this inequality must take an integer value. Therefore, the right hand side can be rounded down to give the following valid inequality for the integer programming problem:

$$x_1 \leq 2.$$

This process can be modified to generate additional inequalities. For example, taking the combination $0.5(x_1 + 2x_2 \leq 7) + 0(2x_1 - x_2 \leq 3)$ gives $0.5x_1 + x_2 \leq 3.5$, which is valid for the relaxation. Since all the variables are constrained to be nonnegative, rounding down the left hand side of this inequality will only weaken it, giving $x_2 \leq 3.5$, also valid for the LP relaxation. Now rounding down the right hand side gives $x_2 \leq 3$, which is valid for the integer programming problem, even though it is not valid for the LP relaxation.

Gomory originally derived constraints using the optimal simplex tableau. The LP relaxation of the simple example above can be expressed in equality form as:

$$\begin{array}{lrcrcrcrcl}
\min & -2x_1 & - & x_2 & & & & & & \\
\text{s.t.} & x_1 & + & 2x_2 & + & x_3 & & & = & 7 \\
& 2x_1 & - & x_2 & & & + & x_4 & = & 3 \\
& & & \multicolumn{5}{c}{x_i \geq 0,\ i = 1, \ldots, 4.} & &
\end{array}$$

Notice that if x_1 and x_2 take integral values then so must x_3 and x_4. Solving this LP using the simplex algorithm gives the optimal tableau

7.4	0	0	0.8	0.6
2.2	0	1	0.4	-0.2
2.6	1	0	0.2	0.6

The rows of the tableau are linear combinations of the original objective function and constraints, and cutting planes can be generated using them. The objective function row implies that $0.8x_3 + 0.6x_4 \geq 0.4$ in any integral feasible solution. It can be seen that this is equivalent to requiring that $2x_1 + x_2 \leq 7$, by substituting for x_3 and x_4 from the equality form given above. It is also possible to generate constraints from the other rows of the tableau. For example, the first constraint row of the tableau is equivalent to the equality $2.2 = x_2 + 0.4x_3 - 0.2x_4$. The fractional part of the right hand side of this equation is $0.4x_3 + 0.8x_4$. This must be at least as large as the fractional part of the left hand side in any feasible integral solution, giving the valid cutting plane $0.4x_3 + 0.8x_4 \geq 0.2$, which is equivalent to $x_1 \leq 2.5$. Similarly, the final

maximum flow problems
shortest path
assignment problem
integer rounding
Chvátal-Gomory cutting planes
`R.E. Gomory`
`V. Chvátal`

row of the tableau can be used to generate the constraint $0.2x_3 + 0.6x_4 \geq 0.6$, or equivalently $7x_1 - x_2 \leq 13$. In practice, the cut added to the tableau should be expressed in the nonbasic variables, here x_3 and x_4, since the tableau will then be in standard form for the dual simplex algorithm.

Gomory's cutting plane algorithm solves an integer program by solving the LP relaxation to optimality, generating a cutting plane from a row of the tableau if necessary, adding this additional constraint to the relaxation, solving the new relaxation, and repeating until the solution to the relaxation is integral. It was shown in [14] that if a cutting plane is always generated from the first possible row then Gomory's cutting plane algorithm will solve an integer program in a finite number of iterations.

Unfortunately, this finite convergence appears to be slow. However, it was shown in [3, 6] that Gomory's cutting plane algorithm can be made competitive with other methods if certain techniques are used, such as adding many Chvátal-Gomory cuts at once.

It follows from the finite convergence of Gomory's cutting plane algorithm that every valid inequality for the convex hull of feasible integral points is either generated by repeated application of integer rounding or is dominated by an inequality generated in such a way. There are many different ways to generate a given inequality using integer rounding. The *Chvátal rank* of a valid inequality is the minimum number of successive applications of the integer rounding procedure that are needed in order to generate the inequality; it should be noted that a rank 2 inequality can be generated by applying the integer rounding procedure to a large number of rank 1 and rank 0 inequalities, for example.

It was shown in [28] that Gomory cutting planes can be generated even when an interior point method is used to solve the LP relaxations, because much of the information in the simplex tableau can still be obtained easily.

Strong cutting planes from polyhedral theory.

The resurgence of interest in cutting plane algorithms in the 1980's was due to the development of *polyhedral combinatorics* and the consequent implementation of cutting plane algorithms that used *facets* of the convex hull of integral feasible points as cuts. A facet is a face of a polytope that has dimension one less than the dimension of the polytope. Equivalently, to have a complete linear inequality description of the polytope, it is necessary to have an inequality that represents each facet.

In the example above, the convex hull of the set of feasible integer points has dimension 2, and all of the dashed lines represent facets. The valid inequality $x_1 + 2x_2 \leq 7$ represents a face of the convex hull of dimension 0, namely the point $(1, 3)$.

If a complete description of the convex hull of the set of integer feasible points is known, then the integer problem can be solved as a linear programming problem by minimizing the objective function over this convex hull. Unfortunately, it is not easy to get such a description. In fact, for an **NP-Complete** problem [12], such a description must contain an exponential number of facets, unless P=NP.

The paper [23] contains a survey of problems that have been solved using *strong cutting plane* algorithms. Typically in these algorithms, first a partial polyhedral description of the convex hull of the set of integer feasible points is determined. This description will usually contain families of facets of certain types. *Separation routines* for these families can often be developed; such a routine will take as input a point (for example, the optimal solution to the LP relaxation), and return as output violated constraints from the family, if any exist.

Gomory's cutting plane algorithm
Chvátal rank
polyhedral combinatorics
facets
NP-Complete
strong cutting plane
Separation routines

The prototypical combinatorial optimization problem that has been successfully attacked using cutting plane methods is the *traveling salesman problem*. In this problem, a set of cities is provided along with distances between the cities. A route that visits each city exactly once and returns to the original city is called a *tour*. It is desired to choose the shortest tour. This problem has many applications, including printed circuit board (PCB) production: a PCB needs holes drilled in certain places to hold electronic components such as resistors, diodes, and integrated circuits. These holes can be regarded as the cities, and the objective is to minimize the total distance traveled by the drill.

The traveling salesman problem can be represented on a graph, $G = (V, E)$, where V is the set of vertices (or cities) and E is the set of edges (or links between the cities). Each edge $e \in E$ has an associated cost (or length) c_e. If the incidence vector x is defined by

$$x_e = \begin{cases} 1 & \text{if edge } e \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

then the traveling salesman problem can be formulated as $\min\{\sum c_e x_e : x \text{ is the incidence vector of a tour}\}$. Notice that for a tour, at each vertex the sum of the edge variables must be two; this is called a degree constraint. This leads to the relaxation of the traveling salesman problem:

$$\begin{array}{lll} \min & \sum c_e x_e & \\ \text{s.t.} & \sum_{e \in \delta(v)} x_e = 2 & \text{for all vertices } v \\ & x_e = 0 \text{ or } 1 & \text{for all edges } e. \end{array}$$

Here, $\delta(v)$ denotes the set of all edges incident to vertex v. All tours are feasible in this formulation, but it also allows infeasible solutions corresponding to *subtours*, consisting of several distinct unconnected loops. To force the solution to be a tour, it is necessary to include *subtour elimination constraints* of the form

$$\sum_{e \in \delta(U)} x_e \geq 2$$

for every subset $U \subseteq V$ with cardinality $2 \leq |U| \leq |V|/2$, where $\delta(U)$ denotes the set of edges with exactly one endpoint in U. Any feasible solution to the relaxation given above which also satisfies the subtour elimination constraints must be the incidence vector of a tour. Unfortunately, the number of subtour elimination constraints is exponential in the number of cities. This led `G.B. Dantzig` *et al.* to propose a cutting plane algorithm in [9], where the subtour elimination constraints are added as cutting planes as necessary.

The degree constraints and the subtour elimination constraints, together with the simple bounds $0 \leq x_e \leq 1$, are still not sufficient to describe the convex hull of the incidence vectors of tours. This approach of [9] has been extended in recent years by the incorporation of additional families of cutting planes — see, for example, [1, 16, 34].

Thus, cutting plane algorithms can be used even when the integer programming formulation of the problem has an exponential number of constraints. Similar ideas are used in papers on the matching problem [11, 15], maximum cut problems [4, 10, 27], and the linear ordering problem [17, 31], among others. The pioneering work of `J. Edmonds` on the matching problem gave a complete description of the matching polytope, and this work was used in subsequent algorithms; it was also an inspiration to future work on many other problems and even to the formulation of complexity theory and the concept of a "good" algorithm.

Alternative general cutting planes.

A knapsack problem is an integer programming problem with just one linear inequality constraint. A general integer programming problem can be regarded as the intersection of several knapsack problems, one for each constraint. This observation was used in [8, 20, 21] to solve general integer programming problems. The approach consists of finding facets and strong cutting planes for the knapsack problem and adding

---

traveling salesman problem
subtour elimination constraints
`G.B. Dantzig`
`J. Edmonds`

these constraints to the LP relaxation of the integer program as cutting planes.

There has been interest recently in other families of cutting planes for general integer programming problems. Two such families of cuts are *lift-and-project cuts* [2] and *Fenchel cuts* [5]. To find a cut of this type, it is generally necessary to solve a linear programming problem.

These alternative general cutting planes are not usually strong enough on their own to solve an integer programming problem, and they are most successfully employed in **branch and cut algorithms for integer programming**; they are discussed in more detail in that entry.

Fixing variables.

If the *reduced cost* of a nonbasic variable is sufficiently large at the optimal solution to an LP relaxation, then that variable must take its current value in any optimal solution to the integer programming problem. To make this more precise, suppose the binary variable x_j takes value zero in the optimal solution to an LP relaxation and that the reduced cost of this variable is r_j. The optimal value of the relaxation gives a lower bound $\underline{z}$ on the optimal value of the integer programming problem. The value z_{UB} of the best known feasible integral solution provides an upper bound on the optimal value. Any feasible point in the relaxation with $x_j = 1$ must have value at least $\underline{z} + r_j$, so such a point cannot be optimal if $r_j > z_{UB} - \underline{z}$. Similar tests can be derived for nonbasic variables at their upper bounds. It is also possible to fix variables when an interior point method is used to solve the relaxations [28].

Once some variables have been fixed in this manner, it is often possible to fix further variables using logical implications. For example, in a traveling salesman problem, if x_e has been set equal to one for two edges incident to a particular vertex, then all other edges incident to that vertex can have their values fixed to zero.

Solving large problems.

It is generally accepted that interior point methods are superior to the simplex algorithm for solving sufficiently large linear programming problems. The situation for cutting plane algorithms for large integer programming problems is not so clear, because the dual simplex method is very good at reoptimizing if only a handful of cutting planes are added. Nonetheless, it does appear that interior point cutting plane algorithms may well have a role to play, especially for problems with very large relaxations (thousands of variables and constraints) and where a large number of cutting planes are added simultaneously (hundreds or thousands). LP relaxations of integer programming problems can experience severe degeneracy, which can cause the simplex method to stall. Interior point methods suffer far less from the effects of degeneracy.

In [29], an interior point cutting plane algorithm is used for a maximum cut problem on a sparse graph, and the use of the interior point solver enables the solution of far larger instances than with a simplex solver, because of both the size of the problems and their degeneracy.

A combined interior point and simplex cutting plane algorithm for the linear ordering problem is described in [31]. In the early stages, an interior point method is used, because the linear programs are large and many constraints are added at once. In the later stages, the dual simplex algorithm is used, because just a few constraints are added at a time and the dual simplex method can then reoptimize very quickly. The combined algorithm is up to ten times faster than either a pure interior point cutting plane algorithm or a pure simplex cutting plane algorithm on the larger instances considered.

The polyhedral combinatorics of the quadratic assignment problem are investigated in [22]. It was found necessary to use an interior point method to solve the relaxations, because of the size of the relaxations.

Provably good solutions.

---

lift-and-project cuts
Fenchel cuts
branch and cut algorithms for integer programming
reduced cost → *reduced cost fixing*

Even if a cutting plane algorithm is unable to solve a problem to optimality, it can still be used to generate good feasible solutions with a *guaranteed bound to optimality*. This approach for the traveling salesman problem is described in [24]. The value of the current LP relaxation provides a lower bound on the optimal value of the integer programming problem. The optimal solution to the current LP relaxation (or a good feasible solution) can often be used to generate a good integral feasible solution using a heuristic procedure. The value of an integral solution obtained in this manner provides an upper bound on the optimal value of the integer programming problem.

For example, for the traveling salesman problem, edges that have x_e close to one can be set equal to one, edges with x_e close to zero can be set to zero, and the remaining edges can be set so that the solution is the incidence vector of a tour. Further refinements are possible, such as using 2-change or 3-change procedures to improve the tour, as described in [26].

This has great practical importance. In many situations, it is not necessary to obtain an optimal solution, and a good solution will suffice. If it is only necessary to have a solution within 0.5% of optimality, say, then the cutting plane algorithm can be terminated when the gap between the lower bound and upper bound is smaller than this tolerance. If the objective function value must be integral, then the algorithm can be stopped with an optimal solution once this gap is less than one.

Equivalence of separation and optimization.

The *separation problem* for an integer programming problem can be stated as follows:

> Given an instance of an integer programming problem and a point x, determine whether x is in the convex hull of feasible integral points. Further, if it is not in the convex hull, find a separating hyperplane that cuts off x from the convex hull.

An algorithm for solving a separation problem is called a separation routine, and it can be used to solve an integer programming problem.

The **ellipsoid algorithm** [18, 25] is a method for solving linear programming problems in polynomial time. It can be used to solve an integer programming problem with a cutting plane method, and it will work in a polynomial number of stages, or calls to the separation routine. If the separation routine requires only polynomial time then the ellipsoid algorithm can be used to solve the problem in polynomial time. It can also be shown that if an optimization problem can be solved in polynomial time then the corresponding separation problem can also be solved in polynomial time.

There are instances of any NP-hard problem that cannot be solved in polynomial time unless P=NP. Therefore, a cutting plane algorithm cannot always generate good cutting planes quickly for NP-hard problems. In practice, fast heuristics are used, and these heuristics may occasionally be unable to find a cutting plane even when one exists.

Conclusions.

Cutting plane methods have been known for almost as long as the simplex algorithm. They have come back into favor since the early 1980's because of the development of strong cutting planes from polyhedral theory. In practice, cutting plane methods have proven very successful for a wide variety of problems, giving provably optimal solutions. Because they solve relaxations of the problem of interest, they make it possible to obtain bounds on the optimal value, even for large instances that cannot currently be solved to optimality.

References

[1] Applegate, D., Bixby, R., Chvátal, V., and Cook, W.: Finding Cuts in the TSP (A preliminary report), Tech. rep., Mathematics, AT&T Bell Laboratories, Murray Hill, NJ, 1994.

[2] Balas, E., Ceria, S., and Cornuéjols, G.: 'Mixed 0–1 programming by lift-and-project in a branch-and-cut framework', *Management Science* **42** (1996), 1229–1246.

---

guaranteed bound to optimality
separation problem
ellipsoid algorithm

[3] Balas, E., Ceria, S., Cornuéjols, G., and Natraj, N.: 'Gomory cuts revisited', *Operations Research Letters* **19** (1996), 1–9.

[4] Barahona, F., Grötschel, M., Jünger, M., and Reinelt, G.: 'An application of combinatorial optimization to statistical physics and circuit layout design', *Operations Research* **36(3)** (1988), 493–513.

[5] Boyd, E. A.: 'Fenchel Cutting Planes for Integer Programs', *Operations Research* **42** (1994), 53–64.

[6] Ceria, S., Cornuéjols, G., and Dawande, M.: 'Combining and strengthening Gomory cuts', in E. Balas and J. Clausen (eds.): *Lecture Notes in Computer Science*, Vol. 920, Springer-Verlag, 1995.

[7] Chvátal, V.: 'Edmonds polytopes and a hierarchy of combinatorial problems', *Discrete Mathematics* **4** (1973), 305–337.

[8] Crowder, H. P., Johnson, E. L., and Padberg, M.: 'Solving large-scale zero-one linear programming problems', *Operations Research* **31** (1983), 803–834.

[9] Dantzig, G. B., Fulkerson, D. R., and Johnson, S. M.: 'Solutions of a large-scale travelling salesman problem', *Operations Research* **2** (1954), 393–410.

[10] De Simone, C., Diehl, M., Jünger, M., Mutzel, P., Reinelt, G., and Rinaldi, G.: 'Exact ground states of Ising spin glasses: New experimental results with a branch and cut algorithm', *Journal of Statistical Physics* **80** (1995), 487–496.

[11] Edmonds, J.: 'Maximum matching and a polyhedron with 0, 1 vertices', *Journal of Research National Bureau of Standards* **69B** (1965), 125–130.

[12] Garey, M. R., and Johnson, D. S.: *Computers and intractibility: a guide to the theory of NP-completeness*, Freeman, 1979.

[13] Gomory, R. E.: 'Outline of an algorithm for integer solutions to linear programs', *Bulletin of the American Mathematical Society* **64** (1958), 275–278.

[14] Gomory, R. E.: 'An algorithm for integer solutions to linear programs', *Recent Advances in Mathematical Programming*, in R. L. Graves and P. Wolfe (eds.). McGraw-Hill, New York, 1963, pp. 269–302.

[15] Grötschel, M., and Holland, O.: 'Solving matching problems with linear programming', *Mathematical Programming* **33** (1985), 243–259.

[16] Grötschel, M., and Holland, O.: 'Solution of large-scale travelling salesman problems', *Mathematical Programming* **51(2)** (1991), 141–202.

[17] Grötschel, M., Jünger, M., and Reinelt, G.: 'A cutting plane algorithm for the linear ordering problem', *Operations Research* **32** (1984), 1195–1220.

[18] Grötschel, M., Lovasz, L., and Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, Germany, 1988.

[19] Grötschel, M., Martin, A., and Weismantel, R.: 'Packing Steiner trees: a cutting plane algorithm and computational results', *Mathematical Programming* **72** (1996), 125–145.

[20] Hoffman, K. L., and Padberg, M.: 'LP-based combinatorial problem solving', *Annals of Operations Research* **4** (1985), 145–194.

[21] Hoffman, K. L., and Padberg, M.: 'Improving LP-Representation of Zero-One Linear Programs for Branch-and-Cut', *ORSA Journal on Computing* **3(2)** (1991), 121–134.

[22] Jünger, M., and Kaibel, V.: A basic study of the QAP polytope, Tech. Rep. 96.215, Institut für Informatik, Universität zu Köln, Pohligstraße 1, D-50969 Köln, Germany, 1996.

[23] Jünger, M., Reinelt, G., and Thienel, S.: 'Practical problem solving with cutting plane algorithms in combinatorial optimization', *Combinatorial Optimization: DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. AMS, 1995, pp. 111–152.

[24] Jünger, M., Thienel, S., and Reinelt, G.: 'Provably good solutions for the traveling salesman problem', *ZOR – Mathematical Methods of Operations Research* **40** (1994), 183–217.

[25] Khachiyan, L. G.: 'A polynomial algorithm in linear programming', *Doklady Akademiia Nauk SSSR* **224** (1979), 1093–1096, English Translation: *Soviet Mathematics Doklady*, Volume 20, pp. 1093–1096.

[26] Lin, S., and Kernighan, B. W.: 'An effective heuristic for the traveling salesman problem', *Operations Research* **21** (1973), 498–516.

[27] Mitchell, J. E.: Computational experience with an interior point cutting plane algorithm, Tech. rep., Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, February 1997, Revised: April 1997.

[28] Mitchell, J. E.: 'Fixing variables and generating classical cutting planes when using an interior point branch and cut method to solve integer programming problems', *European Journal of Operational Research* **97** (1997), 139–148.

[29] Mitchell, J. E.: 'An interior point cutting plane algorithm for Ising spin glass problems', in P. Kischka and H.-W. Lorenz (eds.): *Operations Research Proceedings, SOR 1997, Jena, Germany*, Springer-Verlag, 1998, pp. 114–119.

[30] Mitchell, J. E., and Borchers, B.: 'Solving real-world linear ordering problems using a primal-dual interior point cutting plane method', *Annals of Operations Research* **62** (1996), 253–276.

[31] Mitchell, J. E., and Borchers, B.: Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm, Tech. rep., Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, September 1997, Accepted for publication in *Proceedings of HPOPT97*, Rotterdam, The Netherlands.

[32] Nemhauser, G. L., and Sigismondi, G.: 'A strong cutting plane/branch-and-bound algorithm for node packing', *Journal of the Operational Research Society* **43** (1992), 443–457.

[33] Nemhauser, G. L., and Wolsey, L. A.: *Integer and Combinatorial Optimization*, John Wiley, New York, 1988.

[34] Padberg, M., and Rinaldi, G.: 'A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems', *SIAM Review* **33(1)** (1991), 60–100.

[35] Schrijver, A.: *Theory of Linear and Integer Programming*, John Wiley, Chichester, 1986.

[36] Schrijver, A.: 'Polyhedral Combinatorics', *Handbook of Combinatorics*, in R. L. Graham, M. Grötschel, and L. Lovász (eds.), Vol. 2. Elsevier Science, 1995, ch. 30, pp. 1649–1704.

John E. Mitchell
Mathematical Sciences
Rensselaer Polytechnic Institute
Troy NY
USA
E-mail address: mitchj@rpi.edu

AMS 1991 Subject Classification: 90C10, 49M35, 90C11, 90C06, 90C08.

Key words and phrases: cutting planes, integer programs, exact algorithms.