An interior point algorithm to solve computationally difficult set covering problems*

Narendra Karmarkar, Mauricio G.C. Resende and K.G. Ramakrishnan

Mathematical Sciences Research Center
AT&T Bell Laboratories
Murray Hill, NJ 07974 - USA

December 1989
(Revised August 1990)

Abstract

We present an interior point approach to the zero-one integer programming feasibility problem based on the minimization of a nonconvex potential function. Given a polytope defined by a set of linear inequalities, this procedure generates a sequence of strict interior points of this polytope, such that each consecutive point reduces the value of the potential function. An integer solution (not necessarily feasible) is generated at each iteration by a rounding scheme. The direction used to determine the new iterate is computed by solving a nonconvex quadratic program on an ellipsoid. We illustrate the approach by considering a class of difficult set covering problems that arise from computing the 1-width of the incidence matrix of Steiner triple systems.

Key words: Integer programming, interior point method, Steiner triple systems, set covering.

1. Introduction

In this paper we consider the following integer programming problem:
INTEGER PROGRAMMING: Let $B \in \Re^{m \times n'}$ and $b \in \Re^{n'}$. Find $w \in \Re^m$ such that:

$$B^T w \leq b \tag{1}$$

$$w_i = \{-1, 1\},\ i = 1, \ldots, m. \tag{2}$$

* To appear in *Mathematical Programming.*

The more common form of integer programming, where variables x_i take on (0,1) values, can be converted to the above form with the change of variables

$$x_i = \frac{1 + w_i}{2}, \ i = 1, \dots, m.$$

We describe an interior point algorithm, based on the approach proposed by Karmarkar [15], to solve INTEGER PROGRAMMING, i.e. an algorithm that generates a sequence of points $\{w^0, w^1, \dots, w^k, \dots\}$ where for all $k = 0, 1, \dots$

$$w^k \in \left\{ w \in \Re^m | B^T w < b; \ \ -e < w < e \right\},$$

where $e^T = (1, \dots, 1)$. In practice, this sequence often converges to a point from which one can roundoff to a ± 1 integer solution to (1-2). The algorithm cannot indicate that a feasible integer solution does not exist. Nor does there exist a guarantee that the algorithm will find a feasible solution, if one exists. However, we present several instances of computationally difficult set covering problems in which it succeeds in providing optimal or best known solutions.

To simplify notation, let I denote an $m \times m$ identity matrix,

$$A = \left[B \vdots I \vdots -I \right] \in \Re^{m \times n}$$

and

$$c = \begin{bmatrix} b \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \Re^n$$

and let

$$\mathcal{I} = \left\{ w \in \Re^m \mid A^T w \leq c \ \text{ and } \ w_i = \{-1, 1\} \right\}.$$

With this notation, INTEGER PROGRAMMING can be restated as: Find $w \in \mathcal{I}$.

Integer programming is covered extensively in many textbooks, e.g. Schrijver [26] and Nemhauser and Wolsey [20]. While the vast majority of algorithms for (0,1) integer programming are based on branch and bound, enumeration or Lagrangian relaxation, several papers have dealt with interior point methods for integer programming. These papers are all based on the work of Hillier [11]. Hillier considers the optimization form of integer programming in inequality form. In a first phase, Hillier's method tentatively produces a feasible integer interior point. There is no guarantee that this phase will terminate successfully. In a second phase a search is conducted by rounding off points on the line segment going from this integer interior point to the optimal solution to the linear programming relaxation produced by the Simplex Method. Ibaraki, Ohashi and Mine [12] extend the approach of Hillier by carrying out the search on a piecewise linear path. In a third phase, Hillier's method attempts to improve on the solution obtained in phase two. Jeroslow and Smith [13] embedded phases one and two of Hillier approximation technique inside a simple branch-and-bound scheme and report significant improvement of the hybrid method over a pure branch-and-bound implementation. Faaland and Hillier [4] devised a method for constructing search paths motivated by a statistical analysis of existing methods.

Let

$$\mathcal{L} = \left\{ w \in \Re^m \mid A^T w \leq c \right\}$$

and consider the linear programming relaxation of (1-2), i.e. find $w \in \mathcal{L}$. One way of selecting ± 1 integer solutions over fractional solutions in linear programming is to introduce the quadratic objective function,

$$\text{maximize } w^T w = \sum_{i=1}^{m} w_i^2$$

and solve the NP-complete [25] nonconvex quadratic programming problem

$$\text{maximize } w^T w \tag{3}$$

$$\text{subject to : } A^T w \leq c \tag{4}$$

Note that $w^T w \leq m$, with the equality only occurring when $w_j = \pm 1, \quad j = 1, \ldots, m$. The following proposition establishes the relationship between (3-4) and (1-2).

Proposition 1.1 *Let $w \in \mathcal{L}$. Then $w \in \mathcal{I} \Longleftrightarrow w^T w = m$, where m is the optimal solution to (3-4).*

Proof: ($\Longrightarrow$) Clearly, if $w \in \mathcal{I}$ then $w \in \mathcal{L}$ and $w_i = \pm 1, \quad i = 1, \ldots, m$. Hence $w^T w = m$. ($\Longleftarrow$) If w is the optimal solution to (3-4) then $w \in \mathcal{L}$. If $w^T w = m$ then $w_i = \pm 1, \ i = 1, \ldots, m$ and therefore $w \in \mathcal{I}$. $\square$

The relationship between zero-one integer programming (with objective function) and concave programming under linear constraints was first pointed out by Raghavachari [23] (See also [3] and [14]). Glover and Klingman [9] develop an algorithm for (0,1) integer programming based on concave programming. They report no experimental results. Surveys on global constrained concave minimization are given in [21] and [22].

The remainder of this paper is summarized next. In Section 2, we define a nonconvex potential function optimization problem that is central to our approach. We discuss how to carry out this nonconvex optimization and relate this to INTEGER PROGRAMMING. In Section 3, we show how to obtain a descent direction for the potential function. In Section 4, we report computational results of applying the interior point method to compute the

1-width of incidence matrices of Steiner triple systems. Concluding remarks are made in Section 5.

2. Nonconvex Optimization

Consider the potential function

$$\varphi(w) = \log(m - w^T w) - \frac{1}{n}\sum_{i=1}^{n} \log d_i(w)$$

where

$$d_i(w) = c_i - a_i^T w, \quad i = 1, \ldots, n,$$

are the slacks. In place of (3-4), consider the equivalent nonconvex optimization problem:

$$\text{minimize } \{\varphi(w) | A^T w \leq c\}. \tag{5}$$

To solve (5), we use an approach similar to the classical Levenberg-Marquardt methods [16] [17]. Let

$$w^0 \in \mathcal{L}_s = \left\{ w \in \Re^m \mid A^T w < c \right\}$$

be a given initial interior point. Our algorithm generates a sequence of interior points of $\mathcal{L}$. Let $w^k \in \mathcal{L}_s$ be the k-th iterate. Around w^k a quadratic approximation of the potential function is set up.

Let $D = \text{diag}(d_1(w), \ldots, d_n(w))$, $e = (1, \ldots, 1)$, $f_0 = m - w^T w$ and C be a constant. The quadratic approximation of $\varphi(w)$ around w^k is given by

$$Q(w) = \frac{1}{2}(w - w^k)^T H(w - w^k) + h^T(w - w^k) + C \tag{6}$$

where the Hessian is

$$H = -\frac{2}{f_0} I - \frac{4}{f_0^2} w^k {w^k}^T + \frac{1}{n} A D^{-2} A^T \tag{7}$$

and the gradient is

$$h = -\frac{1}{f_0} w^k + \frac{1}{n} A D^{-1} e. \tag{8}$$

Minimizing (6) subject to $A^T w \leq c$ is NP-complete. However, if the polytope is substituted by an inscribed ellipsoid, the resulting approximate problem is easy. Ye [27] has independently proposed a polynomial time algorithm for nonconvex quadratic programming on an ellipsoid. The following proposition describes such an inscribed ellipsoid.

Proposition 2.1 *Consider the polytope defined as*

$$\mathcal{L} = \left\{ w \in \Re^m | A^T w \leq c \right\}$$

and let $w^k \in \mathcal{L}_s = \text{int}(\mathcal{L})$ be an interior point of $\mathcal{L}$. Consider the ellipsoid

$$\mathcal{E}(r) = \left\{ w \in \Re^m | (w - w^k)^T A D^{-2} A^T (w - w^k) \leq r^2 \right\}.$$

Then for $r \leq 1$, $\mathcal{E}(r) \subset \mathcal{L}$, i.e. $\mathcal{E}(r)$ is an inscribed ellipsoid in $\mathcal{L}$.

Proof: It is sufficient to prove case when $r = 1$, since $\mathcal{E}(r) \subset \mathcal{E}(1)$, for $0 \leq r < 1$. Assume $y \in \mathcal{E}(1)$. Then

$$(y - w^k)^T A D^{-2} A^T (y - w^k) \leq 1$$

and consequently

$$D^{-1} A^T (y - w^k) \leq 1.$$

Denoting the i-th row of A^T by $a_{i\cdot}^T$, we have

$$\begin{aligned} \frac{1}{c_i - a_{i\cdot}^T w^k} a_{i\cdot}^T (y - w^k) &\leq 1, \quad \forall i = 1, \ldots, n \\ a_{i\cdot}^T (y - w^k) &\leq c_i - a_{i\cdot}^T w^k, \quad \forall i = 1, \ldots, n \\ a_{i\cdot}^T y &\leq c_i, \quad \forall i = 1, \ldots, n. \end{aligned}$$

Therefore $A^T y \leq c$. Consequently $y \in \mathcal{L}$. $\square$

Substituting the polytope by the appropriate ellipsoid and letting $\Delta w \equiv w - w^k$ results in the optimization problem

$$\text{minimize } \frac{1}{2} (\Delta w)^T H \Delta w + h^T \Delta w \tag{9}$$

$$\text{subject to} : (\Delta w)^T A D^{-2} A^T (\Delta w) \leq r^2. \tag{10}$$

The optimal solution Δw^* to (9-10) is a descent direction of $Q(w)$ from w^k. For a given radius $r > 0$, the value of the original potential function $\varphi(w)$ may increase by moving in the direction Δw^*, because of the higher order terms ignored in the approximation. It can be easily verified, however, that if the radius is decreased sufficiently, the value of the potential function will decrease by moving in the new Δw^* direction. We shall say a *local minimum* to (5) has been found if the radius must be reduced below a tolerance ϵ to achieve a reduction in the value of the potential function.

The following theorem, proved in [15], characterizes the optimal solution of (9-10). In place of the ellipsoid

$$\left\{ x \in \Re^m \mid x^T A D^{-2} A^T x \leq r^2 \right\} \tag{11}$$

the theorem considers the sphere

$$\left\{ x \in \Re^m \mid x^T x \leq r^2 \right\} \tag{12}$$

without loss of generality since $AD^{-2}A^T$ is, by assumption, positive definite and (11) can be converted to the form given in (12) by means of a nonsingular linear transformation of the space.

Theorem 2.2 *Consider the optimization problem:*

$$\textit{minimize } \frac{1}{2}x^T H x + h^T x \tag{13}$$

$$\text{subject to} : x^T x \leq r^2 \tag{14}$$

where $H \in \Re^{m\times m}$ *is symmetric and indefinite,* $x, h \in \Re^m$ *and* $0 < r \in \Re$. *Let* $u_1, \ldots, u_m$ *denote a full set of orthonormal eigenvectors spanning* $\Re^m$ *and let* $\lambda_1, \ldots, \lambda_m$ *be the corresponding eigenvalues ordered so that* $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{m-1} \leq \lambda_m$. *Denote* $0 > \lambda_{min} = min\{\lambda_1, \ldots, \lambda_m\}$ *and* u_{min} *the corresponding eigenvector. Furthermore, let* q *be such that* $\lambda_{min} = \lambda_1 = \cdots = \lambda_q < \lambda_{q+1}$. *To describe the solution to (13-14) we shall consider two cases:*
Case 1: Assume $\sum_{i=1}^q (h^T u_i)^2 > 0$. *Let the scalar* $\lambda \in (-\infty, \lambda_{min})$ *and consider the parametric family of vectors*

$$x(\lambda) = -\sum_{i=1}^{m} \frac{(h^T u_i)u_i}{\lambda_i - \lambda}.$$

For any $r > 0$, *denote by* $\lambda(r)$ *the unique solution of the equation* $(x(\lambda))^T x(\lambda) = r^2$ *in* λ. *Then* $x(\lambda(r))$ *is the unique optimal solution of (13-14).*
Case 2: Assume $h^T u_i = 0, \forall i = 1, \ldots, q$. *Let the scalar* $\lambda \in (-\infty, \lambda_{min})$ *and consider the parametric family of vectors*

$$x(\lambda) = -\sum_{i=q+1}^{m} \frac{(h^T u_i)u_i}{\lambda_i - \lambda}. \tag{15}$$

Let

$$r_{max} = \|x(\lambda_{min})\|_2.$$

If $r < r_{max}$ *then for any* $0 < r < r_{max}$, *denote by* $\lambda(r)$ *the unique solution of the equation* $(x(\lambda))^T x(\lambda) = r^2$ *in* λ. *Then* $x(\lambda(r))$ *is the unique optimal solution of (13-14).*
If $r \geq r_{max}$, *then let* $\alpha_1, \alpha_2, \ldots, \alpha_q$ *be any real scalars such that*

$$\sum_{i=1}^{q} \alpha_i^2 = r^2 - r_{max}^2.$$

Then

$$x = \sum_{i}^{q} \alpha_i u_i - \sum_{i=q+1}^{m} \frac{(h^T u_i)u_i}{(\lambda_i - \lambda_{min})}$$

is an optimal solution of (13-14). Since the choice of α_i*'s is arbitrary, this solution is not unique.*

The key result of Theorem 2.2 used in this paper is the existence of a unique optimal solution to (13-14) if $r < r_{max}$. The proof of Theorem 2.2 relies on a lemma that is used to develop the algorithm described in this paper.

Lemma 2.3 *Let the length of $x(\lambda)$ be*

$$l\left(x(\lambda)\right) \equiv \|x(\lambda)\|_2^2 = \left(x(\lambda)\right)^T x(\lambda).$$

Part 1: Assume $\sum_{i=1}^{q}(h^T u_i)^2 > 0$. Consider the parametric family of vectors

$$x(\lambda) = -\sum_{i=1}^{m} \frac{(h^T u_i)u_i}{\lambda_i - \lambda},$$

for $\lambda \in (-\infty, \lambda_{min})$. Then $l\left(x(\lambda)\right)$ is monotonically increasing in λ in the interval $\lambda \in (-\infty, \lambda_{min})$.

Part 2: Assume $h^T u_i = 0, \forall i = 1, \ldots, q$ and consider the parametric family of vectors

$$x(\lambda) = -\sum_{i=q+1}^{m} \frac{(h^T u_i)u_i}{\lambda_i - \lambda}, \tag{16}$$

for $\lambda \in (-\infty, \lambda_{min})$. Furthermore, assume

$$r < \|x(\lambda_{min})\|_2.$$

Then $l\left(x(\lambda)\right)$ is monotonically increasing in λ in the interval $\lambda \in (-\infty, \lambda_{min})$.

Proof: (Part 1) Since

$$x(\lambda) = -\sum_{i=1}^{m} \frac{(h^T u_i)u_i}{\lambda_i - \lambda},$$

then

$$l\left(x(\lambda)\right) = -\sum_{i=1}^{m} \frac{(h^T u_i)^2}{(\lambda_i - \lambda)^2}.$$

In the range $\lambda \in (-\infty, \lambda_{min})$ each of the terms $1/(\lambda_i - \lambda)^2$, $i = 1, \ldots, m$, is a strictly monotonically increasing function of λ, each of the numerators $(h^T u_i)^2$ $(i = 1, \ldots, m)$ is nonnegative and at least one of the numerators is strictly positive. Hence $l\left(x(\lambda)\right)$ is monotonically increasing in λ. □

(Part 2) As in part 1, since

$$x(\lambda) = -\sum_{i=q+1}^{m} \frac{(h^T u_i)u_i}{\lambda_i - \lambda},$$

then

$$l\left(x(\lambda)\right) = -\sum_{i=q+1}^{m} \frac{(h^T u_i)^2}{(\lambda_i - \lambda)^2}.$$

In the range $\lambda \in (-\infty, \lambda_{min})$ each of the terms $1/(\lambda_i - \lambda)^2$ $(i = q+1, \ldots, m)$ is a strictly monotonically increasing function of λ. Since $r > 0$, then $\|x(\lambda_{min})\|_2 > 0$ and therefore $\exists\, i > q$ such that $h^T u_i \neq 0$. Hence $|h^T u_i| > 0$ and consequently $l\left(x(\lambda)\right)$ is monotonically increasing in λ. $\Box$

The fact that the length increases monotonically in the interval $(-\infty, \lambda_{min})$ was first discovered by Reinsch [24]. Theorem 2.2 suggests an approach to solve the nonconvex optimization problem (5). At each iteration, a quadratic approximation of the potential function $\varphi(w)$ around the iterate w^k is optimized on an ellipsoid inscribed in the polytope $\{w \in \Re^m | A^T w \leq c\}$ and centered at w^k. Either a descent direction Δw^* of $\varphi(w)$ is produced by this optimization or w^k is said to be a local minimum. A new iterate w^{k+1} is computed such that $\varphi(w^{k+1}) < \varphi(w^k)$ by moving from w^k in the direction Δw^*. At each iteration the current iterate w^k is rounded off to the nearest ± 1 vertex: $\tilde{w}^k = (\pm 1, \ldots, \pm 1)$. If $\tilde{w}^k$ is such that $A^T \tilde{w}^k \leq c$ then $\tilde{w}^k$ is a global optimal solution of (5) and consequently a solution of INTEGER PROGRAMMING.

If a local minimum of (5) is found, the problem is modified by adding a cut and the algorithm is applied to the augmented problem. Let v be the integer solution rounded off from the local minimum. A valid cut is

$$v^T w \leq m - 2. \tag{17}$$

Proposition 2.4 *Cut (17) excludes v but does not exclude any other feasible integral solution of (1-2).*

Proof: Follows immediately from the fact that $v^T w = m$ if $w = v$ and $v^T w \leq m - 2$ otherwise. $\Box$

We note that adding a cut of the type above will not, theoretically, prevent the algorithm from converging to the same local minimum twice. However, the addition of the cut changes the objective function and, consequently, should alter the trajectory followed by the algorithm. As we will note later, in the section on experimental results, we observed no case in which the algorithm returned to a previously visited local minimum.

Figure 1 details pseudo-code for **procedure** `ip`, the integer programming algorithm. Procedure `ip` takes as input the A matrix, the c right hand side vector, an initial estimate γ_0 of parameter γ and initial lower and upper bounds on the acceptable length, $\underline{l}_0$ and $\overline{l}_0$, respectively. In the first line of `ip` the minor iteration counter (k), lower and upper bounds on the acceptable length region $(\underline{l}, \overline{l})$ and major iteration counter (K) are initialized. In

procedure $\texttt{ip}(A, c, \gamma_0, \underline{l}_0, \overline{l}_0)$

$k := 0;\;\; \gamma := \gamma_0;\;\; \underline{l} := \underline{l}_0;\;\; \overline{l} := \overline{l}_0;\;\; K := 0;$

$w^k := \texttt{get\_start\_point}(A, c);$

$\tilde{w}^k := \texttt{round\_off}(w^k);$

do $A^T \tilde{w}^k \not< c \rightarrow$

$\quad \Delta w^* := \texttt{descent\_direction}(\gamma, w^k, \underline{l}, \overline{l});$

$\quad$ **do** $\varphi(w^k + \alpha \Delta w^*) \geq \varphi(w^k)$ **and** $\overline{l} > \epsilon \rightarrow$

$\quad\quad \overline{l} := \overline{l} / \overline{l}_r;$

$\quad\quad \Delta w^* := \texttt{descent\_direction}(\gamma, w^k, \underline{l}, \overline{l})$

$\quad$ **od**;

$\quad$ **if** $\varphi(w^k + \alpha \Delta w^*) < \varphi(w^k) \rightarrow$

$\quad\quad w^{k+1} := w^k + \alpha \Delta w^*;$

$\quad\quad \tilde{w}^{k+1} := \texttt{round\_off}(w^{k+1});$

$\quad\quad k := k + 1$

$\quad$ **fi**;

$\quad$ **if** $\overline{l} \leq \epsilon \rightarrow$

$\quad\quad A := \texttt{new\_matrix}(A);\;\; c := \texttt{new\_rhs}(c);$

$\quad\quad k := 0;\;\; \gamma := \gamma_0;\;\; \underline{l} := \underline{l}_0;\;\; \overline{l} := \overline{l}_0;\;\; K := K + 1;$

$\quad\quad w^k := \texttt{get\_start\_point}(A, c);$

$\quad\quad \tilde{w}^k := \texttt{round\_off}(w^k)$

$\quad$ **fi**

od

end $\texttt{ip}$;

Figure 1: Pseudo-Code: The ip Algorithm

line 2, `get_start_point` returns a strict interior point of the polytope under consideration, i.e. $w^k \in \mathcal{L}_s$. In many situations this is a trivial task. In others, a phase I interior point linear programming algorithm may be required. In line 3, the array w^k is rounded off to the nearest ± 1 vertex by procedure `round_off` and the result is placed in array $\tilde{w}^k$.

The algorithm iterates in the loop between lines 4 and 21, terminating only when a feasible ± 1 integer solution $\tilde{w}^k$ is found. At each iteration, a descent direction of the potential function $\varphi(w)$ is produced in lines 5 through 9. In line 5 the optimization (18-19) is realized. Because of higher order terms the direction returned by `descent_direction` may not be a descent direction for $\varphi(w)$. Loop 6-9 is repeated until an improving direction for the potential function is produced or the largest acceptable length falls below a given tolerance ϵ. These two cases are treated in lines 10-14 and 15-20, respectively.

In case the direction produced is a descent for $\varphi(w)$, a new point w^{k+1} is defined (in line 11) by moving from the current iterate w^k in the direction Δw^* by a step length $\alpha < 1$. In line 12 this new point is rounded off and set to $\tilde{w}^{k+1}$.

If in loop 6-9 the largest acceptable length has fallen below ϵ we say the algorithm has converged to a local (not global) minimum. A new problem is defined in line 16 and the algorithm is restarted in lines 17-19.

3. Computing the Descent Direction

We now consider in more detail the computation of the direction of descent for the potential function. The algorithm described in this section is similar to the one in Moré and Sorensen [19]. However, our implementation differs from the Moré and Sorensen approach in several aspects. Moreover, Moré and Sorensen describe limited computational results on very small problem instances and it is not known how their implementation will perform for the potential function used in this paper. In this section we describe the algorithm in detail for completeness and so that our computational results can be duplicated and verified.

As discussed previously, the algorithm solves the optimization problem

$$\text{minimize } \frac{1}{2}(\Delta w)^T H \Delta w + h^T \Delta w \tag{18}$$

$$\text{subject to} : (\Delta w)^T A D^{-2} A^T \Delta w \leq r^2 \leq 1 \tag{19}$$

to produce a descent direction Δw^* for the potential function $\varphi(w)$. A solution $\Delta w^* \in \Re^m$ to (18-19) is optimal if and only if there exists $\mu \geq 0$ such that:

$$\left(H + \mu A D^{-2} A^T\right) \Delta w^* = -h \tag{20}$$

$$\mu\left((\Delta w^*)^T A D^{-2} A^T \Delta w^* - r^2\right) = 0 \tag{21}$$

$$H + \mu A D^{-2} A^T \text{is positive semidefinite.} \tag{22}$$

With the change of variables $\gamma = 1/(\mu + 1/n)$ and substituting (7) and (8) into (20) we obtain an expression for Δw^* satisfying (20):

$$\Delta w^* = -\left(AD^{-2}A^T - \frac{4\gamma}{f_0^2} w^k {w^k}^T - \frac{2\gamma}{f_0} I \right)^{-1} \gamma \left(-\frac{1}{f_0} w^k + \frac{1}{n} AD^{-1} e \right) \tag{23}$$

Note that r does not appear in (23). However, (23) is not defined for all values of r. Theorem 2.2 guarantees that if the radius r of the ellipsoid (19) is kept within a certain bound, then there exists an interval $0 \leq \gamma \leq \gamma_{max}$ such that

$$AD^{-2}A^T - \frac{4\gamma}{f_0^2} w^k {w^k}^T - \frac{2\gamma}{f_0} I \tag{24}$$

is nonsingular. The following proposition establishes that for γ small enough Δw^* is a descent direction of $\varphi(w)$.

Proposition 3.1 *There exists $\gamma > 0$ such that the direction Δw^*, given in (23), is a descent direction of $\varphi(w)$.*

Proof:

$$\begin{aligned}
\Delta w^* &= -\left(AD^{-2}A^T - \frac{4\gamma}{f_0^2} w^k {w^k}^T - \frac{2\gamma}{f_0} I \right)^{-1} \gamma \left(-\frac{1}{f_0} w^k + \frac{1}{n} AD^{-1} e \right) \\
&= -\left[AD^{-2}A^T \left\{ I - \gamma (AD^{-2}A^T)^{-1} \left(-\frac{4}{f_0^2} w^k {w^k}^T - \frac{2}{f_0} I \right) \right\} \right]^{-1} \times \\
&\quad \gamma \left(-\frac{1}{f_0} w^k + \frac{1}{n} AD^{-1} e \right) \\
&= -\gamma \left[I + \gamma (AD^{-2}A^T)^{-1} \left(\frac{4}{f_0^2} w^k {w^k}^T + \frac{2}{f_0} I \right) \right]^{-1} (AD^{-2}A^T)^{-1} \times \\
&\quad \left(-\frac{1}{f_0} w^k + \frac{1}{n} AD^{-1} e \right) \\
&= \gamma \left[I + \gamma (AD^{-2}A^T)^{-1} \left(\frac{4}{f_0^2} w^k {w^k}^T + \frac{2}{f_0} I \right) \right]^{-1} (AD^{-2}A^T)^{-1} (-h) \qquad (25)
\end{aligned}$$

Let $\gamma = \epsilon > 0$ and consider $\lim_{\epsilon \to 0^+} h^T \Delta w^*$. We have

$$\lim_{\epsilon \to 0^+} \Delta w^* = \epsilon \, (AD^{-2}A^T)^{-1}(-h)$$

and therefore

$$\lim_{\epsilon \to 0^+} h^T \Delta w^* = -\epsilon \, h^T (AD^{-2}A^T)^{-1} h.$$

Since, by assumption, $\epsilon > 0$ and $h^T (AD^{-2}A^T)^{-1} h > 0$ then

$$\lim_{\epsilon \to 0^+} h^T \Delta w^* < 0. \ \square$$

The idea of the algorithm is to solve (18-19), more than once if necessary, with the radius r as a variable. Parameter γ is varied until r takes a value in some given interval. Each iteration of this algorithm is comprised of two tasks. To simplify notation, let

$$H_c = AD^{-2}A^T \tag{26}$$

$$H_o = -\frac{4}{f_0^2} w^k {w^k}^T - \frac{2}{f_0} I \tag{27}$$

and define

$$M = H_c + \gamma H_o.$$

Given the current iterate w^k, we first seek a value of γ such that $M\Delta w = \gamma h$ has a solution Δw^*. This can be done by binary search, as we will see shortly. Once such a parameter γ is found, the linear system

$$M\Delta w^* = \gamma h \tag{28}$$

is solved for $\Delta w^* \equiv \Delta w^*(\gamma(r))$. Lemma 2.3 guarantees that the length $l(\Delta w^*(\gamma))$ is a monotonically increasing function of γ in the interval $0 \leq \gamma \leq \gamma_{max}$. Optimality condition (21) implies that $r = \sqrt{l(\Delta w^*(\gamma))}$ if $\mu > 0$. Small lengths result in small changes in the potential function, since r is small and the optimal solution lies on the surface of the ellipsoid. A length that is too large may not correspond to an optimal solution of (18-19), since this may require $r > 1$. We maintain an interval $(\underline{l}, \overline{l})$ called the *acceptable length region* and accept a length $l(\Delta w^*(\gamma))$ if $\underline{l} \leq l(\Delta w^*(\gamma)) \leq \overline{l}$. If $l(\Delta w^*(\gamma)) < \underline{l}$, γ is increased and (28) is resolved with the new M matrix and h vector. On the other hand, if $l(\Delta w^*(\gamma)) > \overline{l}$, γ is reduced and (28) is resolved. Once an acceptable length is produced we use $\Delta w^*(\gamma)$ as the descent direction. The notion of acceptable length region has been previously used by Moré [18], Gay [7] and Moré and Sorensen [19].

Figure 2 details pseudo-code for procedure `descent_direction`, where (18-19) is optimized. As input, procedure `descent_direction` is given an estimate for parameter γ, the current iterate w^k around which the inscribing ellipsoid is to be constructed and the current acceptable length region defined by $\underline{l}$ and $\overline{l}$. The value of γ passed to `descent_direction` at minor iteration k of `ip` is the value returned by `descent_direction` at minor iteration $k-1$. It returns a descent direction Δw^* of the quadratic approximation of the potential function $Q(w)$ from w^k, the next estimate for parameter γ and the current lower bound of the acceptable length region, $\underline{l}$.

In line 1 the length l is set to a large number and several logical keys are initialized: LD_{key} is **true** if a linear dependency in the rows of M is found during the solution of the

procedure `descent_direction`$(\gamma, w^k, \underline{l}, \overline{l})$

$l := \infty$; $LD_{key} :=$ **false**; $\overline{\gamma}_{key} :=$ **false**; $\underline{\gamma}_{key} :=$ **false**;

do $l > \overline{l}$ **or** $(l < \underline{l}$ **and** $LD_{key} =$ **false**$) \rightarrow$

    $M := H_c + \gamma H_o$; $b := \gamma h$;

    do $M \Delta w = b$ has no solution $\rightarrow$

        $\gamma := \gamma / \gamma_r$; $LD_{key} :=$ **true**;

        $M := H_c + \gamma H_o$; $b := \gamma h$

    od;

    $\Delta w^* := M^{-1} b$; $l := (\Delta w^*)^T A D^{-2} A^T \Delta w^*$;

    if $l < \underline{l}$ **and** $LD_{key} =$ **false** $\rightarrow$

        $\underline{\gamma} := \gamma$; $\underline{\gamma}_{key} :=$ **true**;

        if $\overline{\gamma}_{key} =$ **true** $\rightarrow \gamma := \sqrt{\underline{\gamma}\overline{\gamma}}$ **fi**;

        if $\overline{\gamma}_{key} =$ **false** $\rightarrow \gamma := \gamma \cdot \gamma_r$ **fi**

    fi;

    if $l > \overline{l} \rightarrow$

        $\overline{\gamma} := \gamma$; $\overline{\gamma}_{key} :=$ **true**;

        if $\underline{\gamma}_{key} =$ **true** $\rightarrow \gamma := \sqrt{\underline{\gamma}\overline{\gamma}}$ **fi**;

        if $\underline{\gamma}_{key} =$ **false** $\rightarrow \gamma := \gamma / \gamma_r$ **fi**

    fi

od;

do $l < \underline{l}$ **and** $LD_{key} =$ **true** $\rightarrow \underline{l} := \underline{l} / l_r$ **od**;

return(Δw^*)

end `descent_direction`;

Figure 2: Pseudo-Code: The `descent_direction` Algorithm

linear system (28) and is **false** otherwise; $\overline{\gamma}_{key}$ ($\underline{\gamma}_{key}$) is **true** if an upper (lower) bound for an acceptable γ has been found and **false** otherwise.

The nonconvex quadratic optimization on the ellipsoid is carried out in the loop going from line 2 to 19. The loop is repeated until either a length l is found such that $\underline{l} \leq l \leq \bar{l}$ or $l \leq \underline{l}$ due to a linear dependency found during the solution of (28) (i.e. if $LD_{key} =$ **true**). Lines 3 to 8 produce a descent direction which may not necessarily have an acceptable length. In line 3 the matrix M and the right hand side vector b are formed. The linear system (28) is tentatively solved in line 4. The solution procedure may not be successful (i.e. M may be singular). This implies that the parameter γ is too large. If this occurs, the parameter γ is reduced in line 5 of loop 4-7, which is repeated until a nonsingular matrix M is produced.

Once a nonsingular M matrix is available, a descent direction Δw^* is computed in line 8 along with its corresponding length l. Three cases can occur: (i) - the length is too small even though no linear dependency was detected in the factorization; (ii) - the length is too large; or (iii) - the length is acceptable. Case (iii) is the termination condition for the main loop 2-19. In lines 9-13 the first case is considered. The value of γ is a lower bound on an acceptable value of γ and is recorded in line 10 and the corresponding logical key is set. If an upper bound $\overline{\gamma}$ for an acceptable value of γ has been found the new estimate for γ is set to the geometric mean of $\underline{\gamma}$ and $\overline{\gamma}$ in line 11. Otherwise γ is increased by a fixed factor in line 12.

Similar to the treatment of case (i), case (ii) is handled in lines 14-18. The value of γ is an upper bound on an acceptable value of γ and is recorded in line 15 and the corresponding logical key is set. If a lower bound $\underline{\gamma}$ for an acceptable value of γ has been found the new estimate for γ is set to the geometric mean of $\underline{\gamma}$ and $\overline{\gamma}$ in line 16. Otherwise γ is decreased by a fixed factor in line 17.

Finally, in line 20, the lower bound $\underline{l}$ may have to be adjusted if $l < \underline{l}$ and $LD_{key} =$ **true**.

4. Experimental Results

We next present results of testing the algorithm described in this paper on a set of computational difficult zero-one integer programming problems. Fulkerson, Nemhauser and Trotter [6] describe a class of computationally difficult set covering problems that arise in computing the 1-width of incidence matrices of Steiner triple systems. They suggest that these are good problems for testing new algorithms for integer programming and set covering because they have far fewer variables than numerous solved problems in the literature, yet experience shows that they are hard to compute and verify. The β-width of a (0,1)-matrix A is the minimum number of columns that can be selected from A such that all row sums of the resulting submatrix of A are at least β. The 1-width of A is:

$$\mathcal{W}(A) = \min\ e_n^T x$$

$$\text{subject to}: Ax \geq e_m$$

$$x \geq 0 \text{ and integral,}$$

where e_n is an n-vector of ones and e_m an m-vector of ones. The 1-width is a set covering problem. The incidence matrices A that arise from Steiner triple systems have precisely 3 ones per row. Furthermore, for every pair of columns j and k there is exactly one row i for which $a_{ij} = a_{ik} = 1$. (i, j, k) are said to be a triple of A if there exists a row q such that $a_{qi} = a_{qj} = a_{qk} = 1$. Hall [10] discusses this structure in detail and shows a standard technique for recursively generating Steiner systems for which $n = 3^k, (k = 1, 2, 3, \ldots)$. A_3 is the 1×3 matrix of ones. A_{3n} is obtained from A_n as follows: The columns of A_{3n} are indexed $\{(i, j), 1 \leq i \leq n, 1 \leq j \leq 3\}$. The set $\{(i, r), (j, s), (k, t)\}$ is a triple of A_{3n} if and only if one of the following holds:

- $r = s = t$ and $\{i, j, k\}$ is a triple of A_n, or
- $i = j = k$ and $\{r, s, t\} = \{1, 2, 3\}$, or
- $\{i, j, k\}$ is a triple of A_n and $\{r, s, t\} = \{1, 2, 3\}$.

We refer to instances of set covering problems that arise from Steiner triple systems by their incidence matrices. Two examples for which $n \neq 3^k$ are given in [6]: A_{15} and A_{45}.

Consider the seed matrices A_3 and A_{15} below:

$$A_3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad A_{15} = \begin{bmatrix} Z & E & 0 \\ 0 & Z & E \\ E & 0 & Z \\ I & I & I \end{bmatrix}$$

where I is a 5×5 identity matrix and

$$Z = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matrices A_{27}, A_{81}, $A_{243}, \ldots$ and A_{45}, A_{135}, $A_{405}, \ldots$ are generated using the recursion

$$A_{3n} = \begin{bmatrix} A_n & 0 & 0 \\ 0 & A_n & 0 \\ 0 & 0 & A_n \\ I & I & I \\ [& \tilde{A}_{3n} &] \end{bmatrix} \tag{29}$$

where I is an $n \times n$ identity matrix and $\tilde{A}_{3n}$ consists of $3! = 6$ blocks corresponding to the permutations of $\{1, 2, 3\}$; for each permutation π, the triples of the corresponding block are

$$\{(i, \pi(1)), (j, \pi(2)), (k, \pi(3))\}$$

where (i, j, k) runs through all triples of A_n.

Fulkerson, Nemhauser and Trotter [6] discuss computational experience with A_9, A_{15}, A_{27} and A_{45}. They are able to solve A_9 with a cutting plane code after generating 44 cuts, but this approach fails with the three other problems. Using an implicit enumeration algorithm similar to the one described in [8] they are able to solve A_{15} and A_{27} but not A_{45}. Avis [1] reports that A_{45} was solved in 1979 by H. Ratliff, requiring over two and a half hours on an Amdahl V7 computer. Using an IBM 3033, Bausch [2] solved A_{27} and A_{45} in 25.1s and 527.1s, respectively. Avis also suggests why these problems may be so difficult to solve by showing that any branch and bound algorithm that uses a linear programming relaxation, and/or elimination by dominance requires the examination of $2^{\sqrt{2m/3}}$ partial solutions, where m is the number of variables of the integer program.

Before we describe the experimental results we make some final remarks regarding the algorithmic setup used to run the experiments. Namely, we will describe the ± 1 integer programming formulation of the 1-width problem, the starting solution w^0 for the interior point algorithm and how rounding off is done.

In the feasibility version of the problem what is wanted is a set cover of size at most k. Let the decision variable be

$$w_j = \begin{cases} -1 & \text{if column } j \text{ is in the cover} \\ +1 & \text{otherwise} \end{cases}$$

In the objective function constraint, at most k variables can contribute a -1, while the remaining $m-k$ must take on the value $+1$, i.e.

$$\sum_{j=1}^{m} w_j \geq -k + (m-k) = -2k + m$$

and therefore

$$-\sum_{j=1}^{m} w_j \leq 2k - m. \tag{30}$$

Of the three variables in each set covering constraint, at least one must take value -1, while the other two can be $+1$ or -1. Hence

$$\sum_{j=1}^{m} a_{ij} w_j \leq 1, \quad i = 1, \ldots, n.$$

Finally, all variables are bounded above by $+1$ and below by -1,

$$-1 \leq w_j \leq 1, \quad j = 1, \ldots, m.$$

We propose two rounding schemes. Rounding scheme $\mathcal{A}$ is done as follows,

$$\tilde{w_j} = \begin{cases} +1 & \text{if } w_j > 0 \\ -1 & \text{if } w_j \leq 0. \end{cases}$$

In rounding scheme $\mathcal{B}$ the entries of the iterate w are sorted in increasing order and a set cover is built greedily, based on the sorted w vector. From this set cover, local neighbor solutions are constructed by performing all two-exchanges (of columns i and j) for which $|w_i - w_j| < 10^{-8}$.

The 1-widths of incidence matrices of Steiner triple systems have the characteristic that $\mathcal{W}(A) > m/2$, and therefore one could use as an initial interior point the origin, $w^0 = (0, \ldots, 0)^T$. Instead, we choose to use $w_j^0 = -(2k-m)/(m+1), \quad j = 1, \ldots, m$, an interior point that rounds off with scheme $\mathcal{A}$ to a cover containing all of the columns of the

Problem	Integer Program Variables/Constraints	Best Known Cover	Optimal
A_{27}	27/116	18	yes
A_{45}	45/330	30	yes
A_{81}	81/1080	61	unknown
A_{135}	135/3015	105	unknown
A_{243}	243/9801	204	unknown

Table I: Problem Set

matrix A. We also use this starting solution as the initial interior point every time a cut is generated.

We use the following algorithm parameter settings for all problem instances: $\gamma_0 = 32$, $\underline{l}_0 = 0.5$, $\bar{l}_0 = 1.0$, $\epsilon = 10^{-12}$, $\bar{l}_r = 4$, $\alpha = 0.5$, $\gamma_r = \sqrt{2}$ and $l_r = 4$. Our implementation uses a direct Cholesky factorization to solve (28) at each iteration. This implementation takes no advantage of the problem structure. The algorithm was implemented in FORTRAN and the tests were carried out on a Silicon Graphics IRIS ®workstation, model 4D/340S, running IRIX System V Release 3.2.3. The `f77` compiler was used to compile the code using the optimization flag `-O2 -Olimit 800`. All times reported are user times given by the system call `times()`.

We ran the interior point algorithm on five set covering problems: A_{27}, A_{45}, A_{81}, A_{135} and A_{243}. Both rounding scheme implementations were run on all instances except problem A_{243} where only rounding scheme $\mathcal{B}$ was used. Problems A_{27} and A_{45} are taken from [6] and problems A_{81}, A_{135} and A_{243} were generated using recursion (29). Of these problems, optimal solutions are known for only the first two. Feo and Resende [5] have produced a cover of size 61 for A_{81} and 204 for A_{243}. It is widely conjectured that $\mathcal{W}(A_{81}) = 61$. Using their code, a cover of size 105 was produced for A_{135}. Table I shows the test problems and the size of the best known cover for each. There, the number of constraints excludes upper and lower bounds on the variables as well as the cover cardinality constraint (30). For each instance, we set the cover cardinality constraint to make the algorithm search for the best known cover, as well as a few larger covers. Tables II and III summarize the results for rounding schemes $\mathcal{A}$ and $\mathcal{B}$, respectively.

Figure 3 summarizes the run of the rounding scheme $\mathcal{A}$ implementation on problem A_{27}. It illustrates the behavior of $f_0 = m - w^T w$, a measure of nonintegrality of the interior point solution, the length $l(\Delta w)$ and the sum of the violations of the rounded solution $\tilde{w}$, $\sum_{j=1}^{n} \max\{0, a_j^T \tilde{w} - c_j\}$, as a function of the minor iterations. Minor iterations are the

Problem	Search Size	Major Iterations	Itrs/Major Iteration	CPU Time	Time/Major Iteration	Size Cover Found
A_{27}	18	1	162.0	3.62s	3.62s	18
A_{45}	32	2	67.0	4.44s	4.44s	31
A_{45}	31	2	147.0	18.59s	9.29s	31
A_{45}	30	5	229.6	1m19.45s	15.89s	30
A_{81}	65	8	335.9	13m05.17s	1m38.15s	63
A_{81}	61	23	334.3	42m36.44s	1m51.15s	61
A_{135}	107	19	787.1	4h55m37.26s	15m33.54s	107
A_{135}	106	38	840.1	11h59m20.88s	18m55.81s	106
A_{135}	105	0	0.0	0s	0s	105

Table II: Summary of results (Rounding scheme $\mathcal{A}$)

Problem	Search Size	Major Iterations	Itrs/Major Iteration	CPU Time	Time/Major Iteration	Size Cover Found
A_{27}	18	1	2.0	0.05s	0.05s	18
A_{45}	32	1	6.0	0.45s	0.45s	32
A_{45}	31	1	66.0	4.35s	4.35s	31
A_{45}	30	8	152.0	1m39.30s	12.41s	30
A_{81}	65	1	30.0	11.11s	11.11s	65
A_{81}	64	3	313.3	4m34.51s	1m31.50s	64
A_{81}	63	3	328.0	4m52.09s	1m37.36s	63
A_{81}	62	3	393.7	5m51.76s	1m57.25s	62
A_{81}	61	3	407.0	6m02.80s	2m00.93s	61
A_{135}	107	3	652.0	36m06.75s	12m02.25s	107
A_{135}	106	5	738.4	1h09m13.00s	13m50.68s	106
A_{135}	105	112	902.4	56h24m35.70s	30m13.18s	105
A_{243}	206	2	653.0	2h23m19.02s	1h11m29.51s	206
A_{243}	205	5	913.4	8h10m57.00s	1h38m11.40s	205
A_{243}	204	0	0.0	0s	0s	204

Table III: Summary of results (Rounding scheme $\mathcal{B}$)

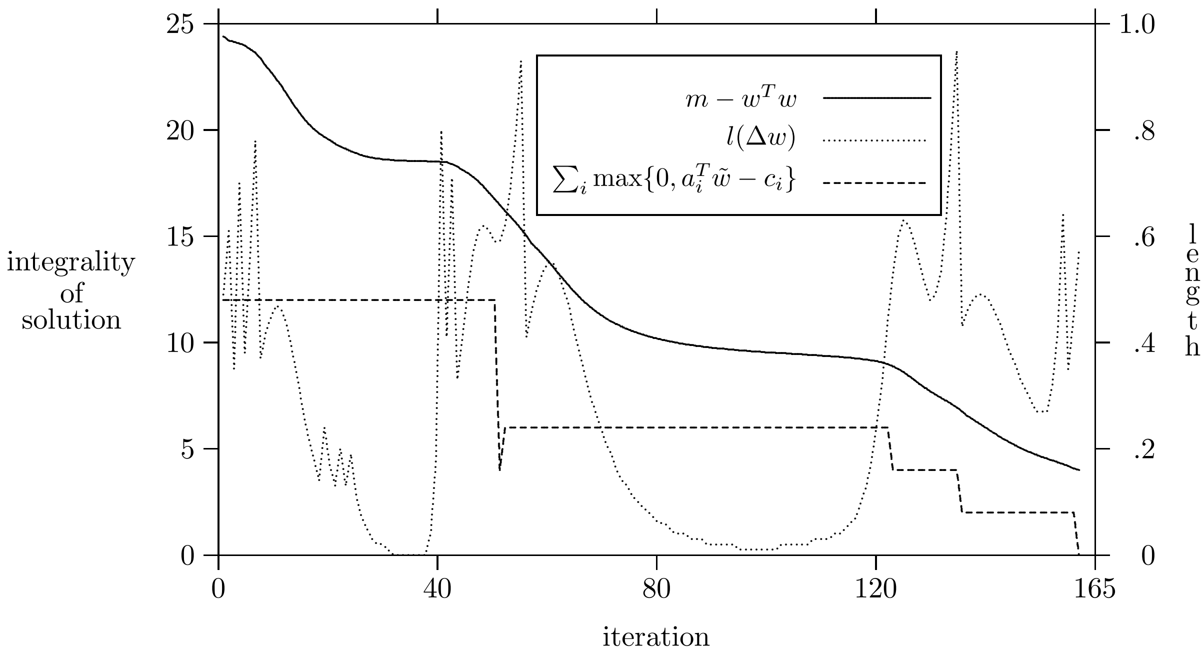

Figure 3: Iteration summary: A_{27} with rounding $\mathcal{A}$

iterations from the start (or restart) of the algorithm until a local (maybe global) minimum is found. It is interesting to observe the behavior of the length. It typically shrinks when the algorithm encounters a flat region and increases abruptly whenever the iterate escapes the flat region or whenever a new improved rounded solution is found. This behavior is observed on all problem instances.

Figure 4 shows the distribution of the interior point solution from which the algorithm rounded off to the cover of size 61 using rounding scheme $\mathcal{A}$. Note that all but five components are close to a ± 1 value.

Figure 5 summarizes the run of the rounding scheme $\mathcal{B}$ implementation on problem A_{81}. The cover size and $f_0 = m - w^T w$ are both plotted as a function of minor iterations. One can observe that the algorithm converges to two nonglobal local minima before finally going to an interior point from where rounding scheme $\mathcal{B}$ rounds to a cover of size 61. The first cover of size 64 or lower was only found in the third major iteration.

We make the following comments regarding the numerical experiments.

- The main objective of the numerical experimentation is to show empirically that the interior point algorithm converges to an interior point from which one can round off to a good cover. Not much effort was devoted to efficient coding of the algorithm. Specifically, the factorization used to solve the linear system in `descent_direction` was implemented in dense form, leaving much room for improvement. Consequently, our implementation is not as fast as the probabilistic heuristic of Feo and Resende

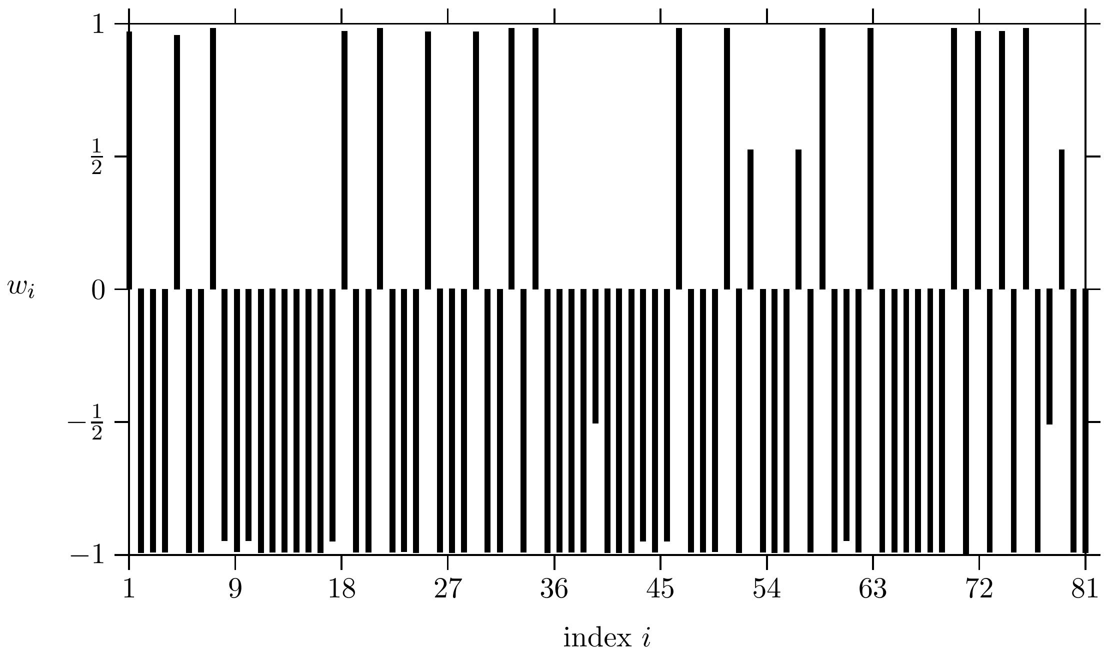

Figure 4: Final interior solution: A_{81} with rounding $\mathcal{A}$

[5] for this class of set covering problems. Table IV summarizes running times for that probabilistic heuristic tested on the same machine used to test our interior point code.

- The algorithm has successfully produced the best known covers for all five instances tested with both rounding schemes (except for A_{243} that was only tested with scheme $\mathcal{B}$).

- Scheme $\mathcal{B}$ is superior to the straightforward rounding scheme $\mathcal{A}$. Even though it is more time consuming it usually requires fewer iterations to find a good cover than does scheme $\mathcal{A}$, e.g. for problem A_{81} scheme $\mathcal{A}$ required over six times more iterations and over seven times more running time to find a cover of size 61 than scheme $\mathcal{B}$.

- Using scheme $\mathcal{A}$, the algorithm encountered some difficulty in finding a cover of size 62 for A_{81}. Interestingly, the probabilistic heuristic of Feo and Resende did not find a cover of size 62 either. We conjecture that there exist more minimal covers of size 61 than of size 62 for A_{81}.

- The algorithm is sensitive to machine precision. This can be illustrated by running the rounding scheme $\mathcal{A}$ implementation on a machine with higher precision (a VAX 8810). For problem A_{45}, the code takes one major iteration on the VAX while taking five on the Silicon Graphics.

Problem	Iterations	CPU Time	Size Cover Found
A_{27}	1	0.05s	19
A_{27}	3	0.07s	18
A_{45}	1	0.15s	33
A_{45}	3	0.17s	31
A_{45}	3096	24.88s	30
A_{81}	1	0.52s	63
A_{81}	376	12.47s	61
A_{135}	1	1.37s	108
A_{135}	5	1.77s	107
A_{135}	31	4.52s	106
A_{135}	108	12.58s	105
A_{243}	1	4.78s	207
A_{243}	31	25.27s	206
A_{243}	47	36.48s	205
A_{243}	205	2m21.75s	204

Table IV: Summary of results (Probabilistic heuristic of Feo & Resende)

Problem	Mean iterations `descent_direction`
A_{27}	2.5
A_{45}	1.53
A_{81}	1.13
A_{135}	1.06
A_{243}	1.08

Table V: Average number of iterations: `descent_direction` with rounding scheme $\mathcal{B}$

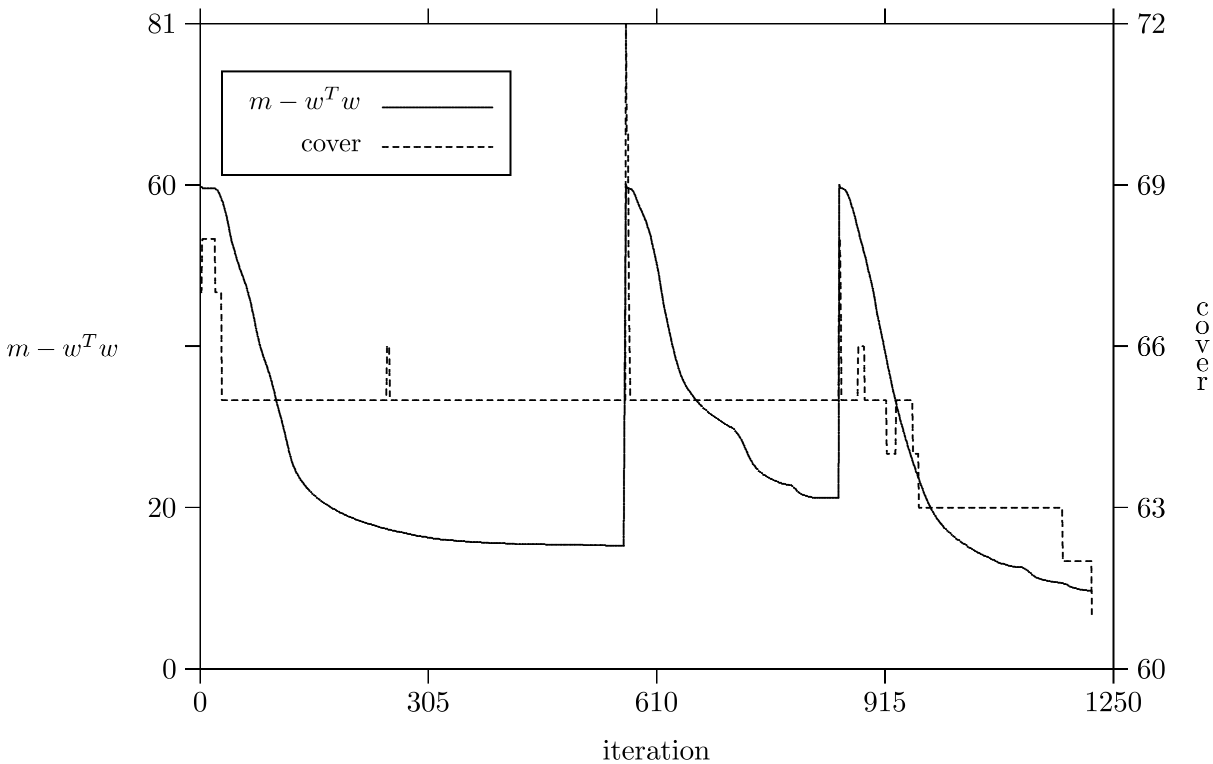

Figure 5: Iteration summary: A_{81} with rounding $\mathcal{B}$

- The cut generation strategy to deal with nonglobal local minima succeeded in changing the path generated by the algorithm to eventually lead to an interior point from which one can round off to an integral feasible solution. Figure 4 shows that on problem A_{81}, the final interior point is indeed close to an integral feasible solution. This characteristic was observed on all instances tested.

- All cuts generated were distinct. This, however, cannot always be guaranteed to occur.

- Figure 3 illustrates an interesting correlation between the trust region and convergence of the algorithm. Prior to reaching an interior point from which a better (less infeasible) integer solution can be generated, the length (or trust region) increases.

- Moré and Sorensen [19] report a value of 1.63 for the average number of iterations (counted by the number of calls to the linear system solver) in their algorithm for quadratic optimization on an ellipsoid. In our implementation of `descent_direction`, this number is as low as 1.06, as summarized in Table V.

5. Concluding Remarks

In this paper, we have introduced an interior point algorithm for zero-one integer programming feasibility. We have described in detail the procedure of generating a descent direction for a potential function whose global minimum corresponds to a feasible zero-one integer solution. We show how to incorporate this descent direction procedure in a general purpose algorithm for zero-one integer programming.

To illustrate the applicability of this general purpose algorithm, we use it to solve several instances of computationally difficult set covering problems that arise from computing the 1-width of the incidence matrix of Steiner triple systems. In that context, we show how to start the algorithm, propose two rounding schemes and show how to proceed when a nonglobal local minimum is encountered. We have found optimal covers for two instances with known optimal solutions and the best known covers for instances varying in size from 81 variables and 1080 constraints to 243 variables and 9801 constraints.

The code used to produce the experimental results solves the descent direction linear system by direct Cholesky factorization, taking no advantage of problem structure. One may expect improved performance with an implementation under development that takes advantage of sparsity and nonzero structure of the linear system.

6. Acknowledgement

The authors acknowledge suggestions from C.C. Gonzaga, D.S. Johnson and two anonymous referees that much improved the paper.

References

[1] D. Avis. A note on some computationally difficult set covering problems. *Mathematical Programming*, 18:138–145, 1980.

[2] D. O. Bausch. Computational advances in the solution of large-scale set covering and set partitioning problems. Master's thesis, Naval Postgraduate School, Monterey, CA, October 1982.

[3] V.J. Bowman and F. Glover. A note on zero-one integer and concave programming. *Operations Research*, 20:182–183, 1972.

[4] B.H. Faaland and F.S. Hillier. Interior path methods for heuristic integer programming procedures. *Operations Research*, 27:1069–1087, 1979.

[5] T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

[6] D.R. Fulkerson, G.L. Nemhauser, and L.E. Trotter Jr. Two computationally difficult set covering problems that arise in computing the 1-width of incidence matrices of Steiner triple systems. *Mathematical Programming Study*, 2:72–81, 1974.

[7] D.M. Gay. Computing optimal locally constrained steps. *SIAM J. Sci. Stat. Comput.*, 2:186–197, 1981.

[8] A.M. Geoffrion. An improved implicit enumeration approach for integer programming. *Operations Research*, 17:437–454, 1969.

[9] F. Glover and D. Klingman. Concave programming applied to a special class of 0-1 integer programs. *Operations Research*, 21:135–140, 1973.

[10] M. Hall Jr. *Combinatorial theory.* Blaisdell Company, Waltham, MA, 1967.

[11] F.S. Hillier. Efficient heuristic procedures for integer linear programming with an interior. *Operations Research*, 17:600–637, 1969.

[12] T. Ibaraki, T. Ohashi, and F. Mine. A heuristic algorithm for mixed-integer programming problems. *Mathematical Programming Study*, 2:115–136, 1974.

[13] R.G. Jeroslow and T.H.C. Smith. Experimental results on Hillier's linear search. *Mathematical Programming*, 9:371–376, 1975.

[14] B. Kalantari and J.B. Rosen. Penalty for zero-one integer equivalent problem. *Mathematical Programming*, 24:229–232, 1982.

[15] N. Karmarkar. An interior-point approach to NP-complete problems. In *Proceedings of the Mathematical Programming Society Conference on Integer Programming and Combinatorial Optimization*, pages 351–366. University of Waterloo, May 1990.

[16] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[17] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[18] J.J. Moré. The Levenberg-Marquandt algorithm: Implementation and theory. In G.A. Watson, editor, *Proceedings of the Dundee Conference on Numerical Analysis.* Springer-Verlag, 1978.

[19] J.J. Moré and D.C. Sorensen. Computing a trust region step. *SIAM J. Sci. Stat. Comput.*, 4:553–572, 1983.

[20] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization.* John Wiley & Sons, 1988.

[21] P.M. Pardalos and J.B. Rosen. Methods for global concave minimization: A bibliographic survey. *Siam Review*, 28:367–379, 1986.

[22] P.M. Pardalos and J.B. Rosen. *Constrained global optimization: Algorithms and applications.* Lecture Notes in Computer Science 268. Springer-Verlag, 1987.

[23] M. Raghavachari. On connections between zero-one integer programming and concave programming under linear constraints. *Operations Research*, 17:680–684, 1969.

[24] C.H. Reinsch. Smoothing by spline functions II. *Numer. Math.*, 16:451–454, 1971.

[25] S. Sahni. Computationally related problems. *SIAM Journal of Computing*, 3:262–279, 1974.

[26] A. Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1986.

[27] Y. Ye. On the interior algorithms for nonconvex quadratic programming. Technical report, Integrated Systems Inc., Santa Clara, CA, 1988.