

Metaheuristic Search with Inequalities and Target Objectives for Mixed Binary Optimization Part I: Exploiting Proximity

Fred Glover, OptTek Systems, Inc., USA

Saïd Hanafi, University Lille Nord de France, France

ABSTRACT

Recent adaptive memory and evolutionary metaheuristics for mixed integer programming have included proposals for introducing inequalities and target objectives to guide the search. These guidance approaches are useful in intensification and diversification strategies related to fixing subsets of variables at particular values, and in strategies that use linear programming to generate trial solutions whose variables are induced to receive integer values. In Part I (the present article), we show how to improve such approaches by new inequalities that dominate those previously proposed and by associated target objectives that underlie the creation of both inequalities and trial solutions. Part I focuses on exploiting inequalities in target solution strategies by including partial vectors and more general target objectives. We also propose procedures for generating target objectives and solutions by exploiting proximity in original space or projected space. Part II of this study (to appear in a subsequent issue) focuses on supplementary linear programming models that exploit the new inequalities for intensification and diversification, and introduce additional inequalities from sets of elite solutions that enlarge the scope of these models. Part II indicates more advanced approaches for generating the target objective based on exploiting the mutually reinforcing notions of reaction and resistance. Our work in the concluding segment, building on the foundation laid in Part I, examines ways our framework can be exploited in generating target objectives, employing both older adaptive memory ideas of tabu search and newer ones proposed here for the first time.

Keywords: Adaptive Search, Parametric Tabu Search, Valid Inequalities, Zero-One Mixed Integer Programming

1. NOTATION AND PROBLEM FORMULATION

$$(MIP) \begin{cases} \text{Minimize} & x_0 = fx + gy \\ \text{subject to} & (x, y) \in Z = \{(x, y) : Ax + Dy \geq b\} \\ & x \text{ integer} \end{cases}$$

We represent the mixed integer programming problem in the form

We assume that $Ax + Dy \geq b$ includes the inequalities $U_j \geq x_j \geq 0, j \in N = \{1, \dots, N\}$, where

DOI: 10.4018/jamc.2010102601

some components of U_j may be infinite. The linear programming relaxation of (MIP) that results by dropping the integer requirement on x is denoted by (LP). We further assume $Ax + Dy \geq b$ includes an objective function constraint $x_0 \leq U_0$, where the bound U_0 is manipulated as part of a search strategy for solving (MIP), subject to maintaining $U_0 < x_0^*$, where x_0^* is the x_0 value for the currently best known solution x^* to (MIP).

The current paper focuses on the zero-one version of (MIP) denoted by (MIP:0-1), in which $U_j = 1$ for all $j \in N$. We refer to the LP relaxation of (MIP:0-1) likewise as (LP), since the identity of (LP) will be clear from the context,

In the following we make reference to two types of search strategies: those that fix subsets of variables to particular values within approaches for exploiting strongly determined and consistent variables, and those that make use of solution targeting procedures. As developed here, the latter solve a linear programming problem $LP(x', c')$ that includes the constraints of (LP) (and additional bounding constraints in the general (MIP) case) while replacing the objective function x_0 by a linear function $v_0 = c'x$. The vector x' is called a *target solution*, and the vector c' consists of integer coefficients c'_j that seek to induce assignments $x_j = x'_j$ for different variables with varying degrees of emphasis.

We adopt the convention that each instance of $LP(x', c')$ implicitly includes the (LP) objective of minimizing the function $x_0 = fx + gy$ as a secondary objective, dominated by the objective of minimizing $v_0 = c'x$, so that the true objective function consists of minimizing $\omega_0 = Mv_0 + x_0$, where M is a large positive number. As an alternative to working with ω_0 in the form specified, it can be advantageous to solve $LP(x', c')$ in two stages. The first stage minimizes $v_0 = c'x$ to yield an optimal solution $x = x''$ (with objective function value $v_0'' = c'x''$), and the second stage enforces $v_0 = v_0''$ to solve the residual problem of minimizing $x_0 = fx + gy$.²

A second convention involves an interpretation of the problem constraints. Selected instances of inequalities generated by approaches of the following sections will be understood

to be included among the constraints $Ax + Dy \geq b$ of (LP). In our definition of $LP(x', c')$ and other linear programs related to (LP), we take the liberty of representing the currently updated form of the constraints $Ax + Dy \geq b$ by the compact representation $x \in X = \{x: (x, y) \in Z\}$, recognizing that this involves a slight distortion in view of the fact that we implicitly minimize a function of y as well as x in these linear programs.³

To launch our investigation of the problem (MIP:0-1) we first review previous ideas for generating guiding inequalities for this problem in Section 2 and associated target objective strategies using partial vectors and more general target objectives in Section 3. We then present new inequalities in Section 4 that improve on those previously proposed. The fundamental issue of creating the target objectives that can be used to generate the new inequalities and that lead to trial solutions for (MIP: 0-1) by exploiting proximity is addressed in Section 5. Concluding remarks are given in Section 6.

2. EXPLOITING INEQUALITIES IN TARGET SOLUTION STRATEGIES

Let x' denote an arbitrary solution, and define the associated index sets

$$\begin{aligned} N(x', v) &= \{j \in N: x'_j = v\} \text{ for } v \in \{0, 1\}, \\ N(x') &= \{j \in N: x'_j \in \{0, 1\}\} \text{ and} \\ N^*(x') &= \{j \in N: x'_j \in]0, 1[\}, \end{aligned}$$

we have $N = N(x') \cup N^*(x')$. For any real number z , $\lceil z \rceil$ and $\lfloor z \rfloor$ respectively identify the least integer $\geq z$ and the greatest integer $\leq z$.

Define

$$\delta(x', x) = \sum_{j \in N} x_j (1 - x'_j) + x'_j (1 - x_j) \quad (1)$$

Proposition 1. *Let x' denote an arbitrary binary solution. Then the inequality*

$$\delta(x', x) \geq 1 \quad (1.1)$$

eliminates the assignment $x = x'$ as a feasible solution, but admits all other binary x vectors.

Proof: It is evident that $\delta(x', x) = \|x - x'\|_1 = \|x - x'\|_2$, so for all $x \neq x'$, we have $\delta(x', x) > 0$. The proposition follows from the fact that the value $\delta(x', x)$ is integer. \square

Remark 1: The inequality (1.1) has been used, for example, to produce 0-1 "short hot starts" for branch and bound by Spielberg and Guignard (2000) and Guignard and Spielberg (2003).

The constraint (1.1) is called a *canonical cut* on the unit hypercube by Balas and Jeroslow (1972). The constraint (1.1) has also been used by Soyster et al. (1978), Hanafi and Wilbaut (2006) and Wilbaut and Hanafi (2006).

Proposition 1 has the following consequence.

Corollary 1. Let x' denote an arbitrary binary solution. Then the inequality

$$\delta(x', x) \leq n - 1 \quad (1.2)$$

eliminates the assignment $x = e - x'$ (the complement of x') as a feasible solution, but admits all other binary x vectors.

Proof: Immediate from the proof on Proposition 1, by using $e - x'$. \square

We make use of solutions such as x' by assigning them the role of *target solutions*. In this approach, instead of imposing the inequality (1.1) we adopt the strategy of first seeing how close we can get to satisfying $x = x'$ by solving the LP problem⁴

LP(x'): Minimize $\{\delta(x', x): x \in X\}$

whereas earlier, $X = \{x: (x, y) \in Z\}$. We call x' the *target solution* for this problem. Let x'' denote an optimal solution to LP(x'). If the target solution x' is feasible for LP(x') then it is also uniquely optimal for LP(x') and hence $x'' = x'$, yielding $\delta(x', x'') = 0$. In such a case, upon testing x' for feasibility in (MIP:0-1) we can impose the inequality (1.1) as indicated earlier in order to avoid examining the solution again. However, in the case where x' is not feasible for LP(x'), an optimal solution x'' will yield $\delta(x', x'') > 0$ and since the distance $\delta(x, x')$ is an integer value we may impose the valid inequality

$$\delta(x, x') \geq \lceil \delta(x', x'') \rceil \quad (2.1)$$

The fact that $\delta(x', x'') > 0$ discloses that (2.1) is at least as strong as (1.1). In addition, if the solution x'' is a binary vector that differs from x' , we can also test x'' for feasibility in (MIP:0-1) and then redefine $x' = x''$, to additionally append the constraint (1.1) for this new x' . Consequently, regardless of whether x'' is binary, we eliminate x'' from the collection of feasible solutions as well as obtaining an inequality (2.1) when $\delta(x', x'')$ is fractional that dominates the original inequality (1.1).

Upon generating the inequality (2.1) (and an associated new form of (1.1) if x'' is binary), we continue to follow the policy of incorporating newly generated inequalities among the constraints defining X , and hence those defining Z of (MIP:0-1). Consequently, we assure that X excludes both the original x' and the solution x'' . This allows the problem LP(x') to be re-solved, either for x' as initially defined or for a new target vector (which can be also be x'' if the latter is binary), to obtain another solution x'' and a new (2.1).

Remark 2: The same observations can be made to eliminate the complement of x' , i.e. $(e - x')$, by solving the following LP problem:

LP*(x'): Maximize $(\delta(x', x): x \in X)$

Let $x^{+''}$ denote an optimal solution to $LP^+(x')$. If the complement of the target solution x' is feasible for $LP^+(x')$ then it is also uniquely optimal for $LP^+(x')$ and hence $x^{+''} = e - x'$, yielding $\delta(x', x^{+''}) = n$. In such a case, upon testing $e - x'$ for feasibility in (MIP:0-1) we can impose the inequality (1.2) as indicated earlier in order to avoid examining the solution again. However, in the case where $e - x'$ is not feasible for $LP^+(x')$, an optimal solution $x^{+''}$ will yield $\delta(x', x^{+''}) < n$ and we may impose the valid inequality

$$\delta(x', x) \leq \lfloor \delta(x', x^{+''}) \rfloor \quad (2.2)$$

The fact that $\delta(x', x^{+''}) < n$ discloses that (2.2) is at least as strong as (1.2).

It is worthwhile to use simple forms of tabu search memory based on recency and frequency in such processes to decide when to drop previously introduced inequalities, in order to prevent the collection of constraints from becoming unduly large. Such approaches can be organized in a natural fashion to encourage the removal of older constraints and to discourage the removal of constraints that have more recently or frequently been binding in the solutions to the $LP(x')$ problems produced (see Glover & Laguna, 1997; Glover & Hanafi, 2002). Older constraints can also be replaced by one or several surrogate constraints.

The strategy for generating a succession of target vectors x' plays a critical role in exploiting such a process. The feasibility pump approach of Fischetti, Glover and Lodi (2005) applies a randomized variant of nearest neighbor rounding to each non-binary solution x'' to generate the next x' , but does not make use of associated inequalities such as (1.x) and (2.x). In subsequent sections we show how to identify more effective inequalities and associated target objectives to help drive such processes.

3. GENERALIZATION TO INCLUDE PARTIAL VECTORS AND MORE GENERAL TARGET OBJECTIVES

We extend the preceding ideas in two ways, drawing on ideas of parametric branch and bound and parametric tabu search (Glover, 1978, 2006a). First we consider *partial x vectors* that may not have all components x_j determined, in the sense of being fixed by assignment or by the imposition of bounds. Such vectors are relevant in approaches where some variables are compelled or induced to receive particular values, while others remain free or are subject to imposed bounds that are not binding.

Let x' denote an arbitrary solution and $J \subseteq N(x')$ define the associated set

$$F(J, x') = \{x \in [0,1]^n : x_j = x'_j \text{ for } j \in J\}$$

Let x, x' two arbitrary binary solutions and $J \subseteq N$, define

$$\delta(J, x', x) = \sum_{j \in J} x_j (1 - x'_j) + x'_j (1 - x_j) \quad (3)$$

Proposition 2. *Let x' denote an arbitrary binary solution and $J' \subseteq N(x')$. Then the inequality*

$$\delta(J', x', x) \geq 1 \quad (3.1)$$

eliminates all solutions in $F(J', x')$ as a feasible solution, but admits all other binary x vectors.

Proof: *It is evident that for all $x \in F(J', x')$, we have $\delta(J', x', x) = 0$. \square*

Proposition 2 has the following consequence.

Corollary 2. *Let x' denote an arbitrary binary solution and $J' \subseteq N(e - x')$. Then the inequality*

$$\delta(J', x', x) \leq |J'| - 1 \tag{3.2} \quad F(x') = F(x', N(x')) = \{x \in [0,1]^n: x_j = x'_j \text{ for } j \in N(x')\}.$$

eliminates all solutions in $F(J', e - x')$ as a feasible solution, but admits all other binary x vectors.

Proof: Immediate from the proof on Proposition 2, by using $e - x'$.

We couple the target solution x' with the associated set $J' \subseteq N(x')$ to yield the problem

LP(x', J'): Minimize $(\delta(J', x', x): x \in X)$.

An optimal solution to $LP(x', J')$, as a generalization of $LP(x')$, will likewise be denoted by x'' . We obtain the inequality

$$\delta(J', x', x) \geq \lceil \delta(J', x', x'') \rceil \tag{4.1}$$

By an analysis similar to the derivation of (2.1), we observe that (4.1) is a valid inequality, i.e., it is satisfied by all binary vectors that are feasible for (MIP:0-1) (and more specifically by all such vectors that are feasible for $LP(x', J')$), with the exception of those ruled out by previous examination.

Remark 3: The same observations can be made to eliminate all solutions in $F(J', e - x')$ as a feasible solution by solving the following LP problem:

LP+(x', J'): Maximize $(\delta(J', x, x'): x \in X)$.

We obtain the inequality

$$\delta(J', x', x) \leq \lfloor \delta(J', x', x'') \rfloor \tag{4.2}$$

where x'' is an optimal solution to $LP+(x', J')$.

In the special case where $J' = N(x')$, we have the following properties. Let $x' \in [0,1]^n$ define the associated set

Let k be an integer satisfying $0 \leq k \leq n - |N(x')|$, the canonical hyperplane associated to the solution x' , denoted $H(x', k)$ is defined by

$$H(x', k) = \{x \in [0,1]^n: \delta(N(x'), x', x) = k\}.$$

Proposition 3. $x \in H(x', k) \cap \{0,1\}^n \Leftrightarrow \delta(x, F(x') \cap \{0,1\}^n) = k$

where $\delta(x, F) = \min\{\delta(x, y): y \in F\}$

Proof: i) Necessity: if $x \in H(x', k) \cap \{0,1\}^n$ this imply that $\delta(N(x'), x', x) = k$. Moreover if $y \in F(x') \cap \{0,1\}^n$ thus $\delta(N(x'), x', y) = 0$ which imply that $y(N(x')) = x'(N(x'))$ where $x(J) = (x_j)_{j \in J}$. Hence, we have

$$\begin{aligned} \delta(x, y) &= \delta(N(x'), x, y) + \delta(N-N(x'), x, y) \\ &= \delta(N(x'), x, x') + \delta(N-N(x'), x, y) \\ &= k + \delta(N-N(x'), x, y) \geq k. \end{aligned}$$

Let $y' \in \{0,1\}^n$ such that $y'(N(x')) = x'(N(x'))$ and $y'(N-N(x')) = x(N-N(x'))$. Then we have $y' \in F(x')$ and $\delta(x, y') = k$. Hence, $\delta(x, F(x') \cap \{0,1\}^n) = \min\{\delta(x, y): y \in F(x') \cap \{0,1\}^n\} = k$.

ii) Sufficiency: Let $y' \in F(x') \cap \{0,1\}^n$ such that $\delta(x, y') = \delta(x, F(x') \cap \{0,1\}^n) = k$. To simplify the notion let $F' = F(x') \cap \{0,1\}^n = \{x \in \{0,1\}^n: x_j = x'_j \text{ for } j \in N(x')\}$. Hence, we have $\delta(N-N(x'), x, F') = 0$ which implies that $\delta(N-N(x'), x, y') = 0$. Moreover if $y' \in F(x')$ we have $y'(N(x')) = x'(N(x'))$. Thus $\delta(N(x'), x, x') = k$. This implies that $x \in H(x', k) \cap \{0,1\}^n$. which completes the proof of this proposition. \square

In the next proposition, we state relation between half-spaces associated to the canonical hyperplanes. Let $H(x', k)$ be the half-space associated with the canonical hyperplane $H(x', k)$ defined by

$$H(x', k) = \{x \in [0,1]^n: \delta(N(x'), x', x) \leq k\}$$

Proposition 4. Let x' and x'' be two arbitrary solutions. Then

$$H(x', k) \cup H(x'', k) \subseteq H((x' + x'')/2, k).$$

Proof: Immediate from the fact that $N(x') \subseteq N((x' + x'')/2)$ and $N(x'') \subseteq N((x' + x'')/2)$. \square

Proposition 5. $Co(H(x', k) \cap \{0,1\}^n) = H(x', k)$, where $Co(X)$ is the convex hull of the set X .

Proof: The inclusion $Co(H(x', k) \cap \{0,1\}^n) \subseteq H(x', k)$ is obvious for any solution x' and integer k . To prove the inclusion

$$H(x', k) \subseteq Co(H(x', k) \cap \{0,1\}^n), \quad (5.1)$$

let $y \in H(x', k)$ and observe that $\delta(N(x'), x', y) = \delta(N(x') \cap N(y), x', y) + \delta(N(x') \cap N^*(y), x', y) = k$. Now, we show by induction the second inclusion (4.3) on $p = \delta(N(x') \cap N^*(y), x', y)$. The statement is evident for $p = 0$. We assume that the statement is true for $\delta(N(x') \cap N^*(y), x', y) = p$. To show that it is also true for $\delta(N(x') \cap N^*(y), x', y) = p+1$, consider the subset $J \subseteq N(x') \cap N^*(y)$ such that

$$\delta(J, x', y) = \sum_{j \in J} \delta(j, x', y) = 1 \quad (5.2)$$

Thus we have $\delta(N(x'), x', y) = \delta(J, x', y) + \delta((N-J) \cap (N(x') \cap N^*(y)), x', y) = k$. For all $j \in J$, define the vector y^j such that

$$\begin{aligned} y^j(N-J) &= y(N-J) \\ y^j(J-\{j\}) &= x'(J-\{j\}) \text{ and} \\ y^j(\{j\}) &= 1 - x'_j. \end{aligned} \quad (5.3)$$

Now we show that

$$\sum_{j \in J} \delta(j, x', y) y^j = y. \quad (5.4)$$

From (5.3) and (5.2), for all $q \notin J$ we have

$$\begin{aligned} \sum_{j \in J} \delta(j, x', y) y_q^j &= \\ y_q \sum_{j \in J} \delta(j, x', y) y_q^j &= y_q. \end{aligned}$$

For all $q \in J$, from (5.3) and (5.2) and since $(1 - 2x'_q)^2 = 1$ and $x'_q(1 - x'_q) = 0$ for all $x'_q \in \{0,1\}$, we have

$$\begin{aligned} \sum_{j \in J} \delta(j, x', y) y_q^j &= \sum_{j \in J - \{q\}} \delta(j, x', y) y_q^j + \\ &\delta(q, x', y) y_q^q \\ &= \sum_{j \in J - \{q\}} \delta(j, x', y) x'_q + \\ &\delta(q, x', y) (1 - x'_q) \\ &= \sum_{j \in J} \delta(j, x', y) x'_q + \\ &\delta(q, x', y) (1 - 2x'_q) \\ &= x'_q \sum_{j \in J} \delta(j, x', y) + \\ &\delta(q, x', y) (1 - 2x'_q) \\ &= x'_q + \delta(q, x', y) (1 - 2x'_q) \\ &= x'_q + ((1 - 2x'_q) y_q + x'_q) (1 - 2x'_q) \\ &= (1 - 2x'_q)^2 y_q + 2x'_q (1 - x'_q) \\ &= y_q \end{aligned}$$

Hence y is on the convex hull of the vector y^j for $j \in J$ (see 5.4) and it is easy to see that

$$\delta(N(x') \cap N^*(y^j), x', y) = p \text{ for all } j \in J.$$

By applying the hypothesis of the induction, we conclude that each vector y^j is also on the convex hull of binary solutions in $H(x', k)$. This completes the proof of the second inclusion (5.1). The proposition then follows from the two inclusions. \square

Proposition 5 is related to Theorem 1 of Balas and Jeroslow.

Let x' denote an arbitrary solution and $c \in IN^n$ define the associated set

$$\begin{aligned} F(x', c) &= \{x \in [0,1]^n: c_j(x_j - x'_j) = \\ &0 \text{ for } j \in N(x') \} \end{aligned}$$

Let x, x' be two arbitrary binary solutions and let c be an integer vector ($c \in IN^n$). Define

$$\delta(c, x', x) = \sum_{j \in N} c_j (x_j (1 - x'_j) + x'_j (1 - x_j)) \quad \delta(c, x', x) \geq \lceil \delta(c, x', x'') \rceil \quad (7.1)$$

$$\delta(c, x', x) \leq \lfloor \delta(c, x', x'') \rfloor \quad (7.2)$$

Remark 4: $\delta(e, x', x) = \|x - x'\|_1 = \|x - x'\|_2$

$$B(c) = \{x \in [0,1]^n : c_j x_j (1 - x_j) = 0\}.$$

Remark 5: $\delta(c, x', x) = \delta(J', x', x)$ if $c_j = 1$ if $j \in J'$ otherwise $c_j = 0$.

Remark 6: $B(e) = \{0,1\}^n$ and $B(0) = [0,1]^n$
 $C(x) = \{c \in \mathbb{N}_+^n : c_j x_j (1 - x_j) = 0\}$.

Proposition 6. Let x' denote an arbitrary solution and $c \in C(x')$. Then the inequality

$$\delta(c, x', x) \geq 1 \quad (6.1)$$

eliminates the solutions in $F(x', c)$ as a feasible solution, but admits all other binary x vectors. The inequality

$$\delta(c, x', x) \leq ce - 1 \quad (6.2)$$

eliminates the solutions in $F(e - x', c)$ as a feasible solution, but admits all other binary x vectors.

Proof: Immediate from the proof on Proposition 2 and Corollary 2 by setting $J' = \{c \in N : c_j \neq 0\}$. □

We couple the target solution x' with the associated vector $c \in C(x')$ to yield the two problems

LP(x', c): Minimize $(\delta(c, x', x) : x \in X)$.

LP⁺(x', c): Maximize $(\delta(c, x', x) : x \in X)$.

An optimal solution to LP(x', c) (resp. LP⁺(x', c)), as a generalization of LP(x') (resp. LP⁺(x')), will likewise be denoted by x'' (resp. $x^{+''}$). Finally, we obtain the inequalities

4. STRONGER INEQUALITIES AND ADDITIONAL VALID INEQUALITIES FROM BASIC FEASIBLE LP SOLUTIONS

Our approach to generate inequalities that dominate those of (7) is also able to produce additional valid inequalities from related basic feasible solution to the LP problem LP(x', c), expanding the range of solution strategies for exploiting the use of target solutions. We refer specifically to the class of basic feasible solutions that may be called y -optimal solutions, which are dual feasible in the continuous variables y (including in y any continuous slack variables that may be added to the formulation), disregarding dual feasibility relative to the x variables. Such y -optimal solutions can be easily generated in the vicinity of an optimal LP solution by pivoting to bring one or more non-basic x variables into the basis, and then applying a restricted version of the primal simplex method that re-optimizes (if necessary) to establish dual feasibility relative only to the continuous variables, ignoring pivots that would bring x variables into the basis. By this means, instead of generating a single valid inequality from a given LP formulation such as LP(x', c), we can generate a collection of such inequalities from a series of basic feasible y -optimal solutions produced by a series of pivots to visit some number of such solutions in the vicinity of an optimal solution.

As a foundation for these results, we assume x'' (or more precisely, (x'', y'')) has been obtained as a y -optimal basic feasible solution to LP(x', c) by the bounded variable simplex method (see, e.g., Dantzig, 1963). By reference to the linear programming basis that produces x'' , which we will call the x'' basis, define $B = \{j \in N : x_j'' \text{ is basic}\}$ and $NB = \{j \in N : x_j'' \text{ is non-basic}\}$. We subdivide NB to identify the two subsets $NB(0) = \{j \in NB : x_j'' = 0\}$, $NB(1) = \{j \in NB : x_j'' = 1\}$. These sets have no necessary relation

to the sets $N(x'',0)$ and $N(x'',1)$, though in the case where x'' is an optimal basic solution⁵ to $LP(x',c)$, we would normally expect from the definition of c in relation to the target vector x' that there would be some overlap between $N(x'',0)$ and $N(x'',1)$ and similarly between $NB(1)$ and $N(1)$.

To simplify the notation, we find it convenient to give $\delta(c, x', x)$ an alternative representation.

$$\delta(c, x', x) = c'x + cx' \text{ with } c'_j = c_j(1 - 2x'_j), j \in N$$

The new inequality that dominates (6) results by taking account of the reduced costs derived from the x'' basis. Letting rc denote the reduced cost to an arbitrary y -optimal basic feasible solution x'' for $LP(x',c)$. Finally, to identify the new inequality, define the vector d' by

$$d' = \lceil c - rc \rceil$$

We then express the inequality as

$$\delta(d', x', x) \geq \lfloor \delta(d', x', x'') \rfloor \tag{8}$$

We first show that (8) is valid when generated from an arbitrary y -optimal basic feasible solution, and then demonstrate in addition that it dominates (7) in the case where (8) is a valid inequality (i.e., where (8) is derived from an optimal basic feasible solution). By our previously stated convention, it is understood that X (and (MIP:0-1)) may be modified by incorporating previously generated inequalities that exclude some binary solutions originally admitted as feasible.

Our results concerning (8) are based on identifying properties of basic solutions in reference to the problem

LP(x', d'): Minimize $(\delta(d', x', x): x \in X)$

Proposition 7. *The inequality (8) derived from an arbitrary y -optimal basic feasible solution x'' for $LP(x', c)$ is satisfied by all binary vectors*

$x \in X$, and excludes the solution $x = x''$ when $\delta(c, x', x'')$ is fractional.

Proof: *We first show that the basic solution x'' for $LP(x', c)$ is an optimal solution to $LP(x', d')$. Let rd denote the reduced cost for the objective function $\delta(d', x', x)$ for $LP(x', d')$ relative to the x'' basis. Assume $X = \{x: Ax \geq b, x \geq 0\}$ and let B be the basis associated to the basic solution x'' . From the definitions the reduced cost $rc = c' - c'_B(A^B)^{-1}A$, and of $d' = \lfloor c' - rc \rfloor$, it follows that*

$$d' = \lfloor c'_B(A^B)^{-1}A \rfloor \text{ and } d'_B = c'_B \tag{8.1}$$

thus the reduced costs rd is null; i.e.,

$$rd' = d' - d'_B(A^B)^{-1}A = \lfloor c'_B(A^B)^{-1}A \rfloor - d'_B(A^B)^{-1}A = \lfloor c'_B(A^B)^{-1}A \rfloor - c'_B(A^B)^{-1}A \leq 0.$$

This establishes the optimality of x'' for $LP(x', d')$. Since the d'_j coefficients are all integers, we therefore obtain the valid inequality $\delta(d', x', x) \geq \lceil \delta(d', x', x'') \rceil$.

The definition of d' yields

$$\delta(d', x', x'') = \delta(c, x', x'') + \delta(\lfloor -rc \rfloor, x', x'').$$

The $\delta(\lfloor -rc \rfloor, x', x'')$ value is integer, since $x'' \in B(\lfloor -rc \rfloor)$. Thus, $\delta(d', x', x'')$ is fractional if and only if $\delta(c', x', x'')$ is fractional, and we also have

$$\lceil \delta(d', x', x'') \rceil = \lceil \delta(c, x', x'') \rceil + \delta(\lfloor -rc \rfloor, x', x'').$$

The proposition then follows from the definitions of (7) and (8).

Proposition 7 has the following novel consequence.

Corollary 3. *The inequality (8) is independent of the c_j values for the non-basic x variables. In particular, for any y -feasible basic solution and specified values c_j for $j \in B$, the coefficients d'_j*

of d' are identical for every choice of the integer coefficients $c_j, j \in NB$.

$$\delta(\lfloor -rc \rfloor, x', x'') - \delta(\lfloor -rc \rfloor, x', x) = \lfloor -rc \rfloor x'' - \lfloor -rc \rfloor x \geq 0 \tag{8.3}$$

Proof: The Corollary follows from the arguments of the Proof of Proposition 7 (see 8.1) thus showing that these changes cancel out, to produce the same final d'_0 and d' after implementing the changes that existed previously.

Moreover we have

$$\delta(d', x', x'') = \delta(c, x', x'') + \delta(\lfloor -rc \rfloor, x', x''). \tag{8.4}$$

$$\delta(d', x', x) = \delta(c, x', x) + \delta(\lfloor -rc \rfloor, x', x). \tag{8.5}$$

In effect, since Corollary 1 applies to the situation where $c_j = 0$ for $j \in NB$, it also allows each d'_j coefficient for $j \in NB$ to be identified by reference to the quantity that results by multiplying the vector of optimal dual values by the corresponding column A^j of the matrix A defining the constraints of (MIP), excluding rows of A corresponding to the inequalities $1 \geq x_j \geq 0$. (We continue to assume this matrix is enlarged by reference to additional inequalities such as (7) or (8) that may currently be included in defining $x \in X$.)

Hence by substituting (8.4) and (8.5) in the inequality (8) we obtain

$$\delta(c, x', x) + \delta(\lfloor -rc \rfloor, x', x) \geq \lceil \delta(c, x', x'') + \delta(\lfloor -rc \rfloor, x', x'') \rceil = \delta(c, x', x'') + \delta(\lfloor -rc \rfloor, x', x'').$$

Thus by using (8.3) we obtain (7). Consequently, this establishes that (8) implies (7).

Now we establish the result that (8) is at least as strong as (7).

Corollary 4. If the basic solution x'' for $LP(x', c)$ is optimal then

$$\delta(d', x', x) - \lceil \delta(d', x', x'') \rceil = \delta(c, x', x) - \lceil \delta(c, x', x'') \rceil.$$

Proposition 8. If the basic solution x'' for $LP(x', c)$ is optimal, and thus yields a valid inequality (7), then the inequality (8) dominates (7).

Proof: If the basic solution x'' for $LP(x', c)$ is optimal then $\delta(\lfloor -rc \rfloor, x', x'')$ is integer so $\delta(\lfloor -rc \rfloor, x', x'') = \lceil \delta(\lfloor -rc \rfloor, x', x'') \rceil$ which implies that $\delta(d', x', x) - \lceil \delta(d', x', x'') \rceil = \delta(c, x', x) - \lceil \delta(c, x', x'') \rceil$.

Proof: We use the fact that x'' is optimal for $LP(x', d')$ as established by Proposition 7. When x'' is optimal for $LP(x', c)$ from the optimal condition of the corresponding dual we have

As in the use of the inequality (7), if a basic solution x'' that generates (8) is a binary vector that differs from x' , then we can also test x'' for feasibility in (MIP:0-1) and then redefine $x' = x''$, to additionally append the constraint (1.1) for this new x' .

$$rc x'' \leq rc x \text{ for all } x \geq 0$$

Thus we have

The combined arguments of the proofs of Propositions 7 and 8 lead to a still stronger conclusion. Consider a linear program $LP(x', h')$ given by

$$\lfloor -rc \rfloor x'' \geq \lfloor -rc \rfloor x \text{ for all } x \geq 0 \tag{8.2}$$

Since

LP(x', h'): Minimize $(\delta(h', x', x): x \in X)$. where the coefficients $h'_j = d'_j$ (and hence $= c_j$) for $j \in B$ and, as before, B is defined relative to a given y -optimal basic feasible solution x'' . Subject to this condition, the only restriction

$$\delta(\lfloor -rc \rfloor, x', x'') = \lfloor -rc \rfloor x'' + \lfloor -rc \rfloor x'$$

$$\delta(\lfloor -rc \rfloor, x', x) = \lfloor -rc \rfloor x + \lfloor -rc \rfloor x'$$

This with (8.2) implies that

on the h'_j coefficients for $j \in NB$ is that they be integers. Then we can state the following result.

Corollary 5. *The x'' basis is an optimal LP basis for $LP(x', h')$ if and only if*

$$h'_j \geq d'_j \text{ for } j \in NB$$

and the inequality (8) dominates the corresponding inequality derived by reference to $LP(x', h')$.

Proof: *Immediate from the proofs on Propositions 7 and 8.*

The importance of Corollary 5 is the demonstration that (8) is the strongest possible valid inequality from those that can be generated by reference to a given y -optimal basic solution x'' and an objective function that shares the same coefficients for the basic variables.

It is to be noted that if (MIP:0-1) contains an integer valued slack variable s_i upon converting the associated inequality $A_i x + D_i y \geq b_i$ of the system $Ax + Dy \geq b$ into an equation – hence if A_i and b_i consist only of integers and D_i is the 0 vector – then s_i may be treated as one of the components of the vector x in deriving (8), and this inclusion serves to sharpen the resulting inequality. In the special case where all slack variables have this form, i.e., where (MIP:0-1) is a pure integer problem having no continuous variables and all data are integers, then it can be shown that the inclusion of the slack variables within x yields an instance of (8) that is equivalent to a fractional Gomory cut, and a stronger inequality can be derived by means of the foundation-penalty cuts of Glover and Sherali (2003). Consequently, the primary relevance of (8) comes from the fact that it applies to mixed integer as well as pure integer problems, and more particularly provides a useful means for enhancing target objective strategies for these problems. As an instance of this, we now examine methods that take advantage of (8) in additional ways by

extension of ideas proposed with parametric tabu search.

5. GENERATING TARGET OBJECTIVES AND SOLUTIONS BY EXPLOITING PROXIMITY

We now examine the issue of creating the target solution x' and associated target objective $\delta(c, x', x)$ that underlies the inequalities of the preceding sections. This is a key determinant of the effectiveness of targeting strategies, since it determines how quickly and effectively such a strategy can lead to new integer feasible solutions.

In this section, we propose a relatively simple approach for generating the vector c of the target objective by exploiting proximity. The proximity procedure for generating target solutions x' and associated target objectives $\delta(c, x', x)$ begins by solving the initial problem (LP), and then solves a succession of problems $LP(x', c)$ by progressively modifying x' and c . Beginning from the linear programming solution x'' to (LP) (and subsequently to $LP(x', c)$), the new target solution x' is derived from x'' simply by setting $x'_j = \langle x''_j \rangle, j \in N$, where $\langle v \rangle$ denotes the nearest integer neighbour of v . (The value $\langle v \rangle$ can be either 0 or 1, by employing an arbitrary tie-breaking rule.)

Since the resulting vector x' of nearest integer neighbors is unlikely to be feasible for (MIP:0-1), the critical element is to generate the target objective $\delta(c, x', x)$ so that the solutions x'' to successively generated problems $LP(x', c)$ will become progressively closer to satisfying integer feasibility.

If one or more integer feasible solutions is obtained during this approach, each such solution qualifies as a new best solution x^* , due to the incorporation of the objective function constraint $x_o \leq U_o < x_o^*$.

The criterion of the proximity procedure that selects the target solution x' as a nearest integer neighbor of x'' is evidently myopic. Consequently, the procedure is intended to be executed for only a limited number of iterations.

However, the possibility exists that for some problems the target objectives of this approach may quickly lead to new integer solutions without invoking more advanced rules. To accommodate this eventuality, we include the option of allowing the procedure to continue its execution as long as it finds progressively improved solutions.

The proximity procedure is based on the principle that some variables x_j should be more strongly induced to receive their nearest neighbor target values x'_j than other variables. In the absence of other information, we may tentatively suppose that a variable whose LP solution value x''_j is already an integer or is close to being an integer is more likely to receive that integer value in a feasible integer solution. Consequently, we are motivated to choose a target objective $\delta(c, x', x)$ that will more strongly encourage such a variable to receive its associated value x'_j . However, the relevance of being close to an integer value needs to be considered from more than one perspective.

5.1 Batwing Function for Proximity

The targeting of $x_j = x'_j$ for variables whose values x''_j already equal or almost equal x'_j does not exert a great deal of influence on the solution of the new LP(x', c), in the sense that such a targeting does not drive this solution to differ substantially from the solution to the previous LP(x', c). A more influential targeting occurs by emphasizing the variables x_j whose x''_j values are more “highly fractional,” and hence which differ from their integer neighbours x'_j by a greater amount. There are evidently trade-offs to be considered in the pursuit of influence, since a variable whose x''_j value lies close to .5, and hence whose integer target may be more influential, has the deficiency that the likelihood of this integer target being the “right” target is less certain. A compromise targeting criterion is therefore to give greater emphasis to driving x_j to an integer value if x''_j lies “moderately” (but not exceedingly) close to an integer value. Such a criterion affords an improved chance that the targeted value will

be appropriate, without abandoning the quest to identify targets that exert a useful degree of influence. Consequently, we select values λ_0 and $\lambda_1 = 1 - \lambda_0$ that lie moderately (but not exceedingly) close to 0 and 1, such as $\lambda_0 = 1/5$ and $\lambda_1 = 4/5$, or $\lambda_0 = 1/4$ and $\lambda_1 = 3/4$, and generate c_j coefficients that give greater emphasis to driving variables to 0 and 1 whose x''_j values lie close to λ_0 and λ_1 .

The following rule creates a target objective $\delta(c, x', x)$ based on this compromise criterion, arbitrarily choosing a range of 1 to 21 for the coefficient c_j . (From the standpoint of solving the problem LP(x', c), this range is equivalent to any other range over positive values from v to $21v$, except for the necessity to round the c_j coefficients to integers.)

Proximity Rule for Generating c_j :

Choose λ_0 from the range $.1 \leq \lambda_0 \leq .4$, and let $\lambda_1 = 1 - \lambda_0$.

If $x'_j = 0$ (hence $x''_j \leq .5$) then
 If $x''_j \leq \lambda_0$, set $c_j = 1 + 20x''_j / \lambda_0$
 Else set $c_j = 1 + 20(.5 - x''_j) / (.5 - \lambda_0)$
 Else if $x'_j = 1$ (hence $x''_j \geq .5$) then
 If $x''_j \leq \lambda_1$, set $c_j = 1 + 20(x''_j - .5) / (\lambda_1 - .5)$
 Else set $c_j = 1 + 20(1 - x''_j) / (1 - \lambda_1)$
 End if

Finally, replace the specified value of c_j by its nearest integer neighbour $\langle c_j \rangle$

Remark 7: $c_j = 1$ if $x''_j = x'_j$.

The values of c_j coefficients produced by the preceding rule describe what may be called a *batwing* function—a piecewise linear function resembling the wings of a bat, with shoulders at $x''_j = .5$, wing tips at $x''_j = 0$ and $x''_j = 1$, and the angular joints of the wings at $x''_j = \lambda_0$ and $x''_j = \lambda_1$. Over the x''_j domain from the left wing tip at 0 to the first joint at λ_0 , the function ranges from 1 to 21, and then from this joint to the left shoulder at .5 the function ranges from 21 back

to 1. Similarly, from right shoulder, also at .5, to the second joint at λ_1 , the function ranges from 1 to 21, and then from this joint to the right wing tip at 1 the function ranges likewise from 21 to 1. (The coefficient c_j takes the negative of these absolute values from the right shoulder to the right wing tip.).

In general, if we let *Tip*, *Joint* and *Shoulder* denote the c_j values to be assigned at these junctures (where typically *Joint* > *Tip*, *Shoulder*), then the generic form of a batwing function results by replacing the four successive c_j values in the preceding method by

$$\begin{aligned} c_j &= \text{Tip} + (\text{Joint} - \text{Tip})x_j'' / \lambda_0, \\ c_j &= \text{Shoulder} + (\text{Joint} - \text{Shoulder})(.5 - x_j'') / (.5 - \lambda_0), \\ c_j &= \text{Shoulder} + (\text{Joint} - \text{Shoulder})(x_j'' - .5) / (\lambda_1 - .5) \\ c_j &= \text{Tip} + (\text{Joint} - \text{Tip})(1 - x_j'') / (1 - \lambda_1) \end{aligned}$$

The values of c_j coefficients called a *batwing* function can also be expressed as follows:

$$\begin{aligned} c_j &= \text{Tip} + (\text{Joint} - \text{Tip}) \delta(j, x', x'') / \lambda_0, \text{ if } x_j'' \notin]\lambda_0, 1 - \lambda_0] \\ c_j &= \text{Shoulder} + (\text{Joint} - \text{Shoulder})(.5 - \delta(j, x', x'')) / (.5 - \lambda_0), \text{ otherwise} \end{aligned}$$

The image of such a function more nearly resembles a bat in flight as the value of *Tip* is increased in relation to the value of *Shoulder*, and more nearly resembles a bat at rest in the opposite case. The function can be turned into a piecewise convex function that more strongly targets the values λ_0 and λ_1 by raising the absolute value of c_j to a power $p > 1$ (affixing a negative sign to yield c_j over the range from the right shoulder to the right wing tip). Such a function (e.g., a quadratic function) more strongly resembles a bat wing than the linear function.⁶

5.2 Design of the Proximity Procedure

We allow the proximity procedure that incorporates the foregoing rule for generating c_j the option of choosing a single fixed λ_0 value, or of choosing different values from the specified interval to generate a greater variety of outcomes. A subinterval for λ_0 centred around .2 or .25 is anticipated to lead to the best outcomes, but it can be useful to periodically choose values outside this range for diversification purposes.

We employ a stopping criterion for the proximity procedure that limits the total number of iterations or the number of iterations since finding the last feasible integer solution. In each instance where a feasible integer solution is obtained, the method re-solves the problem (LP), which is updated to incorporate both the objective function constraint $x_o \leq U_o < x_o^*$ and inequalities such as (8) that are generated in the course of solving various problems $LP(x', c)$. The instruction "Update the Problem Inequalities" is included within the proximity procedure to refer to this process of adding inequalities to $LP(x', c)$ and (LP), and to the associated process of dropping inequalities by criteria indicated in Section 2.

Proximity Procedure

1. Solve (LP). (If the solution x'' to the first instance of (LP) is integer feasible, the method stops with an optimal solution for (MIP:0-1).)
2. Construct the target solution x' derived from x'' by setting $x_j' = \alpha_j x_j''$, for $j \in N$. Apply the Rule for Generating c_j to each $j \in N$, to produce the vector c .
3. Solve $LP(x', c)$, yielding the solution x'' . Update the Problem Inequalities.
4. If x'' is integer feasible: update the best solution $(x^*, y^*) = (x'' , y'')$, update $U_o < x_o^*$, and return to Step 1. Otherwise, return to Step 2.

A preferred variant of the proximity procedure does not change all the components of

c each time a new target objective is produced, but changes only a subset consisting of k of these components, for a value k somewhat smaller than n . For example, a reasonable default value for k is given by $k = 5$. Alternatively, the procedure may begin with $k = n$ and gradually reduce k to its default value.

This variant results by the following modification. Let c° identify the form of c produced by the Proximity Rule for Generating c_j , as applied in Step 2 of the Proximity Procedure. Re-index the x_j variables so that $c_1^\circ \geq c_2^\circ \geq \dots \geq c_n^\circ$, and let $J(k) = \{1, \dots, k\}$, thus identifying the variables x_j , $j \in J(k)$, as those having the k largest c_j° values. Then proximity procedure is amended by setting $c = 0$ in Step 1 and then setting $c_j = c_j^\circ$ for $j \in J(k)$ in Step 2, without modifying the c_j values for $j \in N - J(k)$. Relevant issues for research involve the determination of whether it is better to begin with k restricted or to gradually reduce it throughout the search, or to allow it to oscillate around a preferred value. Different classes of problems will undoubtedly afford different answers to such questions, and may be susceptible to exploitation by different forms of the batwing function (allowing different magnitudes for the Tip, Joint and Shoulder, and possibly allowing the location of the shoulders to be different than the .5 midpoint, with the locations of the joints likewise asymmetric).

6. CONCLUSION

Branch-and-bound (*B&B*) and branch-and-cut (*B&C*) methods have long been considered the methods of choice for solving mixed integer programming problems. This orientation has resulted in eliciting contributions to these classical methods from many researchers, and has led to successive improvements in these methods extending over a period of several decades. In recent years, these efforts to create improved *B&B* and *B&C* solution approaches have intensified and have produced significant benefits, as evidenced by the existence of MIP

procedures that are appreciably more effective than their predecessors.

It remains true, however, that many MIP problems resist solution by the best current *B&B* and *B&C* methods. It is not uncommon to encounter problems that confound the leading commercial solvers, resulting in situations where these solvers are unable to find even moderately good feasible solutions after hours, days, or weeks of computational effort. As a consequence, metaheuristic methods have attracted attention as possible alternatives or supplements to the more classical approaches. Yet to date, the amount of effort devoted to developing good metaheuristics for MIP problems is almost negligible compared to the effort being devoted to developing refined versions of the classical methods.

The view adopted in this paper is that metaheuristic approaches can benefit from a change of perspective in order to perform at their best in the MIP setting. Drawing on lessons learned from applying classical methods, we anticipate that metaheuristics can likewise profit from generating inequalities to supplement their basic functions. However, we propose that these inequalities be used in ways not employed in classical MIP methods, and indicate two principal avenues for doing this: first by generating the inequalities in reference to strategically created target solutions and target objectives, as in the current Part I, and second by embedding these inequalities in special intensification and diversification processes, as described in Part II.

ACKNOWLEDGMENT

The present research work has been supported by International Campus on Safety and Intermodality in Transportation, the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the Ministry of Higher Education and Research, the National Center for Scientific Research, and by a "Chaire d'excellence" from "Pays de la Loire" Region (France). A restricted

(preliminary) version of this work appeared in Glover (2008).

REFERENCES

- Balas, E., & Jeroslow, R. (1972). Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23(1), 60–69. doi:10.1137/0123007
- Dantzig, G. (1963). *Linear programming and extensions*. NJ: Princeton University Press.
- Fischetti, M., Glover, F., & Lodi, A. (2005). Feasibility pump. *Mathematical Programming - Series A*, 104, 91–104. doi:10.1007/s10107-004-0570-3
- Glover, F. (1978). Parametric branch and bound. *OMEGA. The International Journal of Management Science*, 6(2), 145–152.
- Glover, F. (2005). Adaptive memory projection methods for integer programming. In C. Rego and B. Alidaee (Eds.), *Metaheuristic optimization via memory and evolution: Tabu search and scatter search* (pp. 425–440). Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Glover, F. (2006a). Parametric tabu search for mixed integer programs. *Computers & Operations Research*, 33(9), 2449–2494. doi:10.1016/j.cor.2005.07.009
- Glover, F. (2006b). *Satisfiability data mining for binary data classification problems*. Boulder: University of Colorado.
- Glover, F. (2007). Infeasible/feasible search trajectories and directional rounding in integer programming. *Journal of Heuristics*, 13(6), 505–542. doi:10.1007/s10732-007-9029-z
- Glover, F. (2008). Inequalities and target objectives for metaheuristic search – part I: Mixed binary optimization. In P. Siarry and Z. Michalewicz (Eds.), *Advances in metaheuristics for hard optimization* (pp. 439–474). New York: Springer.
- Glover, F., & Greenberg, H. (1989). New approaches for heuristic search: A bilateral linkage with artificial intelligence. *European Journal of Operational Research*, 39(2), 119–130. doi:10.1016/0377-2217(89)90185-9
- Glover, F., & Hanafi, S. (2002). Tabu search and finite convergence. *Discrete Applied Mathematics*, 119, 3–36. doi:10.1016/S0166-218X(01)00263-3
- Glover, F., & Laguna, M. (1997). *Tabu search*. Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Glover, F., & Sherali, H. D. (2003). Foundation-penalty cuts for mixed-integer programs. *Operations Research Letters*, 31, 245–253. doi:10.1016/S0167-6377(03)00014-2
- Guignard, M., & Spielberg, K. (2003). *Double contraction, double probing, short starts and bb-probing cuts for mixed (0, 1) programming*. Philadelphia: Wharton School of the University of Pennsylvania.
- Hanafi, S., & Wilbaut, C. (2006). Improved convergent heuristics for the 0-1 multidimensional knapsack problem. *Annals of Operations Research*. doi:doi:10.1007/s10479-009-0546-z
- Hvattum, L. M., Lokketangen, A., & Glover, F. (2004). Adaptive memory search for boolean optimization problems. *Discrete Applied Mathematics*, 142, 99–109. doi:10.1016/j.dam.2003.06.006
- Nowicki, E., & Smutnicki, C. (1996). A fast taboo search algorithm for the job shop problem. *Management Science*, 42(6), 797–813. doi:10.1287/mnsc.42.6.797
- Soyster, A. L., Lev, B., & Slivka, W. (1978). Zero-one programming with many variables and few constraints. *European Journal of Operational Research*, 2(3), 195–201. doi:10.1016/0377-2217(78)90093-0
- Spielberg, K., & Guignard, M. (2000). *A sequential (quasi) hot start method for bb (0,1) mixed integer programming*. Paper presented at the Mathematical Programming Symposium, Atlanta.
- Ursulenko, A. (2006). *Notes on the global equilibrium search*. College Station, TX: Texas A&M University.
- Wilibaut, C., & Hanafi, S. (2009). New convergent heuristics for 0-1 mixed integer programming. *European Journal of Operational Research*, 195, 62–74. doi:10.1016/j.ejor.2008.01.044

ENDNOTES

- ¹ The vector c' depends on x' . As will be seen, we define several different linear programs that are treated as described here in reference to the problem LP(x' , c').

- ² An effective way to enforce $v_o = v_o''$ is to fix all non-basic variables having non-zero reduced costs to compel these variables to receive their optimal first stage values throughout the second stage. This can be implemented by masking the columns for these variables in the optimal first stage basis, and then to continue the second stage from this starting basis while ignoring the masked variables and their columns. (The masked non-basic variables may incorporate components of both x and y , and will generally include slack variables for some of the inequalities embodied in $Ax + Dy \geq b$.) The resulting residual problem for the second stage can be significantly smaller than the first stage problem, allowing the problem for the second stage to be solved very efficiently.
- ³ In some problem settings, the inclusion of the secondary objective x_o in $v_{oo} = Mv_o + x_o$ is unimportant, and in these cases our notation is accurate in referring to the explicit minimization of $v_o = c'x$.
- ⁴ This strategy is utilized in the parametric branch and bound approach of Glover (1978) and in the feasibility pump approach of Fischetti, Glover and Lodi (2005).
- ⁵ We continue to apply the convention of referring to just the x -component x'' of a solution (x'', y'') , understanding the y component to be implicit.
- ⁶ Calibration to determine a batwing structure, either piecewise linear or nonlinear, that proves more effective than other alternatives within Phase 1 would provide an interesting study.

Fred Glover is the chief technology officer in charge of algorithmic design and strategic planning initiatives for OptTek Systems, Inc., heading the development of commercial computer software systems currently serving more than 80,000 users in the United States and abroad. He also holds the title of distinguished professor at the University of Colorado, Boulder, where he is affiliated with the Leeds School of Business and the Department of Electrical and Computer Engineering. Dr. Glover is widely known for his work in the applications of computer decision support systems, including industrial planning, financial analysis, systems design, energy and natural resources planning, logistics, transportation and large-scale allocation models. He has authored or co-authored more than 370 published articles and eight books in the fields of mathematical optimization, computer science and artificial intelligence. He is also the originator of Tabu Search (Adaptive Memory Programming), an optimization search methodology of which more than 200,000 Web pages can be found with a simple Google search. Dr. Glover is the recipient of the highest honor of the Institute of Operations Research and Management Science, the von Neumann Theory Prize, and is an elected member of the National Academy of Engineering. He has also received numerous other awards and honorary fellowships, including those from the American Association for the Advancement of Science (AAAS), the NATO Division of Scientific Affairs, the Institute of Operations Research and Management Science (INFORMS), the Decision Sciences Institute (DSI), the U.S. Defense Communications Agency (DCA), the Energy Research Institute (ERI), the American Assembly of Collegiate Schools of Business (AACSB), Alpha Iota Delta, the Institute of Cybernetics of the Ukrainian Academy of Science, and the Miller Institute for Basic Research in Science.

Said Hanafi holds a full professor position in computing science at Institute of Techniques and Sciences, University of Valenciennes and is currently in charge of the team Operations Research and Decision Support. His research lies in the design of effective heuristic and metaheuristic algorithms for solving large-scale combinatorial search problems. He is interested in theoretical as well as algorithmic modelling and application aspects of integer programming and combinatorial optimisation and has published over 30 articles on the topic. His current interests revolve around the integration of tools from hybrid methods mixing exact and heuristics for solving hard problems.