PARAMETRIC TABU SEARCH FOR 0-1 MIXED INTEGER PROGRAMMING: A COMPUTATIONAL STUDY

Vinícius Amaral Armentano
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas
Av. Albert Einstein, 400, CEP 13083-952, Campinas, SP, Brasil
vinicius@densis.fee.unicamp.br

Luís Henrique Sacchi
Escola de Engenharia de Piracicaba
Av. Monsenhor Matinho Salgot, 560, CEP 13414-049, Piracicaba, SP, Brasil
lhsacci@gmail.com

RESUMO

Neste trabalho é apresentado um estudo computacional da busca tabu paramétrica para resolver problemas de programação inteira mista (PIM) com variáveis binárias. Trata-se de uma heurística genérica para problemas PIM gerais que resolve uma série de problemas de programação linear ao incorporar desigualdades de ramificação de variáveis inteiras como termos ponderados na função objetivo. Novas estratégias são propostas para encontrar soluções de alta qualidade e extensivos testes computacionais são realizados em instâncias da literatura.

PALAVRAS CHAVE. Programação inteira mista 0-1. Ramificação penalizada. Busca tabu. Metaheuristicas

ABSTRACT

We present a computational study of parametric tabu search for solving 0-1 mixed integer programming (MIP) problems, a generic heuristic for general MIP problems. This approach solves a series of linear programming problems by incorporating branching inequalities as weighted terms in the objective function. New strategies are proposed for uncovering feasible and high-quality solutions and extensive computational tests are performed on instances from the literature.

KEYWORDS. Mixed integer programming, Penalized branching, Tabu search, Metaheuristics

1. Introduction

The new versions of MIP commercial solvers allow the user to have a greater control on parameter settings that yield better performance for solving a specific model. However, in some complex cases, a general-purpose solver may not be an adequate choice, and one tends to develop a heuristic, thus loosing the advantage of working in a generic and well-explored framework. Recently, generic heuristics for solving MIP problems have been proposed in the literature that interact with a commercial MIP-solver in order to quickly find a high quality solution or the first feasible solution (see, for example, Achterberg and Berthold (2007), Danna et al. (2005), Fischetti and Lodi (2003), Fischetti et al. (2005)). This approach allows a promising integrated way for obtaining multiple high quality solutions in a reasonable time.

In this paper, we present an implementation of the core version of parametric tabu search, a generic heuristic for solving MIP problems proposed by Glover (2006) that solves a series of linear programming problems incorporating branching inequalities as weighted terms in the objective function. The approach extends and modifies a parametric branch-and-bound proposed by Glover (1978), by replacing its tree search memory by the adaptive memory framework of tabu search that provides greater flexibility and facilitates the use of strategies outside the scope of tree search. Computational experiments are conducted on instances from the literature and results are compared with those obtained by the objective feasibility pump suggested by Achterberg and Berthold (2007).

The 0-1 mixed integer program is represented as

$$\begin{aligned} (MIP) \quad & Minimize \quad x_0 = cx + dy \\ & subject\ to \quad (x, y) \in W \\ & x \in X \end{aligned}$$

where

$$W = \{(x, y) : Ax + Dy \geq b, e \geq x \geq 0\},$$

$$X = \{e \geq x \geq 0 \ \ and \ \ x \ \ integer\},$$

A and D are matrices of dimensions $(m \times n)$ and $(m \times p)$, respectively, e denotes a vector with all components equal to 1, x is the vector of binary variables and y represents the vector of continuous variables. Assume that the inequalities $Ax + Dy \geq b$ include an objective function constraint $cx + dy \leq x_0^* - \varepsilon$, such that x_0^* is the currently best known solution value to (MIP) and ε is a small positive number.

The linear programming (LP) relaxation of (MIP), which contains only the restriction $(x, y) \in W$, will be denoted by (LP). A vector (x, y) is said to be LP feasible if it is feasible for (LP), and a vector x is integer feasible if the components of x are integers, and hence a (MIP) feasible solution is one that is LP feasible and integer feasible.

2. Principles of Parametric Tabu Search

Closely following Glover (2006), we now describe the principles of the search. Let N^+ and N^- denote selected subsets of $N = \{1, 2, \Lambda\ , n\}$, the index set for x. Consider the set $N' = N^+ \cup N^-$, and let $x' \in X$ be a trial solution such that its components are $x'_j, j \in N'$, and the remaining components $j \in N - N'$ are disregarded.

As in parametric branch-and-bound, the parametric tabu search attempts to impose the following conditions:

$$x_j \geq x'_j, \quad j \in N^+ \quad (if \;\; x'_j = 1) \quad (UP),$$

$$x_j \leq x'_j, \quad j \in N^- \quad (if \;\; x'_j = 0) \quad (DN).$$

The conditions (UP) and (DN) represent *goal conditions* and x'_j is called the *goal value*. Such conditions are not enforced directly as in branch-and-bound but rather indirectly by including them in the objective function of the problem (LP). The resulting linear penalized linear programming, in which M_j denotes a positive weight is given by

$$(LP') \quad Minimize \quad u_0 = cx + dy + \sum_{j \in N^-} M_j x_j + \sum_{j \in N^+} M_j (1 - x_j)$$

$$subject\ to \quad (x, y) \in W.$$

The problem (LP') is said to target the conditions (UP) and (DN). A two-phase approach is used to solve (LP'), by first creating a primary objective that disregards the component $cx + dy$. In the optimal solution of this phase, the non-basic variables are fixed at their current assigned binary values. In the second phase, the component $cx + dy$ is minimized over the residual constraints.

In the first phase we used an approach to the weight M_j which places more emphasis on recent goals and then it decays exponentially according to the number of iterations, which is expressed as

$$M_j = [(1+r)^{NIter - (Iter - GIter_j)}], \quad if \quad NIter - (Iter - GIter_j) \geq 0; \quad 1\ otherwise,$$

where $GIter_j$ denotes the iteration where a goal for the variable was established, *Iter* the current iteration, *NIter* a parameter that indicates the number of iterations that this memory lasts, r a parameter that defines the magnitude of the weight M_j during *NIter*, and the operator $[\cdot]$ rounds a real number to its nearest integer number.

The parametric tabu search method starts with an instance of (LP') associated with the original linear programming relaxation, when N' is empty. An optimal solution of an instance of (LP') is represented by (x'', y'') and as we are interested in the optimal values of the binary vector x'', we refer to x'' as the solution of (LP'), with the understanding that y'' is implicit. The parametric TS method proceeds by using information from the solution to (LP') and an associated new instance (LP'), which corresponds to the next iteration.

2.1. (LP') Transitions

The transition from one instance of (LP') to another is based on rules that define a new goal value x'_j as one of the values $\lfloor x''_j \rfloor$ and $\lceil x''_j \rceil$. There are three types of transitions (T-UP), (T-DN), and (T-FREE), expressed as follows:

(i) Set $x'_j := \lfloor x''_j \rfloor + 1$ and add j to N^+ (to target) $x_j \geq x'_j$ (T-UP).

(ii) Set $x'_j := \lceil x''_j \rceil - 1$ and add j to N^- (to target) $x_j \leq x'_j$ (T-DN).

(iii) Remove j from N' (to release x_j from (UP) and (DN)) (T-FREE).

The execution of these transitions for a suitable set of values depends on two types of conditions, called *goal infeasibility* and *integer infeasibility*, describe next.

2.2. Goal infeasibility

An optimal solution $x = x''$ to (LP$^{'}$) is said to be goal infeasible if it violates a current goal condition (UP) or (DN), i.e.,

(i) for some $j \in N^{+}, x''_j < x'_j$ (V-UP)

(ii) for some $j \in N^{-}, x''_j > x'_j$ (V-DN).

A variable associated with a violation (V-UP) or (V-DN) is called a *goal infeasible variable*, and let $G = \{ j \in N^{'} : x_j \textit{ is goal infeasible} \}$. The primary goal response to such an infeasibility consists of defining new goals in the opposite direction for a selected subset $G_p \subseteq G$ such that $\{ G_p = j \in N^{'} : \textit{primary response } (R-DN) \textit{ or } (R-UP) \textit{ is executed} \}$, i.e.,

(i) If $x''_j < x'_j$, $j \in N^{+}$, transfer j from N^{+} to N^{-} and set $x'_j = \lceil x''_j \rceil - 1$ (R-DN).

(ii) If $x''_j > x'_j$, $j \in N^{-}$, transfer j from N^{-} to N^{+} and set $x'_j = \lfloor x''_j \rfloor + 1$ (R-UP).

In addition to the primary responses, we define a secondary goal response that consists of freeing a goal infeasible variable that belongs to set G. Correspondingly, a selected subset $G_s \subseteq G$ is such that $G_s = \{ j \in G : \textit{response } (R-FREE) \textit{ is executed, i.e, remove } j \textit{ from } N^{'} \}$. A measure called *goal resistance* GR_j (UP) or GR_j (DN) of each variable $x_j, j \in G$ represents the amount of violation (V-UP) or (V-DN) resists the imposition of the associated goal condition (UP) or (DN). Since a higher resistance value cause a higher impact value on the objective function of (LP$^{'}$), we then choose the g_p variables in G with largest goal resistance and the g_s variables in G_s are those with largest goal resistance over $G - G_p$. In the presence of goal infeasibility we always choose $g_p \geq 1$, but we may have $g_s = 0$. Goal resistance measures are discussed in section 4.

2.3. Potentially goal infeasibility

A variable x_j is called *potentially goal infeasible* if it is goal feasible, but a limited change in M_j causes x_j to become goal infeasible. In order to deal with such variables, a measure of goal resistance related to the decrease in M_j is given by $GR^0_j = -RC_j$, where RC_j is the LP reduced cost for the variable x_j associated with an optimal solution to (LP$^{'}$). Goal infeasible variables are considered more important than potentially goal infeasible variables, and the goal resistance values GR_j of the first variables are larger than the goal resistance GR^0_j of the latter. Potentially goal infeasible are identified by sorting all goal feasible variables in decreasing value of GR^0_j and then selecting the first T^0 variables to be admitted as potentially goal infeasible. The primary and secondary goal responses for potentially goal infeasible variables are similar to those applied to variables that are goal infeasible, i.e., if a variable belongs to G_p, its goal is inverted, and if it belongs to G_p, it is freed from being goal infeasible.

2.4. Integer infeasibility

Let $F = \{ j \in N : x_j = x''_j \textit{ is fractional} \}$. A variable x_j is called unrestricted fractional variable if is fractional but does not have a goal. Such variables are defined by the set $D = F - G$. There is no specific primary transition for the case of integer infeasibility, and we may select a preferred transition (T-UP) or (T-DN). The relative preference for an unrestricted variable x_j is based on a choice preference measure CP_j that depends on the up penalty IP_j(UP) and the down penalty IP_j(DN), also discussed in section 4. For $D_0 \subseteq D$, the $d_0 = |D_0|$ variables with largest values of CP_j are selected to have goals associated with the following responses: if IP_j(DN) ≤ IP_j(UP), $j \in D_0$, execute (T-DN), otherwise execute (T-UP).

3. Tabu Search

A tabu restriction is associated to an (R-DN) or (R-UP) response for a given variable, by forbidding the response from being executed, if the opposing response was executed within the most recent tabu tenure iterations. Let $TabuTenure_j$(UP) and $TabuTenure_j$(DN) denote the tabu tenure of a variable x_j if the restriction was activated by an (R-DN) or (R-UP), respectively. Let α denote the condition UP or DN and β denote the opposite condition. When the reaction (R-α) is triggered, the reaction (R-β) becomes tabu for a number of iterations that is selected from the interval [$TabuTenure_j\alpha$, MaxTenure] with uniform distribution. In our implementation, the duration of $TabuTenure_j\alpha$ ($TabuTenure_j\beta$) starts from a minimum value and is increased by a factor that is proportional to the number of binary variables and has an upper limit. After a given number of iterations, and each time the best solution is updated, the tabu tenure of all variables are set to a minimum value.

The aspiration criterion in tabu search allows a tabu response to be released from a tabu restriction if the response has special merit. In the present context, we consider the aspiration by resistance, based on the greatest resistance a specific response has generated in the past. Let $Aspire_j$(DN) and $Aspire_j$(UP) denote the largest goal resistance GR_j(DN) and GR_j(UP) that occurred for x_j on any iteration in which this variable was chosen to execute an (R-UP) or (R-DN). The tabu restriction for an (R-UP) response is disregarded if GR_j (DN) > $Aspire_j$(UP) Analogously, the tabu restriction for an (R-DN) response is disregarded if GR_j (UP) > $Aspire_j$(DN).

A response is called *admissible* if it is either not tabu or else satisfies the aspiration, and is called inadmissible, otherwise. If the response for a goal infeasible variable is inadmissible, then the variable is not permitted to enter the sets G_p and G_s. The only exception to this rule is when G_p is empty. In this case the aspiration by default that allows G_p to contain a variable with a smallest remaining tabu tenure is used.

4. Measures and Cardinality

In this section we present two measures for goal infeasibility GR_j and choice preference CP_j associated with a variable x_j, as well as the definition of the cardinality of sets G_p, G_s and D_0.

4.1. Measures for goal infeasibility and choice preference

i) At the simplest version $GR_j = |x''_j - x'_j|$ that identifies how distant is the (LP') solution x''_j from its current goal value x'_j. For the integer penalty, we first compute the fractions $f_j^+ = \lceil x''_j \rceil - x''_j$ and $f_j^- = x''_j - \lfloor x''_j \rfloor$, and then we set $IP_j(UP) = f_j^+, IP_j(DN) = f_j^-$. The choice preference is expressed as

$$CP_j = (IP_j(UP) + IP_j(DN))(|IP_j(UP) - IP_j(DN)| + w)$$

where w is a small positive weight to give additional influence to the sum of the penalties, as in the case when their absolute difference is zero.

ii) When a new (LP') is defined, every new added goal that is satisfied causes an increase in the objective function of (LP'). However, when the goal infeasibility condition is reached, the inversion of a set of goals may result in a decrease in the objective function of (LP'). Thus, at a more sophisticated level, the goal resistance GR_j can be defined as an estimate of the variation in the objective of (LP'). We use the estimate given by reliability branching proposed by Achterberg et al. (2005). At the beginning of the search the upward ψ_j^+ and downward ψ_j^- pseudocosts of a variable are not reliable due to the lack of information, and in this case strong branching is applied to all variables that take on a fractional value and for each variable this mechanism is repeated until its pseudocost become reliable. At this point the pseudocost for each variable is fixed, thus avoiding the use of post-optimization dual simplex in (LP'). Reliability branching was applied with success to select branching variables in the cut-and-branch method for solving instances from the literature. For this reason this technique was used to compute measures of goal resistance and integer penalty. The integer penalties are given by IP_j (UP) = $f_j^+\psi_j^+$ and IP_j (DN)= $f_j^+\psi_j^+$. The choice preference is given by

$$CP_j = (1-\mu)\min\{IP_j(UP), IP_j(DN)\} + \mu\max\{IP_j(UP), IP_j(DN)\}$$

where μ is a parameter such that $0 \le \mu \le 1$.

4.2. Cardinality of sets

A simple scheme for determining the cardinality of the sets G_p, G_s and D_0 is to define suitable positive fractional parameters f_p, f_s, f_d, and then set $d_0 = |D_0| = f_d \times |D|$, $g_p = |G_p| = f_p \times |G|, g_s = |G_s| = f_s \times |G|$. This scheme is restrictive in that the fractional parameters do not change as the search evolves. The following adaptive strategy here proposed is more elaborate and defines the cardinality of the sets according to the conditions of the region that is being visited. The idea of the algorithm shown below is to seek to solve the goal infeasibility condition by inverting a small number of goals and if this infeasibility persists the emphasis is shifted to free a large number of variables with unsatisfied goals. Consider the parameters $\delta_2 > 1, 0 < \delta_1 < 1, 0 < k \le 1, 0 < f_s \le 1$, the maximum value $0 < f_{d\max} \le 1$ for the fractional parameter f_d and the extreme set cardinality values $g_{\max}, g_{p\min}, g_{s\min}$. Let $Iter$ denote the current iteration and $IterTrans$ the iteration where the search transits from the integer infeasibility condition to the goal infeasibility condition or when the search remains for $k \times m$ consecutive iterations in one of these conditions. The dynamic cardinality of sets G_p, G_s and D_0 is defined by the following algorithm.

Step 0. *Initialization.* $g_p \leftarrow g_{p\min}, g_s \leftarrow g_{s\min}, IterTrans \leftarrow 0, Iter \leftarrow 0$, initial condition: integer infeasibility

Step 1. *Search is in the integer infeasibility condition.* If $(Iter - IterTrans) > k \times m$ then $f_d \leftarrow \min(f_{d\max}, f_d \times \delta_2)$, and $IterTrans \leftarrow Iter$. Go to step 3.

Step 2. *Search is in the goal infeasibility condition.* If $(Iter - IterTrans) > k \times m$, and if $(g_p > g_{\max})$ or $(g_s > g_{\max})$ then $g_p \leftarrow 1$ and $g_s \leftarrow \max(g_{\max}, f_s \times |G|)$. Else, increase alternatively the parameters g_p and g_s by one unit.

Step 3. *Construction of a new* (LP$'$). $Iter \leftarrow Iter + 1$. Create new goals, update the tabu list if the current (LP$'$) is goal infeasible, and solve the new (LP$'$). If the current (LP$'$) is integer infeasible and the new (LP$'$) is goal infeasible, set $f_d \leftarrow f_d \times \delta_1, g_p \leftarrow g_{p\min}$ and $g_s \leftarrow g_{s\min}$. If the new (LP$'$) is goal infeasible go to step 2, else go to step 1.

In step 1, the fraction of unrestricted fractional variables is defined as $f_d \leftarrow \min(f_{d\max}, f_d \times \delta_2)$ and as long as $f_d \le f_{d\max}$, this parameter is slightly increased, for example, $\delta_2 = 1.05$ at every $k \times m$ consecutive iterations in the integer infeasibility condition. Since $0 < \delta_1 < 1$, the fraction f_d is decreased in step 3 according to $f_d \leftarrow (f_d \times \delta_1)$, and this reduction always takes place when the search changes from the integer infeasibility condition to the goal infeasibility condition. The reasoning of this policy is to establish several goals at each iteration, while the search remains in the integer infeasibility condition. When the search transits from the integer infeasibility condition to the goal infeasibility condition, the fraction f_d is substantially reduced by setting, for example, $\delta_1 = 0.5$.

Step 2 deals with the goal infeasible condition and the search tries to resolve conflicts, by altering the minimum number of established goals. Tipically, $g_{p\min} = 1$ and $g_{s\min} = 0$, which implies that one goal is changed and no goals are removed. At every $k \times m$ consecutive iterations g_p and g_s, in that order, are alternately increased by one unit. If g_p or g_s reach $g_{\max}$ and the goal infeasibility condition still persists after the following $k \times m$ iterations, the objective is to move a large number of variables from their goal condition to the free condition and hence $g_s \leftarrow \max(g_{max}, f_s \times |G|)$ and $g_p \leftarrow 1$. Every time the search leaves the goal infeasibility condition, g_p and g_s are reset, respectively, to the initial values $g_{p\min}$ and $g_{s\min}$ in step 3, where a new (LP$'$) is constructed. Better results were achieved by using the adaptive strategy.

5. Core parametric tabu search

In its initial execution, the core method begins by specifying (LP$'$) to be the original relaxation (LP) of (MIP), where no goal conditions exist and N' is empty. It then proceeds as follows.

Step 1. Solve (LP$'$) to obtain an optimal solution (x'', y''). If this solution is (MIP) feasible, update x_0^* and repeat this step to re-optimize (LP$'$). If (LP$'$) has no feasible solution, the method stops and the best solution is optimal. Terminate the solution process if this step has been executed a chosen number of times. Otherwise, continue to Step 2.

Step 2. (a) If the solution to (LP$'$) is goal infeasible, create the sets G_p and G_s to consist of the g_p and g_s highest ranking goal infeasible (and potentially goal infeasible) admissible variables from G, defining admissibility in relation to the current tabu restrictions and aspiration criteria. (b) If the solution to (LP$'$) is not goal infeasible, but is integer infeasible, create the set D_0 to consist of the highest ranking unrestricted fractional variables from D.

Step 3. According to the outcome of Step 2, generate the new goal conditions and identify the new problem (LP$^{'}$). If the stipulations of Step 2(a) apply, update the associated tabu tenures and aspiration values. Then return to Step 1.

5. Computational tests

The algorithms of the core parametric tabu search were coded in C++ by using the version 4.0.2 of the GCC compiler and computational tests were carried on a PC Intel Pentium IV 3.2 GHz, 3Gbyte RAM with the operating system Linux Fedora 4. We tested our implementation on 78 instances from the MIPLIB (Achterberg et al., 2003) and the Mittelmann (2003) test set, and present the best solution values for the parametric tabu search versions PTS-GD and PTS-RB, which denote the goal distance and reliability branching associated with the measures i) and ii) for goal infeasibility, respectively. Such results are compared with the first feasible solution values obtained by the objective feasibility pump (Achterberg and Berthold, 2007) and the the best solution values of CPLEX 10 after the root node was solved and the heuristics were applied. As in (Achterberg and Berthold, 2007) we applied the MIP preprocessing of CPLEX prior to running parametric tabu search and we also set a time limit of one hour in all runs of each instance. The instances of (LP$^{'}$) were solved by CPLEX. The values of the parameters used in our implementation are as follows. The weight M_j depends on the parameter *NIter* that expresses the number of iterations that the memory of an UP or DN condition lasts for a variable x_j was set to 64. The positive parameter *r* that establishes the influence o the weight M_j has value 0.2. For almost all instances, the parameter ε in the objective function constraint $cx + dy \le x_0^* - \varepsilon$ was set to the value 1. The exceptions are the instances *modglob* and *glass*4, which have very large objective functions values, and for this reason the parameter ε takes on the values 1000 and 450,000, respectively. The parameters *w* and μ related to choice preferences assume the values 0.01 and 5/6, respectively. The parameter values for the adaptive strategy that defines the cardinality of the sets G_p, G_s and D_0 are: $f_d = 0.05$, $f_{d\max} = 0.3$, $f_s = 0.05$, $g_{p\min} = 1$, $g_{s\min} = 0$, $k = 0.25$, $g_{\max} = 3$, $\delta_2 = 1.05$ and $\delta_1 = 0.5$.

Table 1 shows the solution value of each instance for each method as well as the percentage relative deviation from the best known solution value. The first column indicates the names of the instances, and the remaining columns show the results for each method, namely the PTS-GD and PTS-RB versions, the objective feasibility pump (Obj. Feas. Pump) and the heuristics from CPLEX (HCplex). Solution values in bold indicate optimal solutions values. The symbol '0^+' is used for solution values with deviation of less than 0.5% in relation to the best known solution. A bar '$-$' means that no solution was found within the time limit of one hour, and deviation equals zero means that the best known solution has been reached. The bottom of this table displays the mean deviation (Mean) over all instances, the mean deviation over those instances for which all methods found a solution, and the number of failures (Failures) which is the number of instances for which a method could not find a feasible solution. Therefore, Mean(56) indicates that all methods found a solution for 56 instances. The parametric tabu search versions obtain a very low deviation as compared with the other methods. However, improved feasibility pump presents a much smaller number of failures compared to the other methods.

Table 2 shows the 78 instances grouped by the range of the percentage deviation relative to the best known solutions. For each method the column 'number' indicates the number of instances that are in a particular range, and column '%' represents the percentage of this number of instances relative to the total of 78 tested instances. The parametric tabu search versions, PTS-GD and PTS-RB are able to find solutions for 53 instances (68%) and 52 instances (66.77%) which have a relative deviation less or equal to 10%, respectively, whereas the improved feasibility pump and the CPLEX heuristics find solutions for 30 instances (38.5%) and 34 instances (43.6%) within the same deviation range. Moreover, PTS-GD and PTS-RB find a larger number of optimal solutions and a fewer number of solutions with deviation greater than 50%, when compared to the other methods.

Since the MIPLIB instances stem from diverse problems we tested the performance of the parametric tabu search versions on 57 instances of the generalized assignment problem, which can be found in the OR-Library. Table 3 shows the average results of the percentage relative deviation from the best known solutions. Each group A, B, C, D, and E corresponds to a set of instances, and the complexity of each group increases from A to E. For the easy groups A and B the PTS-GD and PTS-RB find good solutions with an average deviation close to 1%, and the deviation increases for the groups C, D and E, as expected. The table also shows the mean deviation (Mean) over all instances, the mean deviation over those instances for which the two versions found a solution, Mean (50) and the number of failures. The results obtained with parametric tabu search are promising and we are working on improvements and tests on other problems.

Table 1: Parametric tabu search compared with objective feasibility pump and CPLEX heuristics

Name	PTS-GD Solution value	PTS-GD Dev. %	PTS-RB Solution value	PTS-RB Dev. %	Obj. Feas. Pump Solution value	Obj. Feas. Pump Dev. %	HCplex Solution value	HCplex Dev. %
10teams	**924**	0	**924**	0	952	3	-	-
a1c1s1	16685.8	45	16026.5	39	16076.6	40	21029.4	83
aflow30a	1216	5	1234	7	4105	254	1239	7
aflow40b	2184	87	2220	90	2049	75	1439	23
air04	56385	0+	**56137**	0	57298	2	-	-
air05	26402	0+	**26374**	0	26942	2	27291	3
cap6000	-2.45E+06	0+	-2.45E+06	0+	-2.43E+06	1	-2.45E+06	0+
dano3mip	740.876	4	798.588	12	769.3	8	714.125	0
danoint	**656.667**	0	70.5	7	87	32	-	-
disctom	-	-	-	-	**-5000**	0	-	-
ds	-	-	-	-	-	-	412,5025	0
fast0507	175	1	177	2	179	3	177	2
fiber	498313	23	460130	13	1.21E+06	197	468924	16
fixnet6	**3983**	0	3985	0+	4807	21	4435	11
glass4	2.60E+09	117	4.00E+09	233	3.10E+09	158	-	-
harp2	-7.02E+07	5	-7.09E+07	4	-5.59E+07	24	-7.26E+07	2
liu	1614	8	1892	26	4100	174	4674	212
markshare1	13	1200	6	500	194	19300	230	22900
markshare2	26	2500	28	2700	365	36400	898	89700
mas74	12884.9	9	12180.1	3	19033.1	61	14372.9	22
mas76	40453.4	1	40607.7	2	50124	25	**40005.1**	0
misc07	**2810**	0	**2810**	0	3425	22	**2810**	0
mkc	-402.06	29	-409.56	27	-289.95	49	-528.53	6
mod011	-5.22E+07	4	-5.25E+07	4	-4.56E+07	16	-4.74E+07	13
modglob	2.08E+07	0+	2.08E+07	0+	2.11E+07	2	2.08E+07	0+
momentum1	-	-	-	-	346535	218	527461	383
net12	-	-	-	-	337	57	-	-
nsrand-ipx	56160	10	54400	6	89120	74	55680	9
nw04	**16862**	0	16876	0+	17856	6	16956	1
opt1217	**-16**	0	**-16**	0	**-16**	0	**-16**	0
p2756	3358	7	3144	1	89266	2757	3425	10
pk1	**11**	0	15	36	83	655	18	64
pp08a	7990	9	7820	6	10940	49	8120	10
Pp08aCUTS	7700	5	7670	4	8530	16	8100	10
protfold	-22	29	-	-	-12	61	-20	35
qiu	-320.578	76	**-132.873**	0	625.709	571	173.979	231
rd-rplusc-21	-	-	-	-	171182	3	-	-
set1ch	60499.5	11	62524.5	15	84167.5	54	66772.5	22
seymour	427	1	428	1	445	5	434	3
sp97ar	7.20E+08	7	6.75E+08	0+	9.41E+08	39	6.83E+08	1
stp3d	-	-	-	-	-	-	-	-
swath	554.298	19	544.367	16	1280.95	174	1521.66	226
t1717	332765	70	237217	21	195779	0+	340588	74
tr12-30	167135	28	159582	22	163794	25	-	-
vpm2	14.25	4	14.5	5	15.25	11	15.25	11
l152lav	**4722**	0	**4722**	0	4757	1	4760	1
stein45	**30**	0	**30**	0	35	17	**30**	0
ran8x32	5422	3	5462	4	5817	11	5837	11
ran10x26	4410	3	4448	4	4833	13	4745	11
ran12x21	3997	9	3987	9	4231	15	4080	11

ran13x13	3377	4	3393	4	3820	17	3508	8
binkar10_1	6910.12	2	6888.48	2	7156.21	6	6917.17	3
irp	12160.7	0+	12159.5	0+	12162.4	0+	12162.4	0+
mas284	**91405.7**	0	**91405.7**	0	99522.7	9	93708.1	3
prod1	-49	13	-55	2	-53	5	-49	13
bc1	339.687	2	339.804	2	54.391	63	344.084	3
bienst1	47.25	1	47.25	1	55.5	19	66.5	42
bienst2	568.333	4	55	1	736.667	35	621.667	14
dano3_3	**576.345**	0	**576.345**	0	**576.345**	0	576.396	0+
dano3_4	**576.435**	0	577.370	0+	**576.435**	0	576.499	0+
dano3_5	577.304	0+	578.907	0+	576.994	0+	578.648	0+
mkc1	-603.92	1	-595.36	2	-563.1	7	-604.86	0+
neos1	**19**	0	20	5	68	258	21	11
neos2	1290.55	184	-	-	958.977	111	-	-
neos3	1390.65	277	-	-	1630.21	342	-	-
neos4	-4.61E+10	5	-4.61E+10	5	-4.81E+10	1	-4.83E+10	1
neos5	**15**	0	**15**	0	*	*	15.5	3
neos6	89	7	84	1	93	12	91	10
seymour1	412.646	0	411.681	0	427.063	4	412.448	0
swath1	**379.071**	0	**379.071**	0	439.106	16	879.034	132
swath2	**385.200**	0	388.603	1	641.544	67	1028.07	167
acc-0	**0**	-	**0**	-	**0**	-	**0**	-
acc-1	**0**	-	**0**	-	**0**	-	**0**	-
acc-2	**0**	-	**0**	-	**0**	-	-	-
acc-3	**0**	-	**0**	-	**0**	-	-	-
acc-4	-	-	-	-	-	-	-	-
acc-5	-	-	-	-	**0**	-	-	-
acc-6	-	-	-	-	-	-	-	-
Mean		74		62		922		1909
Mean(56)		75		64		1101		2038
Failures		9		12		4		16

Table 2: Comparison of methods according to the relative deviation range

	PTS-GD		PTS-RB		Obj. Feas. Pump		HCplex	
Range of deviation	Number	%	Number	%	Number	%	Number	%
Less than 1%	30	38.5	28	35.9	16	20.5	19	24.4
1% to 5%	13	16.7	16	20.5	7	9.0	9	11.5
5% to 10%	10	12.8	8	10.3	7	9.0	6	7.7
10% to 25%	4	5.1	6	7.7	14	17.9	15	19.2
25% to 50%	4	5.1	4	5.1	8	10.3	2	2.6
50% to 100%	3	3.8	1	1.3	8	10.3	3	3.8
Greater than 100%	5	6.4	3	3.8	14	17.9	8	10.3
Failures	9	11.5	12	15.4	4	5.1	16	20.5

Table 3: Parametric tabu search for generalized assignment instances

Group	PTS-GD Dev. %	PTS-RB Dev. %
A	0.00	0.00
B	1.06	1.04
C	2.05	2.32
D	4.23	3.10
E	9.62	7.94
Mean	4.02	3.42
Mean(50)	4.05	3.42
Failures	6	7

6. Conclusions

We have conducted a computational study of the core version of parametric tabu search for solving 0-1 MIP problems. In the parametric approach, branching inequalities associated with binary variables, called goal conditions, are imposed indirectly by including them with associated weights in the objective function of the linear programming relaxation of a MIP, denoted LP'. Parametric tabu search involves measures to classify optimal values of binary variables to LP' in suitable sets and assign a response to the variables of each set.

Optimal values of binary variables in LP' that violate goal conditions define the set G of goal infeasible variables. A measure called goal resistance is used to select the set of variables $x_j, j \in G_p \subseteq G$, which take on new goals in the opposite direction, and the set of variables $x_j, j \in G_s \subseteq G$ that are freed from their goal conditions. The set of potentially goal infeasible variables involve variables with satisfied goals and a much smaller goal resistance compared to the goal infeasible variables. The same responses for goal infeasible variables apply to variables that are potentially goal infeasible. The set D_0 contains binary variables with no goal whose optimal solutions in LP' are fractional and have the largest values of a choice preference measure that depends on up and down penalties. Goals are assigned to such variables according to the penalties. Finally, a tabu restriction is associated with a response for a given variable, by forbidding the response from being executed, if the opposing response was executed within the most recent tabu tenure iterations.

We have examined three strategies to measure goal resistance, namely, goal distance (GD), reliability branching (RB) and active-constraint variable ordering (ACVO). The Wilcoxon test shows that ACVO is superior to GD and RB. We have also shown that it is better to have a memory for the weight associated with a goal established in a given iteration that decays exponentially with the number of iteration. In addition, the adaptive strategy that determines the cardinality of the sets G_p, G_s and D_0 according to the conditions of the region that is being visited is more effective than a static strategy.

The Wilcoxon test could not determine any dominance between objective feasibility pump (OFP), CPLEX heuristics (HCPLEX) and parametric tabu search method PTS-first. However, the Wilcoxon test determined that parametric tabu search PTS-3600 outperforms OFP and HCPLEX, which shows the potential of parametric tabu search for obtaining a large number of feasible and high-quality solutions when more running time is allowed. Future research involves the integration of intensification and diversification methods suggested by Glover (2006), such as approaches based on exploiting strongly determined and consistent variables, approaches derived from scatter search and methods based on frequency analysis.

7. References

Achterberg, T. and Berthold, T. (2007), Improving the Feasibility Pump. *Discrete Optimization*, 4:77-86.

Achterberg, T., Koch, T. and Martin, A. (2005), Branching Rules Revisited. *OperationsResearch Letters*, 33:42-54.

Achterberg, T., Koch, T. and Martin, A. (2003), The mixed integer programming library:Miplib. *http://miplib.zib.de*, 4, 2010.

Danna, E., Rothberg, E. and Le Pape, C. (2005), Exploring Relaxation Induced Neighborhoods to Improve MIP Solutions. *Mathematical Programming Series A*, 102 (1):71-90.

Fischetti, M. and Lodi, A. (2003). Local Branching. *Mathematical Programming*, 98 (1-3):23-47

Fischetti, M., Glover, F. and Lodi, A. (2005), The Feasibility Pump. *MathematicalProgramming*, 104 (1):91-104.

Glover, F. (2006), Parametric Tabu-Search for Mixed Integer Programs. *Computers and Operations Research*, 33:2449-2494.

Glover, F. (1978), Parametric Branch and Bound. *OMEGA*, 6 (2):145-152.

Mittelmann, H. (2003), Decision tree for optimization software: Benchmarks for optimization software. *http://plato.asu.edu/bench.html*, 4, 2010.