BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **11**, No 2

Sofia • 2011

A Heuristic Algorithm for Solving Mixed Integer Problems

Krasimira Genova

Institute of Information and Communication Technologies, 1113 Sofia
E-mail: kgenova@iinf.bas.bg

Abstract: *The paper proposes a hybrid heuristic algorithm, which uses procedures for search of feasible integer directions with one or two nonzero components and linear optimization. This algorithm is iterative and it combines constructive and locally improving strategies for finding a new current solution. A subsequence of sub-problems is solved, aimed at seeking a feasible solution of the general Mixed Integer Problem (MIP), after which the feasible solution found is improved with respect to the problem objective function. The algorithm is characterized by polynomial-time computing complexity.*

Keywords: *Mixed integer programming, heuristic algorithm, polynomial-time complexity.*

1. Introduction

Many real life optimization problems (transport problems, problems of scheduling and distribution, resources planning, etc.) are presented by linear mixed integer models. The integer variables refer to particular items, which are indivisible (number of machines, vehicles and others), and the continuous variables reflect mainly the estimates of price, time and other divisible objects. Nowadays Mixed Integer Problems (MIP) are used in applications oriented towards decision making in many industrial and business activities. This is due not only to the increased computing power of modern computers, but also to improved MIP solvers, which enable easier formulation and solution of these models [1].

The computing complexity of the integer problems, which have not continuous variables, comes from their combinatorial nature. In mixed integer problems the computing complexity is increased, since it is necessary to find both the optimal combination of the integer variables and the optimal value of the continuous variables. Problems of this type belong to the Non-deterministic Polynomial-time (NP) class. The well known methods for optimal solution finding (such as branch-and-bound or cutting planes) have exponential computing complexity and may require considerable computing time even for not large problems [7, 13]. These computing problems have lead to the development of a large number of heuristic algorithms. Heuristics are a very important tool in applied optimization [3, 15]. Very often they are only applicable in solving large and complex problems, met in practice. Their main purpose is to ensure reasonable solution time of the optimization problem, close to the optimal feasible solution. Moreover, they can be very useful in complex exact methods for finding feasible initial solutions and also for speeding up the search process at given steps of these methods (such as in a node of the branch and bound tree) [2, 5, 6]. The heuristic algorithms may be divided in two groups: oriented towards problems with a specific structure [8, 17] and general purpose methods [9-12, 14, 16]. In order to find application in commercial software products, the heuristic algorithms are evaluated in relation to their efficiency, as well as to techniques universality, i.e., their applicability and success for different types of problems [3].

In this paper a heuristic algorithm is proposed, designed to solve linear mixed integer problems in a general form, the main idea of which is to combine the concept of the component algorithms of feasible directions [16] and linear optimization. The search for an approximate solution of MIP problems is executed in three successive phases. The purpose of the first phase is to choose an appropriate initial solution, closest to the feasible region and suitable for reaching a good solution with relation to the optimal solution. During the execution of the second phase a feasible solution of MIP problem (in case it exists) is sought, and the purpose of the third phase is to improve the solution found with respect to the objective function. The algorithm is iterative and at each following iteration new current solutions are found, that decrease the total infeasibility or improve the objective function of MIP problem.

2. Basic idea of the algorithm

This algorithm is intended to solve a linear mixed integer problem in a general form:

(1) $$z = \sum_{j\in J} c_j x_j + \sum_{l\in L} h_l y_l \to \max ,$$

under the constraints

(2) $$\sum_{j\in J} a_{ij} x_j + \sum_{l\in L} g_{il} y_l \le b_i , \; i \in I,$$

(3) $$0 \le x_j \le d_j, \quad j \in J,$$

(4) $$0 \le y_l \le u_l, \quad l \in L,$$

(5) $$x_j - \text{ integer for } j \in J,$$

where $I = \{1, 2, \ldots, m\}$, $J = \{1, 2, \ldots, n\}$, $L = \{1, 2, \ldots, k\}$, a_{ij} are elements of matrix A of dimension $m\times n$, and g_{il} are elements of matrix G of dimension $m\times k$. The coefficients of the objective function are determined by the n-dimensional vector c and k-dimensional vector h. The feasible region of the problem is formed by constraints (2)-(5), where b is an m-dimensional vector of the constraints right hand sides, and the vectors d and u – being respectively n-dimensional and k-dimensional, set the upper bounds of the integer and continuous variables. This problem belongs to the class of NP-hard problems, because there does not exist an exact algorithm, that finds an optimal or even only feasible solution for a time duration that is in polynomial dependence on the problem parameters.

The algorithm proposed is hybrid. It uses procedures for search of internal feasible integer directions with one or two non-zero components [16] in order to improve the integer part of the current solution and an Linear Problem (LP) solver for the continuous part, respectively. This algorithm is iterative and it combines constructive and locally improving strategies for finding a new current solution. Sub-problems are constructed consecutively, their feasible region being defined by all problem constraints (1)-(5), satisfied at the current iteration, and the objective function is one of the remaining unmet constraints. If a feasible solution is already found, the objective function (1) is included and every current solution found leads to its gradual improvement. The algorithm operation may be regarded as a sequence of three phases.

Phase 1. Selection of an initial solution

The algorithm can start its work from an arbitrary initial solution, but the appropriate choice can considerably decrease the computing time and improve the possibilities for finding a feasible, as well as a better approximate solution. Two types of initial solutions are suggested, for which it is expected that the largest number of problem constrains (1)-(5) are satisfied.

The first type of an initial solution is connected with the optimal solution of the relaxed continuous problem (1)-(4). The following way is proposed to determine the components of this initial solution (x^0, y^0):

$$x_j^0 = \begin{cases} [\hat{x}_j] + 1 & \text{if} \quad \hat{x}_j - [\hat{x}_j] \geq 0.3, \\ [\hat{x}_j] & \text{otherwise}, \end{cases}$$

$$j \in J;$$

$$y_l^0 = \hat{y}_l - 1, \quad l \in L;$$

$(\hat{x}, \hat{y})$ are the components of the optimal solution of the continuous problem (1)-(4) and $[\hat{x}_j]$ is the integer part of $\hat{x}_j$. The choice of this initial point is based on the idea to start the search in a neighbourhood, close to the optimal solution, but for a more specific structure of the feasible region, finding feasible solutions may be hampered.

For the other type of initial solutions (x^0, y^0), starting with a smaller general infeasibility of MIP problem is intended. It is suggested that these are solutions, close to the line, connecting the optimal solution of the continuous problem (1)-(4) and Chebyshev's centre of the feasible region of the problem. Chebyshev's centre is defined, solving the following continuous problem.

(6) $$q \to \max,$$

under the constraints:

(7) $$\sum_{j \in J} a_{ij} x_j + \sum_{l \in L} g_{il} y_l + q \le b_i, \ i \in I,$$

(8) $$x_j + q \le d_j, \quad j \in J,$$

(9) $$y_l + q \le u_l, \quad l \in L,$$

(10) $$x_j \ge 0, \quad j \in J,$$

(11) $$y_l \ge 0, \quad l \in L,$$

(12) $$q \ge 0.$$

Let $(\hat{x}, \hat{y})$ and $(\tilde{x}, \tilde{y})$ denote the components of the optimal solution of the continuous problem (1)-(4) and of Chebyshev's centre respectively (the optimal solution of the problem (6)-(12)). The following scheme is proposed for the components of this initial solution (x^0, y^0):

$$x_j^0 = \begin{cases} [x_j^{'}] + 1 & \text{if} \quad x_j^{'} - [x_j^{'}] \ge 0.3, \\ [x_j^{'}] & \text{otherwise}, \end{cases}$$

$$j \in J,$$

where $x_j^{'} = 0.25\tilde{x}_j + 0.75\hat{x}_j$; $y_l^0 = 0.25\tilde{y}_l + 0.75\hat{y}_l, \quad l \in L.$

Phase 2. Search for a feasible solution

At every iteration k of this phase the so called infeasibility measure $\overline{b_i^k}$ is determined for each of the constraints (2) of the initial MIP problem:

$$\overline{b_i^k} = b_i - \sum_{j \in J} a_{ij} \ x_j^{k-1} - \sum_{l \in L} g_{il} \ y_l^{k-1}, \quad i \in I.$$

Depending on the sign of $\overline{b_i^k}$, two sets are formed: F_1^k – a set of the indices of the unfulfilled constraints (2) and F_2^k – a set of the indices of the satisfied constraints (2):

$$F_1^k = \{i \in I, \quad \overline{b_i^k} < 0\},$$

$$F_2^k = \{i \in I, \quad \overline{b_i^k} \ge 0\}.$$

The indices of the constraints in the set F_1^k are ordered according to the absolute value of the infeasibility measure of the corresponding constraint, the constraint with the largest infeasibility degree being placed first.

The following problem is formulated and solved for the first unsatisfied constraint of the set F_1^k at iteration k (let us denote it by i_k):

(13) $$z_{k+1} = -\sum_{j\in J} a_{i_k j}\, x_j - \sum_{l\in L} g_{i_k l} y_l \rightarrow \max,$$

(14) $$\sum_{j\in J} a_{ij} x_j + \sum_{l\in L} g_{il} y_l \le b_i\,, \quad i \in F_2^k\,,$$

(15) $$z_{k+1} + \bar{b}_{i_k}^k \ge 0\,,$$

(16) $$0 \le x_j \le d_j\,, \quad j \in J,$$

(17) $$0 \le y_l \le u_l\,, \quad l \in L,$$

(18) $$x_j \text{ – integer for } j \in J.$$

Each feasible solution of problem (13)-(18) satisfies all constraints with indices from the set F_2^k and decreases the infeasibility of at least one constraint of the set F_1^k with an index i_k. Problems of the type (13)-(18) are formulated and solved until a feasible solution of the initial problem (1)-(5) is found, i.e., until $F_1^k \ne \varnothing$. If any of these problems cannot be solved, a new problem is formulated for the next unsatisfied constraint of the set F_1^k. If for all problems with constraints from this set, no feasible solution is found, it is necessary to start Phase 1 for selection of another initial solution. The number of restarts in Phase 1 is a parameter for operation stopping of the heuristic algorithm herein offered. In this case the algorithm has not succeeded to find a feasible solution of problem (1)-(5), but this does not mean that MIP problem (1)-(5) has not a feasible solution.

Phase 3. Improvement of the feasible solution found

In order to improve the feasible solution found, a new problem of the following type is formulated:

(19) $$z_{k+1} = \sum_{j\in J} c_j x_j + \sum_{l\in L} h_l y_l \rightarrow \max$$

(20) $$\sum_{j\in J} a_{ij} x_j + \sum_{l\in L} g_{il} y_l \le b_i\,,\ i \in I,$$

(21) $$-z_{k+1} + z_k + \varepsilon \le 0\,,$$

(22) $$0 \le x_j \le d_j\,, \quad j \in J,$$

(23) $$0 \le y_l \le u_l\,, \quad l \in L,$$

(24) $$x_j \text{ – integers for } j \in J,$$

where z_k is the value of the objective function for the feasible solution, found at the previous iteration, and ε is a relatively small positive number.

In case a solution of problem (19)-(24) is found, this means that a new better solution of the initial MIP problem (1)-(5) is discovered. After that a new problem of the type (19)-(24) is formulated and solved. If any of these problems cannot be solved, the algorithm operation is terminated. The last feasible mixed integer solution found is the best feasible solution, found by the suggested heuristic algorithm.

As it can be seen, solving of the initial problem (1)-(5) is reduced to solving a series of problems of the type (13)-(18) and (19)-(24). In order to solve these sub-problems, feasible values of the integer part of the solution are sought consequently according to the following scheme:

(25) $$x^k = x^{k-1} + h^k p^k,$$

where p^k is an integer vector, setting a feasible integer direction from point x^{k-1}. It is selected from the set of *n*-dimensional vectors with one or two nonzero components. The step length is h^k. After that in order to find feasible values of the continuous part of solution y^k, problem (13)-(18) or (19)-(24) is solved as a standard linear optimization problem with fixed values x^k for the integer variables.

The feasible integer direction p^k from the point x^{k-1} for problems of the type (13)-(18) is defined as a vector, for which the following conditions are fulfilled at $h^k \geq 1$:

(26) $$\bar{b}_{i_k}^{k} - \bar{b}_{i_k}^{k-1} > 0 \quad \wedge \quad \bar{b}_i^k \geq 0, \quad i \in F_2^{k-1},$$

where

(27) $$\bar{b}_{i_k}^{k} = \bar{b}_{i_k}^{k-1} - h^k \sum_{j \in J} a_{i_k j} p_j^k.$$

The conditions of feasibility of the integer direction p^k from point x^{k-1} for problems of the type (19)-(24), at $h^k \geq 1$ are set as follows:

(28) $$-z_k + z_{k-1} + \varepsilon \leq 0 \quad \wedge \quad \bar{b}_i^k \geq 0, \quad i \in I,$$

(29) $$z_k = z_{k-1} + h^k \sum_{j \in J} c_j p_j^k.$$

Condition (26) indicates that when choosing a feasible integer direction, the infeasibility of the constraint with an index i_k decreases. Condition (28) guarantees that in this case the objective function of the initial problem is improved. In the paper [16] a heuristic algorithm of the internal feasible integer directions is described, designed to solve linear, entirely integer problems. At each iteration of this algorithm, some procedures are used to find feasible integer directions with one or two nonzero variables. These procedures may be applied to find the integer variables x^k that satisfy the conditions (26) or (28) in conformance with scheme (25).

When solving problem (13)-(18) after a feasible solution (x^k, y^{k-1}) is found, the solution passes to optimizing the values of the continuous variables, which satisfy the feasibility conditions of the problem. The optimization problem, solved by an LP solver, is:

$$z_k(x^k, y) = -\sum_{l \in L} g_{i_k l} y_l - \sum_{j \in J} a_{i_k j} x_j^k \rightarrow \max$$

$$\sum_{l \in L} g_{il} y_l \leq b_i - \sum_{j \in J} a_{ij} x_j^k, \quad i \in F_2^{k-1},$$

$$z_k(x^k, y) + \bar{b}_{i_k}^{k-1} \geq 0,$$
$$0 \leq y_l \leq u_l, \quad l \in L,$$

where $\bar{b}_{i_k}^{k-1}$ is the value of non-fulfillment of constraint i_k at the previous iteration.

The obtained value of the continuous variables y^k forms the solution of problem (13)-(18), which fulfills, or at least decreases the infeasibility of the constraint i_k and does not break the satisfied constraints from the set F_2^k.

In an analogous way, in order to find an optimal solution of problem (19)-(24), after finding feasible values for the integer variables x^k, the following linear optimization problem is solved:

$$z_k(x^k, y) = \sum_{l \in L} h_l y_l + \sum_{j \in J} c_j x_j^k \to \max,$$
$$\sum_{l \in L} g_{il} y_l \leq b_i - \sum_{j \in J} a_{ij} x_j^k, \quad i \in I,$$
$$-z_k(x^k, y) + z_{k-1} + \varepsilon \leq 0,$$
$$0 \leq y_l \leq u_l, \quad l \in L,$$

where z_{k-1} is the value of the objective function at the previous iteration.

A solution of the initial MIP problem (1)-(5) (x^k, y^k) is found with the obtained value of the continuous variable y^k, which is better with respect to the objective function in comparison to the one, found at the previous iteration.

3. Algorithm main features

The heuristic algorithm proposed is denoted to find an approximate feasible solution of mixed linear integer problems in a general form. For this purpose it includes integrated heuristic procedures for finding feasible integer directions and linear optimization. Since the problems, for which it is intended, are NP-problems, this means that there is no guarantee that it can be used to find a solution to every problem. That is why it is of interest to evaluate the main features of the algorithm, such as computing complexity, conditions for finding a feasible solution, possibilities for entering a cycle, etc.

In order to find a feasible solution of the initial MIP problem (1)-(5), a number of $m - 1$ problems of the type (13)-(18) must be solved in the worst case. Feasible integer directions with one or two nonzero components are sought at each iteration. This limiting of the number of nonzero components is connected with the fact that the search for feasible integer directions is reduced to solving a system of linear inequalities. The problem of solving a system of linear inequalities with one variable is easy from a computing viewpoint. But solving a system of linear inequalities with two variables is more difficult. Solving a system of linear inequalities with three variables is already as difficult as solving the problem in the general case. In [16] it is proved that in order to determine an integer point x^k from the integer point x^{k-1}, a number of $O(q^2 n^2)$ elementary operations of the type +, –,

\, ×, ≥, ≤ are required, where q is the number of constraints of the current n-dimentional problem being solved to find a feasible solution. In order to find the continuous part of the current solution y^k, a continuous LP problem is solved. These problems belong to P-class with respect to their computing complexity. The most often used method for their solution is the simplex method, and although from a theoretic viewpoint it requires exponential time complexity in the worst case, its wide practical application has shown that it has polynomial-time average-case complexity. Under the condition that at each iteration k there exists a feasible integer direction with one or two nonzero components, the algorithm finds a feasible solution for a finite number of iterations [16]. Hence, finding a feasible solution (x^k, y^k) of problem (1)-(5) can be executed for time that is in polynomial dependence on the problem parameters. The improvement of the feasible solution obtained $(x^k,\ y^k)$ is done in an analogous way to finding a feasible solution, but problem (19)-(24) is used. Thus it might be concluded that the computing complexity of the heuristic algorithm considered is polynomial-time.

From a theoretical viewpoint finding of a feasible solution from an arbitrary initial point of MIP problem is a very difficult task. This is true for every approximate algorithm of mixed integer programming. If at a given iteration, when solving a problem of the type (13)-(18) or (19)-(23), no feasible solution can be found, then the algorithm terminates its functioning. However, this does not mean that the corresponding problem has not a solution. If the problem of type (13)-(18) cannot be solved, the algorithm cannot find a feasible solution of the initial MIP problem. When solving a problem of the type (19)-(23), the termination of the algorithm operation means that the last feasible solution of the initial problem found cannot be improved. In case no feasible integer solution with one or two nonzero components can be found for problem (13)-(18) which is currently solved, a new problem of the same kind is formulated with another current non-fulfilled constraint of the initial point from set F_1^k. In the worst case the number of these constraints is $m - 2$. If after that a new current solution (x^k, y^k) is not found again, the algorithm passes to a procedure of selecting a new initial point. Different strategies may be used to alter the initial point. It may be selected along the direction from the last solution found towards Chebyshev's centre, determined by problem (6)-(12) or along the direction from the optimal solution of the continuous problem towards Chebyshev's centre. Phase 2 is started from the new initial point with a search process for new feasible directions with one or two nonzero components for the integer part of the current solution and optimization of the corresponding continuous part of the current solution. In order to ensure protection against falling into a cycle, a Tabu list is used and a stop parameter, limiting the number of restarts with a new initial point.

4. Conclusion

In many practical problems of medium and large dimension, finding a good feasible solution is entirely satisfactory, even in case its deviation from the optimal solution is unknown, using reasonable computing resources at that. The development of heuristic algorithms, as well as hybrid algorithms, that combine the advantages of different approaches (exact and heuristic) and have polynomial-time complexity, is a problem of the present day for many researchers.

The heuristic algorithm, proposed to solve MIP problems, will be included in a web-based interactive system for optimization and decision making, developed by a team at the Institute of Information and Communication Technologies at BAS. Its application in this software system will be utilized both for approximate solution or speeding the discovery of exact solutions in applied single-criterion MIP problems, and also for a solver in the realization of interactive classification-oriented algorithms for multiobjective integer optimization.

Acknowledgement. The paper is partially supported by the Bulgarian National Science Fund, Grant No DTK02/71 “Web-Based Interactive System, Supporting Building Models and Solving Optimization and Decision Making Problems” and by the European Social Fund and Bulgarian Ministry of Education, Youth and Science under Operative Program “Human Resources Development”, Grant No BG051PO001-3.3.04/40.

References

1. Ashford, R. Mixed Integer Programming: A Historical Perspective with Xpress-MP. – Ann Oper Res, **149**, ISSN: 1572-9338, 2007, 5-17.
2. Balas, E., S. Schmieta, C. Wallace. Pivot and Shift – A Mixed Integer Programming Heuristic. – Discrete Optimization, 1, ISSN: 1572-5286, 2004, 3-12.
3. Bixby, R., E. Rothberg. Progress in Computational Mixed Integer Programming – A Look Back from the other Side of the Tipping Point. – Ann. Oper. Res., **149**, ISSN: 1572-9338, 2007, 37-41.
4. Chen, Der-San, R. G. Batson. Yu Dang. Applied Integer Programming: Modeling and Solution, Wiley, 2010, ISBN: 978-0-470-37306-4.
5. Danna, E., E. Rothberg, C. Le Pape. Exploring Relaxations Induced Neighborhoods to Improve MIP Solutions. – Mathematical Programming, **102 (A)**, 2005, 71-90.
6. Fischetti, M., A. Lodi. Local Branching. – Mathematical Programming, **98 (B)**, 2003, 23-47.
7. Garey, M., D. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, W. H. Freeman, 1979.
8. Gendreau, M, Jean-Yves Potvin. Metaheuristics in Combinatorial Optimization. – Ann. Oper. Res., **140**, 2005, 189-213.
9. Glover, F., M. Laguna. General Purpose Heuristics for Integer Programming: Part I. – Journal of Heuristics, **2**, 1997, ISSN: 1572-9397, 343-358.
10. Glover, F., M. Laguna. General Purpose Heuristics for Integer Programming: Part II. – Journal of Heuristics, **3**, 1997, ISSN: 1572-9397, 161-179.
11. Ibaraki, T., T. Ohashi, H. Mine. A Heuristic Algorithm for Mixed Integer Programming Problems. – Mathematical Programming Study, **2**, 1974, 115-136.
12. Luo, Yuh-Chyun, M. Guignard, Chun-Hung Chen. A Hybrid Approach for Integer Programming Combining Genetic, Algorithm, Linear Programming and Ordinal Optimization. – Journal of Intelligent Manufacturing, **12**, 2001, 509-519.
13. Papadimitriou, C. H., K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. NJ, Englewood Cliffs, Prentice-Hall, 1982.

14. S g u r e v, V., V. V a s s i l e v, P. V l a d i m i r o v. An Algorithm of External Feasible Integer Direction for Integer Programming Problems. – In: J. D. Coelo and L. V. Tavares (Eds.), OR Models on Microcomputers, North-Holland, 1986, 137-146.
15. S i l v e r, E. A. An Overview of Heuristic Solution Methods. – Journal of the Operational Research Society, **55**, 2004, 936-956.
16. V a s s i l e v, V., K. G e n o v a. An Algorithm of Integer Feasible Directions for Linear Integer Programming. – European Journal of Operational Research, **52**, 1991, 203-214.
17. W i l b a u t, Ch., S. H a n a f i. New Convergent Heuristics for 0-1 Mixed Integer Programming. – European Journal of Operational Research, Vol. **195**, 2009, No 1, ISSN: 0377-2217, 62-74.