International Journal of Industrial Engineering & Production Research

March 2012, Volume 23, Number 1
pp. 23-33

http://IJIEPR.iust.ac.ir/

ISSN: 2008-4889

A Heuristic Approach for Solving LIP with the Optional Feasible or Infeasible Initial Solution Points

Yahia Zare Mehrjerdi*

Yahia Zare Mehrjerdi, *Department of Industrial Engineering, Yazd University, Yazd, Iran,*

KEYWORDS

Linear Programming,
Integer Programming,
Heuristic,
Decision Making,
and Interactive

ABSTRACT

An interactive heuristic approach can offer a practical solution to the problem of linear integer programming (LIP) by combining an optimization technique with the Decision Maker's (DM) judgment and technical supervision. This is made possible using the concept of bicriterion linear programming (BLP) problem in an integer environment. This model proposes two bicriterion linear programs for identifying a feasible solution point when an initial infeasible solution point is provided by the decision maker or when the searching process leaves the region of feasibility seeking for a better pattern to improve the objective function. Instructions regarding the structure of such BLP problems are broadly discussed. This added property offers a great degree of flexibility to the decision making problem solving process.

The heuristic engine is comprised of four algorithms: Improve, Feasible, Leave, and Backtrack. In each iteration, when a selected algorithm has been terminated, the DM is presented with the results and asked to reevaluate the solution process by choosing an appropriate algorithm to follow. It is shown that the method converges to the optimal solution for most of the time. A solution technique for solving such a problem is introduced with sufficient details.

1. Introduction

Within the last three decades the dynamic role of super computer in advancing technology and enhancing productivity has been astonishing and thrilling. The small and medium sized firms have been indirectly disqualified from the benefits of super computers as a result of either insufficient budget or its tremendous expenses. The new generation of microcomputers, however, has diminished this obstacle and introduced a new dimension into the concept of supply and demand regarding the computer software and hardware. Microcomputer has successfully demonstrated its important role in corporate success and achievements.

The daily growth in utilizing microcomputers in industry has made the design and development of new and sound competitive computer programs and intelligent software more apparent.

A number of heuristic approaches for pure and mixed integer LP problems appeared in the literature. Echols and Cooper [1] used the concept of "Direct Search" methods to develop a solution procedure for LIP problems. According to authors, the procedure is successful on small sized problems. Senju and Toyoda [5] and Toyoda [6] proposed heuristic approaches for solving LP problems with 0-1 decision variables. The

* **Corresponding author: Yahia Zare Mehrjerdi**
Email: *yazm2000@yahoo.com*
Paper first received July 07, 2011, and in revised form Oct. 20, 2011.

Senju-Toyoda's heuristic approach requires that all technological and cost coefficients to be positive. Martin [8] designed a heuristic technique for solving 0-1 type LIP problems. The heuristic approach developed by Kochenberger, McCarl, and Wyman [3] is not limited to binary decision variables. Magazin and Oguz [4] designed a heuristic algorithm for the 0-1 knapsack problem. A combination of Senju-Toyoda's heuristic approach and the Everett's Generalized Lagrangian Multiplier Procedure is being employed in their development. Cote and Laughton [2] proposed an approach for solving mixed integer LP problems with Benders type heuristics.

In 1998, Lokketangen and Glover [11] proposed a tabu search algorithm for solving a zero-one mixed integer programming problem. In 2008, Wilbaut, C. and Hanafi S. [12] addressed a new convergent heuristics for 0-1 mixed integer programming problem. Kodayif and Kandemir et al. [13] addressed an integer linear programming based tool for wireless sensor networks. Canakgos and Beasley [14] developed a mixed integer programming approach for index tracking and enhanced indexation.

Boland et al. [15] have discussed a new integer linear programming approaches for course timetabling. Gnanendran et al. [16] developed the problem of selecting a subset of items to stock from among a large set of potential items. Roslöf, J. et al. [17] considered a large-scale industrial production scheduling problem which includes the allocation of a number of production runs with release and due dates into a processing unit. A mixed integer linear programming (MILP) model is used to describe the scheduling task. Vassilev, V. and Genova, K. [18] proposed an approximate algorithm for solving pure integer problems. The algorithm belongs to the class of component algorithms using feasible integer directions. Boehning et al. [19] observed that the solution of large scale integer linear programming models is dependent upon the branch and bound technique.

Authors described a parallel branch and bound algorithm which achieves super linear efficiency in solving integer linear programming models. Benders et al. [20] have discussed a property of assignment type mixed integer linear programming problems. Authors proved that rather tight upper bounds can be given for the number of non-unique assignments that are achieved after solving the linear programming relaxation of some types of mixed integer linear assignment problems. Brosh et al. [21] have addressed the warehouses location problem using a mixed integer programming and a heuristic algorithm. A simplification of freight rates schedules, based upon shipments consolidation and a linear regression of rates *vs.* distances was made.

The heuristic approach described here has the following advantages:

1. It can begin with either a feasible or an infeasible starting solution point
2. If the selected starting point is infeasible then algorithm searches for the best possible feasible solution using the concept of the BLP problem.
3. No restriction of any kind is imposed on the coefficients in the problem
4. The decision variables can be binary or of mixed integer type values.
5. The procedure is interactive, which means it goes back and forth between the model and the DM.

The search for a feasible solution point upon an identified infeasible point is not a simple task. Since, the selected mathematical model for finding a feasible solution is comprised of two linear objective functions, BLP is chosen as a tool for solving this problem. Knowing that more than one feasible point is potentially eligible to be selected, a promising technique such as BLP can become highly valuable in tradeoff identifications. The conflicting nature of objectives $f_1(X)$ and $f_2(X)$ do constraints their simultaneous optimization. A feasible solution point that best fits these two objectives and satisfies the DM's requirements is defined as "a Compromise Solution" [10, 11]. In this case, at each stage, the DM decides whether he/she is willing to compromise the achievement of one objective against the other dis-achievement. However, stopping rules are also defined according to the DM criteria.

The point of departure of this paper from the previous works is threefold. First, it introduces an interactive heuristic approach that has a powerful searching capability to originate the process of the solution from either a feasible or an infeasible solution point. Second, it facilitates the man-machine interface process by proposing a menu driven computer algorithm. Third, the proposed algorithm, which is coded in Pascal computer language, can be executed on the IBM-PC or compatible computers.

2. The BLP Model

Generally speaking, in real life situations, it requires a great deal of interaction with the DM to get a reliable response and professional feedbacks. This is not, however, an unusual problem. It is common among all interactive CADM procedures (Adulbahan and Tabucanon [11]). As an alternative tool, this method attempts to reduce the DM's presence and introduces a surrogate and simulating role for him/her instead. This expediency is presented by the functional utilities of the following kind:

$$U_1(Z_1, Z_2) = Max\{|\, Z_1.Z_2 \,|; \forall X_j, X_s\} \qquad (1)$$

$$U_2(Z_1, Z_2) = Min\{|\, Z_1.Z_2 \,|; \forall X_j, X_{sr}\} \qquad (2)$$

More details concerning variables X_s and X_{sr} are given in appendix A. However, the first utility function should be employed when both objectives of the BLP are of the maximization type. On the other hand, when we are dealing with two minimization types objectives we need to use the second utility function. The corresponding feasible solution of the utility function U1(Z1, Z2) will be referred to as the "compromise solution'.

If the compromise solution point of $X_c^* = (X_{tc}^*, X_{sc}^*)^t$ does not satisfy the feasibility conditions of the LIP problem then algorithm would proceed as follows. In the next step, the feasibility of the corresponding compromise solution of the next maximum values; read $U_1(Z_1, Z_2)$ = Second Max {.}, or $U_1(Z_1, Z_2)$=Third Max{.},…,etc.; would be investigated.

The process continues until either a feasible solution point has been obtained or the list of the values has been exhausted.

If no feasible solution for the LIP can be obtained in this manner, then the program would proceed to the next step as instructed. The above discussion is demonstrated by a simple numerical example and presented in Appendix A.

3. The LIP Model

The problem studied in this article is presented in Figure 1 by model (3)-(6). For generalization purposes, no restrictions will be placed on the signs of C_j, a_{ij}, or b_i. Notations used in this study are:

m= Number of constraints

n= Number of variables

X_j = the jth component of the current feasible or infeasible solution point

X_{0j}= the jth component of the initial feasible point

X'_{0j} = the jth component of the initial infeasible point

C_j = the objective function coefficient of the jth variable

b_i = the right hand side of the ith constraint

a_{ij}=technological coefficient of the jth variable in the ith constraint

R= a decreasing order of variables whose ranks are based on their coefficients in the objective function

V_1, V_2 = Vector of technological coefficients.

P1: $$Maximize Z = \sum_{j=1}^{n} C_j X_j \quad (3)$$

S.t.:

$$\sum_{j=1}^{n} a_{ij} X_j \ge b_i \quad \forall\ i=1,2,3,\dots,r \quad (4)$$

$$\sum_{j=1}^{n} a_{ij} X_j \le b_i \quad \forall\ i=r+1,\dots,m \quad (5)$$

$$X_j \ge 0 \text{ and integer, } j=1,2,\dots,n \quad (6)$$

Fig. 1. Problem 1

4. The Heuristic Engine

Because an interactive method involves a high degree of human interaction in problem solving, a menu driven computer algorithm would ease the interaction processes. The list of options available for the users in the menu is illustrated in Figure 2. The body of this heuristic engine is comprised of algorithms **Improve**, **Feasible, Leave**, and **Backtrack**. In the section that follows, following algorithms are discussed in detail:

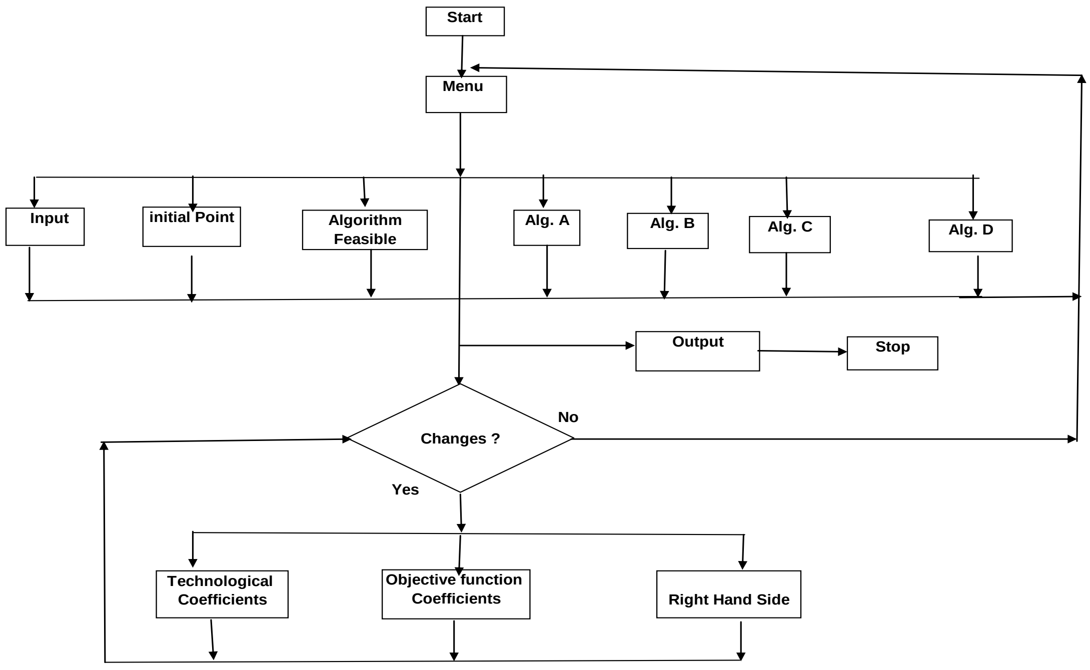
1. **Algorithm Improve**: to improve the current optimal situation
2. **Algorithm Feasible**: to search and find a feasible solution point whenever an initial infeasible solution point is given.
3. **Algorithm Leave**: to leave the feasible region for improving the optimal value of the objective function.
4. **Algorithm Backtrack**: to backtrack a suitable variable to stimulate the solution process more actively.

The "interactive" method is a general approach in which a high degree of human interaction is incorporated into the problem solving process [9, 10]. One of the main advantages of interactive solution techniques is the involvement of the decision makers (DMs) in the solution process of the problem. In spite of consuming the DMs valuable time it is accounted as a learning process for the DM. The "interactive" procedure for the LIP problem involves the following steps:

1. The DM identifies an initial solution for the machine
2. Algorithm searches for the most feasible solution and prompts the DM with the results
3. The DM chooses a feasible strategy to guide the forthcoming steps to improve the current solution
4. The process of steps (2) and (3) continues until the DM is satisfied with the final solution.

The DM interacts with the machine by entering an alphabetic character. The alpha character is an instruction that will guide the computer algorithm to either improve the current best solution or perform otherwise as instructed.

The "interactive' characteristic of the heuristic approach provides a learning process for the DM. This interactive heuristic approach becomes highly valuable as interaction between the man and computer increases and a basic knowledge of the system is thoroughly gained.



Alg.A = Algorithm Improve **Alg.B = Algorithm Feasible**
Alg.C = Algorithm Leave **Alg.D. =Algorithm Backtrack**

Fig. 2. The Heuristic Engine Structure

4.1. Algorithm Improve

In algorithm **Improve**, the solution process progresses toward the optimality when a feasible solution point is available. Simply, this algorithm maximizes the objective function on one variable at a time. This is, it starts with a variable with the largest coefficient in the objective function and moves toward the smallest by increasing or decreasing the value of each variable in the direction which results in the improvement of the objective function.

This algorithm takes the following steps:
Step1. Set k=1
Step2. Choose the kth element of the vector R and call it X_t
Step3. Solve problem P2 presented in figure 3,
Step4. An algorithmic form of this step is shown in figure 4.

P2: $$MaximizeZ=\sum_{j=1}^{n} C_j X_j \quad (7)$$

S.t.:

$$a_{it} X_t \geq b_i - \sum_{\substack{j=1 \\ j \neq t}}^{n} a_{ij} X_{0j} \qquad \forall\ i=1,2,3,\ldots, r \quad (8)$$

$$a_{it} X_t \leq b_i - \sum_{\substack{j=1 \\ j \neq t}}^{n} a_{ij} X_{0j} \qquad \forall\ i = r+1,\ldots,m \quad (9)$$

$X_t \geq 0$ and integer, j=1,2,…., n (10)

Fig. 3. Problem 2

```
Begin
If (Problem $P_2$ has a solution) & ($k \leq n$) then
$X_0$:= New point
k := k+1
Go to step 2
End

If (Problem $P_2$ has no solution) & ($k \leq n$) then
        k:= k+1
        Go to step 2
        End
If (($k \geq n$) & if (Final point = Starting Point)) then
        Terminate
        Else
        Go to step 1
        End
End
```

Fig. 4. A Computer algorithm for step 4

4.2. Algorithm Feasible

The primary objective of algorithm **Feasible** is to find a feasible point when the initial solution point provided by the user is infeasible.

It searches for the most feasible solution that decreases the overall number of the iterations toward the optimality. This algorithm is based upon the assumption that an infeasible solution point, say X'_0 is given, vector R is constructed, infeasibility is checked and the violated constraint is identified as p, where $1 \leq p \leq M$.

Algorithm **Feasible** can be summarized as follows:

Step 1: Set k=1
Step 2: Choose the kth element of R and call it X_t
Step 3: If the violated constraint is of "$\geq$" type, then solve problem P3 (see figure 5), if it is of the "$\leq$" type, then solve problem P4 (see figure 6). By the fact that each objective function is concerned with the optimization of one and only one variable the BLP problem always deals with a two dimensional decision space.
One decision variable is either a slack or a surplus type but the second decision variable belongs to the set of the original decision variables of the LIP problem. Since X_s and X_{sr} are restricted to the range $(-\infty,0]$ and X_j is required to be nonnegative, Z_1 and Z_2 can accept nonnegative and non-positive values, respectively. Appendix A provides sufficient information to understand why X_s and X_{sr} belong to the range $(-\infty,0]$.

P3: **Maximize** $Z_1 = X_t$ (11)

Maximize $Z_2 = X_{sr}$ (12)

S.t:

$$a_{it}X_t - X_{sr} = b_i - \sum_{\substack{j=1\\ j\neq t}}^{n} a_{ij}X'_{0j} \quad i=p,\ \forall\ p=1,2,3,\ldots,r \tag{13}$$

$$a_{it}X_t \geq b_i - \sum_{\substack{j=1\\ j\neq t}}^{n} a_{ij}X'_{0j} \quad \forall\ i=1,\ldots,r,\ r\neq p \tag{14}$$

$$a_{it}X_t \leq b_i - \sum_{\substack{j=1\\ j\neq t}}^{n} a_{ij}X'_{0j} \quad \forall\ i=r+1,\ldots,m \tag{15}$$

$$X_t \geq 0,\ X_{sr} \leq 0 \text{ and integer, } j=1,2,\ldots,n \tag{16}$$

Fig. 5. BLP: Maximization type problem- Problem 3

A sample example presented in appendix B demonstrates how BLP works and why slack and surplus variables must be defined over the range $(-\infty,0]$.

Step 4: The algorithmic form of this step is presented in figure 7.

Two points need to be defined before proceeding to the details of algorithm Leave. The last feasible point which algorithm found before its return into the infeasible region will be referred to as the "leaving feasible point". On the other hand, the last infeasible point which algorithm found before its return into the feasible region will be referred to as the "infeasible base point".

P4: ***Minimize*** $Z_1 = X_t$ (17)

Minimize $Z_2 = X_s$ (18)

S.t.:

$$a_{it}X_t \geq b_i - \sum_{\substack{j=1\\ j\neq t}}^{n} a_{ij}X'_{0j} \quad \forall\ i=1,2,3,\ldots,r \tag{19}$$

$$a_{it}X_t + X_s = b_i - \sum_{\substack{j=1\\ j\neq t}}^{n} a_{ij}X'_{0j} \quad i=p,\ r+1< p < m \tag{20}$$

$$a_{it}X_t \leq b_i - \sum_{\substack{j=1\\ j\neq t}}^{n} a_{ij}X'_{0j} \quad \forall\ i=r+1,\ldots,m,\ i\neq p \tag{21}$$

$$X_i \geq 0,\ X_s < 0 \text{ and integer, } j=1,2,\ldots,n \tag{22}$$

Fig. 6. BLP: Minimization type problem- Problem P4

```
Begin
  1. If no solution (NSP) found in step 3.
     If (k ≤ n) then
     k:= k+1
     go to step 2
     end
     if ( k > n ) then terminate and start with a new
     initial solution point (NISP)
with a new initial solution point (NISP).
  2. If a solution point found in step 3 accept the
     new solution
     If (NSP = feasible) then terminate,
     If (k ≤ n) & (NSP = feasible) then k=k+1
     and go to step 2.
     If (k > n) & (NSP = infeasible) &
     (NSP<>NISP) then go to step 1.
     If (k > n) & (NSP =NISP) then terminate and
     start with a new initial solution point.
End
```

Fig. 7. An Algorithmic form for Step 4 of Algorithm Feasible

4.3. Algorithm Leave

The main purpose of this algorithm is to add extra power to the search routine. Hence, it is used for the purpose of leaving the region of feasibility for improving the optimal value of the objective function. There exists many cases in which the structure of the mathematical model constraints the pattern of movement toward the optimality.

This, consequently, will render the solution process unsuccessful. To eliminate such an undesirable situation, we have included algorithm Leave for increasing the flexibility of this approach and expanding the domain of the operations. This algorithm

chooses the variable with the largest coefficient in the objective function and adds or subtracts one unit from the value of the selected variable, depending upon the positive or negative sign of its coefficient. This move is made regardless of whether the point is feasible. However, to determine a new feasible point based upon the "leaving feasible point", the next most promising variable, apart from those that were chosen before, will be selected. Now, the algorithm takes one or two steps in all positive or negative directions until a new feasible point which is either equivalent to or preferred to the "leaving feasible point" has been found. The evaluation process will terminate immediately after finding a new feasible point or by repeating the process n times. The summary of the steps used in algorithm **Leave** is followed by the definition of key variables employed in this algorithm. The notations used are:
SS = Maximum step size
S = the length of the step size is set to 1 or -1
L = Direction in which the variable moves
W = Variable taking values from 1 to n

This algorithm is comprised of fourteen steps:
Step1: Set SS=1, if decision variables are 0-1 type. Set SS=2, otherwise. Set Δ =1 and k=1.
Step2: Set w=1. While k < n go to step 3; otherwise go to step 14.
Step3: Choose the kth element of vector R and call it X_t
Step4: If the coefficient of X_t in the objective function is positive, set $X_t = X_t + \Delta$
Step5: While $w \le n$ go to step 6. Otherwise, go to step 13.
Step6: For $w \ne t$, set S=1, L=0, Δ=1, POINT = "infeasible", and go to step 7. For w=1, go to step 12.
Step7: While L < 2 go to step 8. Otherwise go to step 12.
Step8: While POINT ="infeasible" and $S \le SS$ go to step 9; otherwise go to step 11.
Step9: $X_w = X_w + S * \Delta$
Step10: If the new point is feasible and the value of objective function has been improved or it is the same then terminate; otherwise set $X_w = X_w - S * \Delta$, S = S+1, and go to step 8.
Step11: Set L =L+1, $\Delta = -1$, S = 1, POINT= "Infeasible", and go to step 7.
Step12: Set w = w +1 and go to step 5.
Step13: Set k = k + 1 and $\Delta = 1$. If the coefficient of X_t is positive then set $X_t = X_t - \Delta$, otherwise, $X_t = X_t + \Delta$ and go to step 2.
Step14; Terminate with no improvement.

4.4. Algorithm Backtrack

This algorithm begins its task whenever algorithm Improve and algorithm Leave is incapable of improving the current solution. This means that no feasible solution can be found by using algorithm Improve or Leave which can improve the value of the objective function. Thus, algorithm Backtrack is designed to improve the current solution when algorithm Improve and Leave can no longer produce an improved feasible solution. When this algorithm is selected then the solution process will start from the final feasible solution obtained by the last algorithm.

Example

To show the significant role of algorithms Leave and Backtrack consider an example with the feasible region shown by figure 8. For simplification purposes, it is assumed that the optimum solution point occurs at point C and that the initial infeasible solution points X (starting Point 1) and Y (starting point 2) are provided by the DM one at the time. Path X starts from initial infeasible point X and ends at the optimal solution point C using algorithms Feasible**,** Improve**,** and Leave**.** On the other hand, path Y starts from initial infeasible point Y and ends at the optimal solution point C using algorithms Feasible**,** Improve**,** Backtrack and Leave.
It is obvious enough that decision maker can find the optimal or near optimum solution by a computer aided decision making engine reinforced with such computer algorithms.

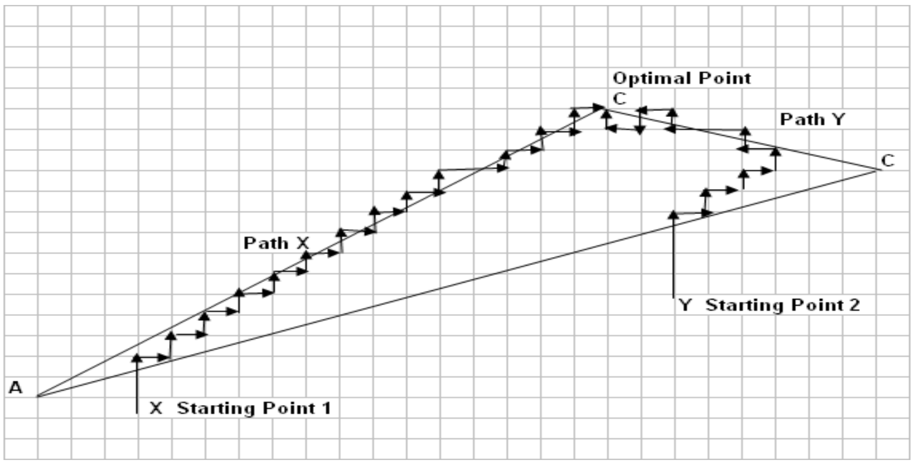


Fig. 8. Presentation of paths X and Y for initial points X and Y

5. Computational Experiment

A wide class of general integer problems has been chosen from the literature to test this interactive heuristic approach. The test problems are chosen from the Trauth and Woolsey [7] categories A, B, and D, respectively. In terms of problem size (number of variables * number of constraints), the fixed charge problems ranged from 5*4 to 12*10 and the IBM test problems ranged from 7*3 to 15*50. However, two problems from category A, ten from category B, and three from category D were chosen and solved via this computer program. A transportation type problem was successfully solved by this computer program that was of size 24*35. The computer program is capable of finding the optimal solution in most of the cases. By changing the starting point, the program has usually shown a better performance. In some cases, it produced alternative integer solutions which were not mentioned by Trauth and Woolsey [7].

Optimal and near optimal solutions to test problems were quickly found by this procedure, whereas Trauth and Woolsey [7] reported that some of their solutions were obtained only after thousands of iterations and hundreds of seconds. Computer results for our 18 test problems are provided and demonstrated in Table 1. This table provides the optimal solution as well as the heuristic solution on the selected test problems.

Our investigation indicates that an optimal solution for problem 10 and 12 can be obtained when new starting points have been chosen. Interestingly enough, by changing the starting point of problem 12 the computer algorithm has been successful to search for an optimal solution as well as alternative optimal solution point. These observations are demonstrated in table 2.

Tab. 1. Comparison of the Optimal and Heuristic Solutions of Test Problems

Problem Number	Alg.A Time (Sec.)	Alg.B Time (Sec.)	Alg.C Time (Sec.)	Alg.D Time (Sec.)	Total Time (Sec.)	Heuristic Solution Value	Optimal Solution Value
1	0.04				0.04	38	38
2	0.08	0.03	0.03		0.14	20	20
3	0.08			0.42	0.50	25	25
4		0.09			0.09	50	50
5		0.09		0.18	0.27	52	52
6	0.08				0.08	7	7
7	0.05				0.05	8	8
8	0.08				0.08	10	10
9	0.25		0.34	0.09	0.68	8	8
10	0.10				0.10	75	76
11	0.07				0.07	106	106
12	0.06				0.06	75	76
13	0.08				0.08	106	106
14	0.46		0.37	0.32	1.15	9	9
15	0.42				0.42	17	17
16	0.44	0.05	0.14		0.63	8	8
17	0.07				0.07	7	7
18	0.08	0.03			0.11	187	187

Alg.**A** = Algorithm **Improve** Alg.**B** = Algorithm **Feasible**
Alg.**C** = Algorithm **Leave** Alg.**D**. =Algorithm **Backtrack**

Tab. 2. Comparison of the optimal solution points based upon the selected starting points

Problem number	Starting Point	Ending Point	Approximate Solution	Optimal Solution
10	(0,0,0,0,0)	(0,0,0,0,75)	75	76
	(1,1,1,1,1)	(1,1,24,52,0)	76	76
12	(0,0,0,0,0)	(0,0,0,0,75)	75	76
	(1,1,20,50,2)	(1,1,24,52,0)	76	76
		(1,1,23,52,0)	76	76

6. An Evaluation of the Proposed Procedure

The proposed heuristic approach is not limited to a special class of the LIP problems. Problems with the binary decision variables can also be solved. In this case, user should provide an appropriate initial integer solution, whether feasible or infeasible, as the type of the problem requires.

This approach is reliable for the solutions bounded by the inequality type constraints. Equality type constraints should be transformed into two inequalities. For instance, any transportation type problem can be solved if equality constraints transforms into two inequality constraints. The 0-1 type problems can be handled easily with no additional effort. In this case,

the user should provide a correct response to a prompted question concerning the 0-1 type problems. This algorithm is not sensitive to the sign of C, A, and the right hand side values. This means that different classes of LIP problems can be handled by the proposed heuristic approach. It should be noticed that most of the previously developed approaches were merely devoted to a special class of the LIP problems.
This approach requires no additional preparation relative to the linear heuristic approaches that were previously addressed in the literature [2, 3, 4, 5, 6, 7, 8]. No data adjustment is required for data processing and file manipulation.

7. Conclusion

The fact that the overall approach consists of four algorithms offers a great deal of flexibility. For example, algorithm **Leave** takes different steps if problem is a 0-1 type. There is no need to apply algorithm **Feasible** when initial point is chosen to be a feasible solution point. Nevertheless, the procedure described in this paper has several disadvantages and limitations as well as certain advantages. One of the disadvantages is that the final solution depends upon the initial starting point when it is infeasible. Test problems 10 and 12 are provided for the purpose of demonstrating this difficulty.
The second disadvantage is that we cannot guarantee that this method is the most efficient in the sense of being an interactive technique for obtaining an approximate solution. This is a problem common to all interactive algorithms sited in the literature which requires the DM's familiarities with the problem.
There are several advantages to our interactive heuristics approach, however. First, it can begin with an initial solution which can be either feasible or infeasible. Second, the decision variables can be binary or a mixture of 0-1 type and other integer values. Third, variables can be binary or a mixture of 0-1 and other integer values. Third, it can be executed on the IBM-PC and compatibles. It is believed, however, that this heuristic approach is efficient because of its powerful search procedure and because it allows DM freely evaluate the final solution by feeding an initial infeasible solution point to the system when the selection of a feasible starting point becomes a tedious task. Our computational experimentation from the selected test problems shows promising results.
Since this methodology works as indicated by our computational experimentations it makes a significant contribution to the optimization of the real world LIP models that requires an approximate solution of the problem.

References

[1] Echols, R.E., Cooper, L., *"Solution of Integer Linear Programming Problems by Direct Search*", Journal of the Association of Computing Machinery, Vol. 15, 1986, pp.75-84.

[2] Cote, G., Laughth, M.A., *"Large Scaled Mixed Integer Programming: Benders Type Heuristics*", European Journal of Operational research, Vol. 16, 1986, pp. 327-333.

[3] Kochenberger, G.A., McCarl, A., Wyman, A., *"A Heuristic for General Linear Programming"*, Decision Science, Vol. 5, 1974, pp.36-44.

[4] Magazine, M. Z., Oguz, O., "*A Heuristic Algorithm for the multidimensional 0-1 Knapsack Problem*", European Journal of Operational Research, Vol. 16, 1984, 319-326.

[5] Senju, S., Toyoda, Y., "*An Approach to Linear Programming with 0-1 Variables*", Management Science, Vol. 15, 1986, B195-B205.

[6] Toyoda, Y., *"A Simplified Algorithm for Obtaining Approximate Solutions to 0-1 Programming Problems*", Management Science, Vol. 21, 1975, pp.1417-1427.

[7] Trauth, C. A., Woolsey, R.E., *"Integer Linear Programming: A Study in Computational Efficiency",* Management Science, Vol.15, 1969, pp.481-493.

[8] Balas, E., Martin, C.H., "*Pivot and Complement-A Heuristic for 0-1 Programming*", Management Science, Vol.26, 1069, pp.86-96.

[9] Sadagopab, S., Ravindran, A., *"Interactive Solution of Bicriteria Mathematical Programming*", Naval Research Logistics Quarterly, Vol.29, 1982, pp.443-459.

[10] Adulbahan, P., Tabucanon, M., "*Bicriterion Linear Programming"*, Computer and Operations Research, Vol.4, 1977, pp.147-153.

[11] Lokketangen, A., Glover, F., "*Solving Zero-One Mixed Integer Programming Problems using Tabu Search*", European journal of operational research 106, 1998, pp.624-658.

[12] Wilbaut, C., Hanafi, S., *"New Convergent Heuristics for 0-1 Mixed Integer Programming*", European journal of operational research, 2008.

[13] Kodayif, Kandemir, Vijaykrishman, N., Irwin, M.J., *"An Integer Programming Based Tool for Wireless Sensor Networks*", Journal of Parallel and distributed computing 65, 2005, pp.247-260.

[14] Canakgoz, N.A., Beasley, J.E., "*Mixed Integer Programming Approaches for Index Tracking and Enhancing Indexation"*, European Journal of Operations Research, 2008.

[15] Boland, Natashia, Hughes, B.D., Merlot, L.T.G., Stuckey, P.J., "*New Integer Linear Programming Approaches for Course Timetabling*", Computers and Operations Research, 35, 2008, pp.2209-2233.

[16] Gnanendran, K., J.K. Ho, Sundarraj, R.P., *"Stock Selection Heuristics for Interdependent Items",* European Journal of Operational Research, Volume 145, Issue 3, 16 March 2003, pp. 585-605.

[17] Roslöf, J., Iiro Harjunkoski, Tapio Westerlund, Johnny Isaksson, "*Solving a Large-Scale Industrial Scheduling*

Problem using MILP Combined with a Heuristic Procedure", European Journal of Operational Research, Vol. 138, Issue 1, 1 April 2002, pp. 29-42.

[18] Vassilev, V., Krassimira Genova, *An algorithm of internal feasible directions for linear integer programming",* European Journal of Operational Research, Vol. 52, Issue 2, 27 May 1991, pp 203-214.

[19] Boehning, R.L., Ralph M., Butler, Billy E., Gillett, *"A Parallel Integer Linear Programming Algorithm"* European Journal of Operational Research, Vol. 34, Issue 3, March 1988, pp. 393-398.

[20] Benders, J.F., J.A. E. E. van Nunen, J.F., Benders, J.A.E. E., van Nunen, "*A Property of Assignment Type Mixed Integer Linear Programming Problems*", European Journal of Operational Research 139, May 2003.

[21] Brosh, I., Marvin Hersh, Eliezer Shlifer, *"A Mixed Integer Programming and Heuristic Algorithm for a Warehouses Location Problem"*, European Journal of Operational Research, 139, June 2003

Appendix A:

The Mathematics of Exchanging one Point with Another Point

This section describes the introductory concept of slack and surplus variables associated with a pair of inequality constraints. The emphasis is on the possible acceptable range of slack and surplus variables for exchanging one point with another point. Consider the following pair of inequality constraints:

$$V_1^{'}X \le b_1 \quad (23)$$

$$V_2^{'}X \ge b_2 \quad (24)$$

These types of constraints can be changed into the following type of equalities:

$$V_1^{'}X + X_s = b_1 \quad (25)$$

$$V_2^{'}X - X_{sr} = b_2 \quad (26)$$

where X_s and X_{sr} stand for the slack and surplus variables, respectively. Let us consider figure 8 where line $V^{'}X = b$ has divided the first quadrant into two sections S_1 and S_2. If one wishes to keep the equality sign for any point from region S_1, say X_{q_1}, then X_s must accept a value of

$$X_s = b - V'X_{q_1} > 0 \quad (27)$$

Similarly, if one chooses point X_{q_2} from region S_2 and demands to keep the equality sign then X_{sr} should accept a value of

$$X_{sr} = V'X_{q_2} - b > 0 \quad (28)$$

Therefore, to move between integer points of regions S_1 or S_2 and within the feasible regions S_1 and S_2 the following set of linear constraints need to be employed (29-30 and 31-32). However, the following two cases can clearly be realized:

1. Move from the region S_1 to region S_2
2. Move from region S_2 to region S_1

To move from region S_1 to S_2 the following set of constraints should be employed:

$$V'X + X_s = b \quad (29)$$

$$X_s \le 0 \text{ or } X_s \in (-\infty, 0] \quad (30)$$

To move from region S_2 to S_1 the following set of constraints can be utilized to mathematically exchange an integer point from S_2 into its equivalent integer point from S_1:

$$V'X - X_{sr} = b \quad (31)$$

$$X_{sr} \le 0 \text{ or } X_{sr} \in (-\infty, 0] \quad (32)$$

The proposed discussion can be easily applied to the solution process of the LIP problem where the initial solution is infeasible. The initial starting point can be chosen either arbitrarily or based upon the DM's knowledge of the problem.

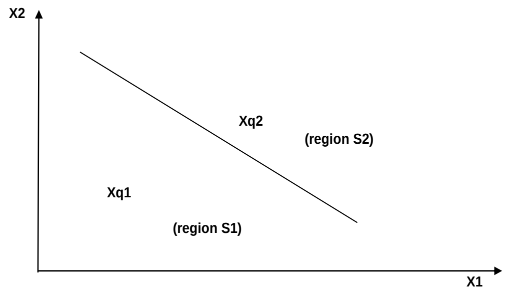


Fig. 9. Exchange of two integer points with one another

Appendix B

Numerical Example

In this section we illustrate the feature of the BLP by a simple example problem. Consider the following simple example where X_1 and X_2 are considered to be integer valued.

Maximize $F(X_1, X_2) = X_1 + X_2$

S.t.

$$G_1(X_1, X_2) = X_2 \le 3$$

$$G_2(X_1, X_2) = 2X_1 + X_2 \le 6 \qquad X_1 \ge 0 \quad X_2 \ge 0$$

As a starting point, choose point A = (3, 3). As it turns out this point does not satisfy the second constraint, because $G_2(X_1, X_2) = 9 > 6$. Since the violated constraint is of "$\le$"type, we need to find a compromise solution for the following BLP problem:

Minimize $Z_1 = X_2$

Minimize $Z_2 = X_s$

S.t.

$$X_2 + X_s = 6 - 2X_0^{'} \qquad X_2 \le 3 \quad X_2 \ge 0 \quad X_s \le 0$$

Equivalently, we need to solve the following BLP problem:

Minimize $Z_1 = X_2$
Minimize $Z_2 = X_s$

S.t.

$$X_2 + X_s = 0 \qquad \begin{aligned} X_2 &\le 3 \\ X_2 &\ge 0 \\ X_s &\le 0 \end{aligned}$$

We notice that the feasible region is a half plane $X_2 + X_s = 0$ with the bounded variables $X_2 \ge 0$ and $X_s \le 0$. All possible integer solution points for the BLP problem are summarized in table 3.

Tab. 3. Possible Integer Solution Points for the BLP

$Z_1 = X_2$	$Z_2=X_s$	$\lvert Z_1.Z_2 \rvert$
0	0	0
1	-1	1
2	-2	4
3	-3	9

Therefore $U_2(Z_1, Z_2)$ = Min {0, 1, 4, 9} = 0. The compromise solution point $X_c^* = (X_2, X_s) = (0,0)^t$ is equivalent to the feasible point (X_1, X_2) = (3, 0). In a similar manner, however, we can find feasible solution points for the selected infeasible solution points C=(1,4), and D = (6, 2).