International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

An Improved CPU Time in Triangle Splitting Method for Solving a Biobjective Mixed Integer Program

Ali Al-Hasani
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia
Department of Mathematics, College of Sciences
University of Basrah, Al- Basrah, Iraq

Masar Al-Rabeeah
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia
Department of Mathematics, College of Sciences
University of Basrah, Al- Basrah, Iraq

Santosh Kumar
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia
Department of Mathematics and Statistics
University of Melbourne, Melbourne, Australia
Corresponding author: santosh.kumarau@gmail.com

Andrew Eberhard
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia

(Received February 8, 2018; Accepted April 2, 2018)

Abstract
In this paper, an existing algorithm known as Triangle Splitting Method (TSM) for the Bi-Objective Mixed Integer Program (BOMIP) has been modified, which has been named "An Improved Triangle Splitting Method (ITSM)". The TSM solves many unnecessary single objectives Mixed Integer Programs (MIP) to split each triangle with two rectangles, and second, it doesn't find the all efficient frontiers. The proposed ITSM has resulted in redaction of CPU time by solving one MIP at each triangle and finds more number of nondominated frontiers compared to the TSM. The proposed modification has been tested in many instances.

Keywords- Biobjective mixed integer programming, Nondominated frontiers, Exact algorithm for multi-objective optimization.

1. Introduction

A Mixed Integer Programming (MIP) problem arises in many situations when some decision variables are restricted to integer values. Branch and Bound (B & B) method is a commonly used approach, that has been used to solve MIP models, for example, see Davis et al. (1971), where a single objective problem has been solved by the B & B. Mavrotas and Diakoulaki (1998, 2005) have applied (B & B) to solve a multi objective mixed integer programming problem, where they have considered three and four objectives with limited successes. Vincent et al. (2013) have corrected and improved the algorithm by Mavrotas and Diakoulaki (1998, 2005) found some nondominated points that were missed out earlier.

Boland et al. (2015) have developed a Triangle Splitting Method (TSM) to find nondominated frontiers for the BOMIP in the criterion space, which was missing some nondominated points, as has been pointed out in this paper. TSM initially depends on splitting the solution region into rectangles and apply a method of to find locally extreme supported nondominated points by a weighted sum method. At each stage of the division, one detects the nondominated frontiers and isolates points.

Soylu and Yildiz (2016) have developed an exact algorithm to find all Pareto (nondominated) frontiers of BOMIP; their algorithm is a combination of ε −constraint algorithm (Ehrgott and Ruzika, 2008) and the enhancement of Tabu constraint method (Fischetti and Lodi, 2003). They compared their algorithm with the TSM and found in small instances they were better, but TSM was better for larger instances.

There are two challenges in determination of the nondominated frontier set for the BOMIP.
- It contains supported and unsupported nondominated points.
- Existence of continuous variables which form line segment for continuous variables considered more challenging to determine the efficient frontier for BOMIP problems.

By the proposed modification of the TSM, we found that the CPU time was reduced and more nondominated frontiers were identified.

The paper has been organized in 6 sections. In Section 2, we present some basic definitions required for the rest of the paper. The TSM has been briefly explained in Section 3. The modified approach has been discussed in Section 4 .The computational results are presented in the Section 5. Finally, a few concluding remarks are given in Section 6.

2. Basic Definitions
The BOMIP problem is defined as:

$$Min\, Z(x) := \left(Z_1(x), Z_2(x)\right) \qquad (\boldsymbol{P})$$

Subject to $x \in X$, where $X = \{\mathbb{R}^m \times \mathbb{Z}^n : Ax \le b; x \ge 0\}$ and $m \ge 1$ and $n \ge 1$ is the all feasible solution of (**P**), which it is called decision space,

where $Z_1(x) = \sum_{i=1}^{n} C_i^1 \, x_i + \sum_{j=1}^{m} C_j^1 \, x_j$, $Z_2(x) = \sum_{i=1}^{n} C_i^2 \, x_i + \sum_{j=1}^{m} C_j^2 \, x_j$, such that C_i^1 and C_i^2 are the coefficient matrix for the integer variables and C_j^1 and C_j^2 are the coefficient matrix for the continuous variables,

When $m = 0$ and $n \ge 1$ the problem turns into a Bi-objective integer program, and when $n = 0$ and $m \ge 1$ the problem turns into a Bi-objective linear program. The image X of the decision space is called criterion space or objective space. Details of theses definitions can be found in Steuer (1986) and Greco et al. (2005).

International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

Definition 1. A feasible solution $\acute{x} \in X$ is called efficient or (Pareto) solution, if there is no other $x \in X$ such that $Z(\acute{x}) \leq Z(x)$. The set of all efficient solutions $\acute{x} \in X$ is denoted X_E and called the efficient solution set.

Definition 2. If $\acute{x}$ is efficient solution, $Z(\acute{x})$ is called non-dominated (Pareto) point, and the set of all non-dominated points $Z(\acute{x}) \in Y$ is called the non-dominated points set or nondominated (Pareto) frontier set and denoted Y_{ND}.

Definition 3. The efficient solution $\acute{x}$ is a supported efficient solution for problem (**P**), if there is some $\lambda \in \mathbb{R}^n_{>}$ such that $\acute{x} \in X_E$ is the optimal solution of the following single objective weighted-sum problem.

$$Min_{x \in X}(\lambda^T Z(x))$$

and $Z(\acute{x})$ is called the supported nondominated point.

Definition 4. If x is a supported efficient solution and $y = Z(x)$ is an extreme point of $convY$, then x is called an extreme supported efficient solution and y is an extreme nondominated point.

3. The Triangle Splitting Method

The TSM for solving bi-objective mixed integer programming problems was developed by Boland et al. (2015), which is an efficient method for finding a nondominated frontier in the criterion space. Aneja and Nair (1979) proposed the Weighted Sum Method (WSM) which has been extensively applied for solving Bi-Objective Integer Programming (BOIP) models. The TSM depends on WSM to find extreme supported nondominated points. Main step of TSM are as follows:

Step 1. Construct an initial rectangle with two bound points.

Step 2. Find all extreme nondominated points in the rectangle by WSM.

Step 3. Triangles are constructed with two adjacent extreme nondominated points.

Step 4. Check that the hypotenuse of the triangle is a nondominated frontier, i.e., the two points of the hypotenuse are nondominated points.

Step 4.1 If hypotenuse of triangle is efficient, select the next triangle and go to step2.

Step 4.2 If not, splitting points are detected by solving the lexicographic minimize problem and the triangle is split into two rectangles, or isolated points and rectangle. Go to step 2.

Unfortunately, this algorithm has a few disadvantages in practice.

- Inside of triangle, it solves unnecessary MIPs to detect the splitting point which are used to split the triangle with rectangles. In this possess, many splitting points are detected, and thus it solves a number of unnecessary MIPs.

- This method does not find all nondominated frontiers of BOMIP, see Appendix 2, where new points have been shown in **bold face**.

These disadvantages have been discussed in Section 4.

4. An Improved Algorithm for Triangle Splitting Method

4.1 Proposed Improvement in the TSM

In this section, we develop an Improved Triangle Splitting Method (ITSM) for solving BOMIP models. The main steps of the proposed approach are similar to the TSM. However, detection of the splitting points of triangle is different from the lexicographic approach used in the TSM. Improved algorithm uses the constrained weight sum of objective functions to detect the splitting points.

The above modification reduces requirement for solving a large number of MIPs as required by the TSM. Note that in the TSM approach, when the $lexmin\ (z1;\ z2)$ functions are solved, one has to solve two MIP to find a split point, whereas in the proposed ITSM, when the $con\_weight\ (z1;\ z2)$ function is used, only one MIP is required for a splitting point.

For more details, we introduce concepts and notation that will facilitate the presentation and discussion of the ITSM.

Let $z^1 = (z_1^1, z_2^1)$ and $z^2 = (z_1^2, z_2^2)$ be two points in the criterion space with $z_1^1 \le z_1^2$ and $z_2^2 \le z_2^1$. We denote by $R(z^1, z^2)$ the rectangle in the criterion space defined by the points z^1 and z^2. Furthermore, we denote by $T(z^1, z^2)$ the right triangle in the criterion space defined by the points $z^1, (z_1^2, z_2^1)$ and z^2. Finally, we denote by $H(z^1, z^2)$ the line segment in the criterion space defined by the points z^1 and z^2, i.e., the hypotenuse of triangle $T(z^1, z^2)$.

Then, lexicographic approach of the TSM can be stated as follows:

$$\bar{z}^1 = \underset{x \in X}{\text{lexmin}}\{z_1(x), z_2(x) : z(x) \in T(z^1, z^2)\} \tag{1}$$

$$\bar{z}^2 = \underset{x \in X}{\text{lexmin}}\{z_2(x), z_1(x) : z(x) \in T(z^1, z^2)\} \tag{2}$$

where $\bar{z}^1$ and $\bar{z}^2$ is splitting points of triangle, X is feasible set.

Next, for the improved algorithm, constrained weight sum of objective functions are defined as follows:

$$\bar{z}^1 = \min_{x \in X}\left\{z_1(x) + \frac{1}{\frac{UB_2 - LB_2}{Epsilon} + 1} z_2(x) \,\middle|\, z(x) \in T(z^1, z^2)\right\} \tag{3}$$

$$\bar{z}^2 = \min_{x \in X}\left\{\frac{1}{\frac{UB_1 - LB_1}{Epsilon} + 1} z_1(x) + z_2(x) \,\middle|\, z(x) \in T(z^1, z^2)\right\} \tag{4}$$

where, UB_1, LB_1, UB_2, LB_2 are upper bounds and lower bounds of first and second objective functions, respectively and "$Epsilon$" is constant defined by 10^{-5}.

Proposition. For two objective function $z_1(x)$ and $z_2(x)$, the solution of two minimizing problems (1) and (3) are equivalent, i.e,

$$\min_{x\in X}\left\{z_1+\frac{1}{\frac{UB_2-LB_2}{Epsilon}+1}z_2\middle|z(x)\in T(z^1,z^2)\right\}=\ \operatorname*{lexmin}_{x\in X}\{z_1,\ z_2|z(x)\in T(z^1,z^2)\},$$

and the solution of two minimizing problems (2) and (4) are also equivalent, i.e.

$$\min_{x\in X}\left\{\frac{1}{\frac{UB_1-LB_1}{Epsilon}+1}z_1+\ z_2\middle|z(x)\in T(z^1,z^2)\right\}=\ \operatorname*{lexmin}_{x\in X}\{z_2,\ z_1|z(x)\in T(z^1,z^2)\}.$$

To analyze the geometrical meaning of equations (1) and (3), see Figs. 1 and 2.

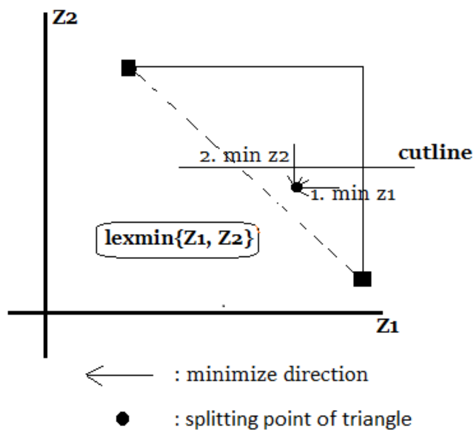

Fig. 1. Geometrical analysis of lexmin{} as used in the TSM

As shown in Fig. 1, problem (1) solves two MIPs to detect the splitting point and focus on the optimization of first objective function, then minimize the second objective function.

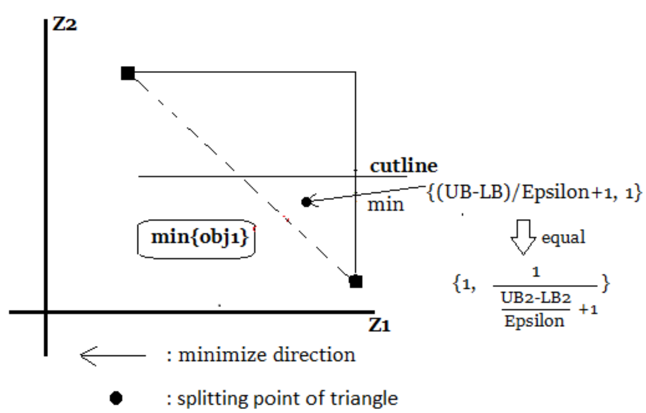

Fig. 2. Geometrical analysis of the equation (3) of the proposed improved algorithm

As shown in Fig. 2, problem (3) focus on the optimization of first objective function, simultaneously, it also minimizes the second objective function with a very small weight, which results in solving only one problem.

The Improved Triangle Splitting Method details are given as Algorithm 1.

Algorithm 1.(Improved Algorithm)
Initialize

1. $L_{ndp} \leftarrow \emptyset$; $L_E \leftarrow \emptyset$; $L_{shape} \leftarrow \emptyset$
2. $Z_{LR} \leftarrow$ solve(weight * f1 + f2)
   $Z_{UL} \leftarrow$ solve(f1 + weight * f2)
3. PUSH(L_{ndp}, (Z_{LR}, 0))
   PUSH(L_{ndp}, (Z_{UL}, 0))
   PUSH(Q, (Z_{UL}, Z_{LR}, ‘R’, ‘H’))

Main Loop

4. **While** L_{shape} is not empty **do**
5. temp ← POP(L_{shape})
6. If shape is Rectangle then
7. $L_E \leftarrow$ Find_ENDP(shape)
8. PUSH(L_{shape}, (E_1,E_2,‘T’,‘H’))
9. If shape is triangle then
10. connection = line_detect(shape)
11. If connection is 1 then
12. Update_NDP(E_1)
13. Else
14. If direction is horizontal then
15. Z_1, Z_2 ← Split_triangle(E_1,E_2,‘T’,‘H’)
16. If Z_1 is equal to Z_2 then
17. PUSH(L_{shape}, (Z_1,E_2,‘R’,‘V’))
    PUSH(L_{shape}, (E_1,Z_2,‘R’,‘V’))
    PUSH(L_{ndp}, (Z_1, 0))
18. If Z_1 isn’t equal E_2 then
19. PUSH(L_{shape}, (Z_1,E_2,‘R’,‘V’))
    PUSH(L_{ndp}, (Z_1, 0))
20. If Z_2 isn’t equal E_1 then
21. PUSH(L_{shape}, (E_1,Z_2,‘R’,‘V’))
    PUSH(L_{ndp}, (Z_2, 0))
22. Else
23. Z_1, Z_2 ← Split_triangle(E_1,E_2,‘T’,‘V’)
24. If Z_1 is equal to Z_2 then
25. PUSH(L_{shape}, (Z_1,E_2,‘R’,‘H’))
    PUSH(L_{shape}, (E_1,Z_2,‘R’,‘H’))
    PUSH(L_{ndp}, (Z_1, 0))
26. If Z_1 isn’t equal E_2 then
27. PUSH(L_{shape}, (Z_1, E_2,‘R’,‘H’))
    PUSH(L_{ndp}, (Z_1, 0))
28. If Z_2 isn’t equal E_1 then
29. PUSH(L_{shape}, (E_1,Z_2,‘R’,‘H’))
    PUSH(L_{ndp}, (Z_2, 0))
30. Return L_{ndp}

International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

5. Experimental Results

5.1 An Illustrative Example

To illustrate the efficiency of the improved algorithm for the TSM, we explain steps of the ITSM by solving an example of a biobjective 0-1 mixed integer program from class size 20 variables (10 variables are real and the remaining 10 are binary variables). The details of this example are given in Appendix 1.

The Fig. 3 illustrates the steps of the ITSM. The Table 1 show the ordered list of nondominated points, which have been identified by the improved algorithm also indicates if they are connected or not. The final results of the example after several stages of splitting are given in Table 2.

It may be noted that that to keep the comparison fair, we have solved the same example by the TSM and the ITSM and noted that:

- The CPU time was less for the ITSM compared to the TSM, where the same numbers of nondominated frontiers were detected. See Table 3, row 1 for example.
- The number of MIP's solved are also less for the ITSM compared to the number required by the TSM, see Table 3, row 1.

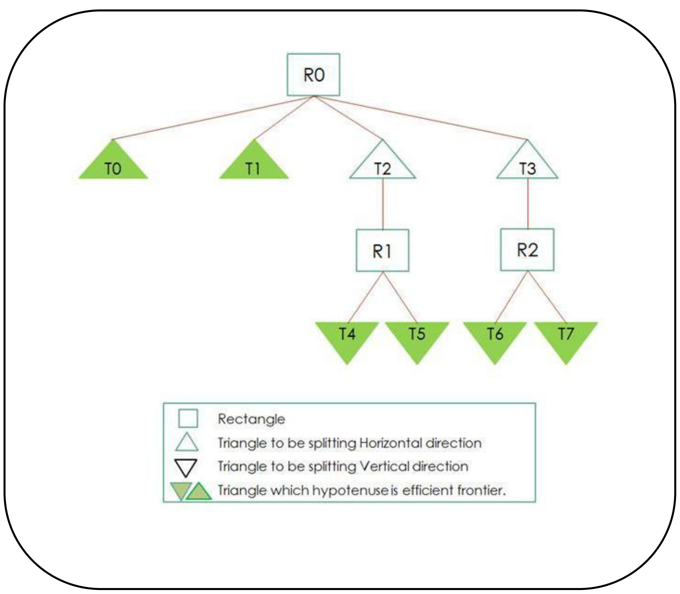

Fig. 3. Search tree of the example

International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

Table 1. Detailed information for the nodes in the search tree for example 5.1

Items	z^1	z^2	Line Detection	Splitting Point	
				z^{-1}	z^{-2}
$R0$	(-230.22, -59.1006)	(-176.008, -348.822)	Yes		
$T0$	(-183.49, -342.985)	(-176.008, -348.822)	Yes		
$T1$	(-193.121, -330.601)	(-183.49, -342.985)	Yes		
$T2$	(-220.354, -251.744)	(-193.121, -330.601)	No	(-193.121, -330.601)	(-197.183, -273.639)
$T3$	(-230.22, -59.1006)	(-220.354, -251.744)	No	(-220.354, -251.744)	(-220.356,-70.5877)
$R1$	(-220.354, -251.744)	(-197.183, 273.639)	Yes		
$T4$	(-212.804, -261.454)	(-197.183, 273.639)	Yes		
$T5$	(-220.354, -251.744)	(-212.804, -261.454)	Yes		
$R2$	(-230.22, -59.1006)	(-220.356, -70.5877)	Yes		
$T6$	(-222.722, -68.7427)	(-220.356, -70.5877)	Yes		
$T7$	(-230.22, -59.1006)	(-222.722, -68.7427)	Yes		

Table 2. Set of all nondominated frontiers of example 5.1

NDP	(Z_1(x); Z_2(x))	Connected
1	(230.22, 59.1006)	1
2	(222.722, 68.7427)	1
3	(220.356, 70.5877)	1
4	(220.354, 251.744)	0
5	(212.804, 261.454)	1
6	(197.183, 273.639)	0
7	(193.121, 330.601)	1
8	(183.49, 342.985)	1
9	(176.008, 348.822)	0

5.2 Comparison of the TSM and Improved Algorithm by New Instances

Here we demonstrate the effectiveness of the improved algorithm by applying it to new instances, which were generated by us.

Both algorithms have been coded in ‘C’ by using CPLEX 12.1 (Cplex, 2010) as a solver and the instances have run on a 2.93 GHz workstation with 1 GB of RAM memory.

To demonstrate the effectiveness of the improved algorithm, we generated new instances, which have same formula instances that was used by Mavrotas and Diakoulaki (1998, 2005). The values of the coefficients of an instance are obtained as follows:

International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

- The objective function coefficients of the continuous variables are drawn uniform randomly from [-10, 10].
- The objective function coefficients of the binary variables are drawn uniform randomly from [-200,200].
- The right hand sides of the constraints are drawn uniform randomly from [50; 150].
- The matrix coefficients (for both continuous and binary variables) are drawn uniform randomly from [1; 20].

Table 3. Comparison CPU time of TSM and ITSM

Class	Triangle Slitting Method			Improved Algorithm		
	CPU (Sec.)	IP's	NDP	CPU (Sec.)	IP's	NDP
C20	0.0394	25	9	**0.0299**	**19**	9
	0.1759	65	26	**0.1595**	**52**	26
	0.1815	74	32	**0.1488**	**64**	32
	0.2376	83	27	**0.1591**	**64**	27
	0.4482	125	49	**0.3769**	**103**	49
C40	1.0421	100	41	**0.9535**	**83**	41
	1.1889	112	43	**1.0508**	**90**	43
	1.8994	162	65	**1.6642**	**136**	65
	1.3979	141	60	**1.2570**	**122**	60
	3.2377	226	93	**2.5868**	**194**	93
C80	6.8623	97	40	**5.4225**	**81**	40
	8.7242	145	63	**7.2029**	**127**	63
	15.7322	235	91	**13.2314**	**197**	91
	18.4946	230	98	**14.5634**	**200**	98
	32.0192	373	160	**25.6081**	**322**	160
C160	47.2226	196	83	**43.8132**	**171**	83
	67.1019	194	84	**58.5214**	**170**	84
	69.8460	234	103	**63.9827**	**210**	103
	83.7549	228	93	**71.2993**	**191**	93
	198.0659	271	120	**157.4936**	**241**	120
C320	243.6620	182	81	**237.1059**	**160**	81
	333.6179	231	104	**298.8252**	**207**	104
	953.7341	332	140	**874.5223**	**288**	140
	1058.3147	303	136	**1026.5948**	**275**	136
	1189.1274	518	223	**1098.2435**	**451**	223
C500	6341.1039	306	128	**6169.1787**	**261**	128
	7289.5818	326	139	**6747.2976**	**284**	139
	14784.9241	512	230	**13665.1904**	**463**	230
	15914.8459	443	194	**13381.3449**	**396**	194
	27618.9191	569	230	**24902.8913**	**487**	230

The Table 3 shows a comparison of the CPU time and number of IP's between TSM and ITSM.

6. Conclusion

In this study, we improved the TSM for finding the nondominated frontier points of BOMILPs. We have used a constrained weight sum of objective functions to detect the splitting points. The method used has created additional nondominated frontier points that were missed out by the TSM. In addition, we also demonstrated the efficiency of the ITSM over the TSM by testing several instances and applying them to larger size instances and comparing the CPU time.

It is proposed to reconsider the work of Bose and Pain (2018), Pramy (2018) in our future work on multi-objective mixed integer programming studies.

References

Aneja, Y. P., & Nair, K. P. (1979). Bicriteria transportation problem. *Management Science*, *25*(1), 73-78.

Boland, N., Charkhgard, H., & Savelsbergh, M. (2015). A criterion space search algorithm for biobjective mixed integer programming: The triangle splitting method. *INFORMS Journal on Computing*, *27*(4), 597-618.

Bose, G. K., & Pain, P. (2018). Metaheuristic approach of multi-objective optimization during EDM process. *International Journal of Mathematical, Engineering and Management Sciences*, *3*(3), 301-314.

Cplex, I. I. (2010). *12.2 User's Manual*. Book 12.2 User's Manual, Series 12.2 User's Manual.

Davis, R. E., Kendrick, D. A., & Weitzman, M. (1971). A branch-and-bound algorithm for zero-one mixed integer programming problems. *Operations Research*, *19*(4), 1036-1044.

Ehrgott, M., & Ruzika, S. (2008). Improved ε-constraint method for multiobjective programming. *Journal of Optimization Theory and Applications*, *138*(3), 375-396.

Fischetti, M., & Lodi, A. (2003). Local branching. *Mathematical Programming*, *98*(1-3), 23-47.

Greco, S., Figueira, J., & Ehrgott, M. (2005). Multiple criteria decision analysis. *Springer's International series*, (ISOR, volume 233).

Mavrotas, G., & Diakoulaki, D. (1998). A branch and bound algorithm for mixed zero-one multiple objective linear programming. *European Journal of Operational Research*, *107*(3), 530-541.

Mavrotas, G., & Diakoulaki, D. (2005). Multi-criteria branch and bound: A vector maximization algorithm for mixed 0-1 multiple objective linear programming. *Applied Mathematics and Computation*, *171*(1), 53-71.

Pramy, F. A. (2018). An approach for solving fuzzy multi-objective linear fractional programming problems. *International Journal of Mathematical, Engineering and Management Sciences*, *3*(3), 280-293.

Soylu, B., & Yıldız, G. B. (2016). An exact algorithm for biobjective mixed integer linear programming problems. *Computers & Operations Research*, *72*, 204-213.

Steuer, R. E. (1986). Multiple criteria optimization: theory, computation, and applications. *Wiley*, pp 546.

Vincent, T., Seipp, F., Ruzika, S., Przybylski, A., & Gandibleux, X. (2013). Multiple objective branch and bound for mixed 0-1 linear programming: Corrections and improvements for the biobjective case. *Computers & Operations Research*, *40*(1), 498-509.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

Appendix 1

Details of the example discussed section 5.1 are as follows:

$$Max\ Z_1(x) = \sum_{i=1}^{n} c_i^1 x_i + \sum_{i=1}^{m} c_j^1 x_j$$

$$Max\ Z_2(x) = \sum_{i=1}^{n} c_i^2 x_i + \sum_{i=1}^{m} c_j^2 x_j$$

Subject to: $Ax \leq b$

where n and m are number of continue and binary variables respectively and $x_i \geq 0$ and x_j are binary, and A is matrix size $n \times m$ and b is vector size $n + m$.

For our example:

$$\text{Max } Z_1(\text{x}) = -6.53x1 - 6.58x2 - 2.25x3 + 1.48x4 + 3.00x5 - 4.21x6 + 0.36x7 + 0.21x8 + 9.35x9 - 6.37x10 - 194.83x11 - 24.66x12 - 15.15x13 - 137.59x14 - 4.08x15 - 164.88x16 - 20.52x17 - 164.86x18 - 165.80x19 + 198.53x20]$$

$$Max\ Z_2\ (x) = +1.96\text{x}1 - 1.97\text{x}2 - 2.26\text{x}3 - 5.84\text{x}4 + 3.78\text{x}5 - 4.88\text{x}6 - 2.27\text{x}7 + 4.71\text{x}8 - 5.13\text{x}9 - 5.23\text{x}10 - 194.25\text{x}11 - 33.40\text{x}12 - 36.48\text{x}13 - 39.40\text{x}14 + 196.28\text{x}15 - 176.43\text{x}16 + 76.29\text{x}17 + 3.65\text{x}18 + 27.95\text{x}19 + 63.40\text{x}20$$

Subject to:

C01: $-0.79x1 + 0.31x2 + 5.59x3 + 19.68x4 + 14.41x5 + 3.58x6 - 0.20x7 + 11.16x8 + 19.10x9 + 8.76x10 + 3.95x11 + 4.56x12 + 6.14x13 + 13.26x14 + 13.34x15 + 10.73x16 + 14.76x17 + 11.50x18 + 14.03x19 + 1.53x20 \leq 104.70$

C02: $+2.42x1 + 13.49x2 + 11.08x3 + 18.44x4 + 2.34x5 + 9.24x6 + 1.28x7 + 16.49x8 + 6.54x9 + 18.52x10 + 16.69x11 + 7.85x12 + 4.12x13 + 16.38x14 + 2.27x15 + 8.70x16 + 17.17x17 + 14.43x18 + 7.81x19 + 5.93x20 \leq 147.13$

C03: $+13.37x1 + 13.08x2 + 12.66x3 + 6.71x4 + 3.81x5 + 7.42x6 + 19.22x7 + 18.85x8 + 9.96x9 + 9.71x10 + 1.28x11 + 3.46x12 + 0.79x13 - 0.27x14 + 6.80x15 + 11.04x16 + 2.01x17 + 3.30x18 + 18.59x19 + 0.54x20 \leq 54.77$

C04: $+6.44x1 + 5.66x2 + 17.38x3 + 9.72x4 + 15.37x5 + 14.56x6 + 4.16x7 + 3.18x8 + 0.50x9 + 3.56x10 + 17.55x11 + 14.58x12 + 17.22x13 + 4.27x14 + 19.40x15 + 4.65x16 + 3.49x17 + 18.26x18 + 15.61x19 + 14.20x20 \leq 52.58$

C05: $-0.92x1 + 16.00x2 + 0.27x3 + 6.87x4 + 7.05x5 + 3.28x6 + 11.17x7 + 5.64x8 + 4.83x9 + 12.18x10 + 13.09x11 + 11.49x12 + 9.56x13 + 2.81x14 + 6.86x15 + 4.12x16 + 7.97x17 + 11.05x18 + 5.63x19 + 12.54x20 \leq 95.76$

C06: $+0.21x1 + 9.76x2 + 13.88x3 - 0.37x4 + 15.41x5 + 18.38x6 + 18.88x7 + 11.02x8 + 12.59x9 + 19.42x10 + 11.10x11 + 8.59x12 - 0.29x13 + 18.97x14 + 16.64x15 + 3.99x16 + 10.15x17 + 2.29x18 + 9.82x19 + 2.33x20 \leq 132.77$

C07: $+1.31x1 + 12.90x2 - 0.80x3 + 9.18x4 + 18.03x5 + 8.17x6 + 0.23x7 + 3.66x8 + 0.71x9 + 9.84x10 + 4.88x11 + 11.48x12 + 3.72x13 + 5.50x14 + 6.89x15 + 2.11x16 + 4.39x17 + 18.92x18 + 15.70x19 + 3.82x20 \leq 102.51$

C08: $+4.30x1 + 4.52x2 + 9.00x3 + 0.95x4 + 9.51x5 - 0.83x6 + 4.24x7 - 0.66x8 + 2.50x9 + 0.62x10 + 1.65x11 + 16.40x12 + 0.82x13 + 11.83x14 + 14.44x15 + 10.00x16 + 13.07x17 + 19.11x18 + 11.72x19 + 2.91x20 \leq 73.79$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

C09:$+3.20x1 + 7.64x2 + 10.50x3 + 11.10x4 + 10.75x5 + 15.90x6 + 10.02x7 + 6.46x8 - 0.27x9 + 0.05x10 + 11.76x11 + 5.25x12 + 10.06x13 + 13.72x14 + 15.77x15 + 10.22x16 + 18.96x17 + 16.10x18 + 13.73x19 - 0.40x20 \leq 144.11$

C10:$+10.14x1 + 1.41x2 + 10.60x3 + 4.58x4 + 12.42x5 + 3.67x6 + 3.69x7 + 4.14x8 + 7.58x9 + 8.69x10 + 8.34x11 + 16.22x12 - 0.80x13 - 0.54x14 + 6.98x15 + 16.10x16 + 10.47x17 + 14.45x18 + 16.82x19 + 11.52x20 \leq 84.38$

C11:$+2.08x1 + 1.59x2 - 0.05x3 + 18.85x4 + 12.81x5 + 19.90x6 + 14.96x7 + 6.54x8 - 0.50x9 + 13.72x10 + 17.68x11 + 1.91x12 + 4.32x13 + 2.27x14 + 15.33x15 + 8.99x16 + 6.96x17 - 0.51x18 + 17.58x19 + 16.66x20 \leq 96.82$

C12:$+13.81x1 + 16.85x2 + 9.28x3 + 0.79x4 + 12.96x5 - 0.23x6 + 16.25x7 + 9.78x8 + 12.29x9 + 2.47x10 + 12.87x11 + 14.88x12 + 3.41x13 + 11.72x14 + 7.69x15 + 3.31x16 + 6.68x17 + 15.24x18 + 3.81x19 + 0.40x20 \leq 116.34$

C13:$+6.73x1 + 5.73x2 + 16.20x3 + 2.07x4 + 15.72x5 + 3.17x6 + 2.55x7 + 13.31x8 - 0.16x9 + 12.39x10 + 7.12x11 + 17.69x12 + 1.67x13 + 8.92x14 + 10.65x15 + 2.43x16 + 5.17x17 + 0.44x18 + 15.72x19 + 8.64x20 \leq 122.92$

C14:$+10.60x1 + 13.05x2 + 6.03x3 + 19.30x4 + 17.37x5 + 13.71x6 + 14.54x7 + 1.19x8 + 15.12x9 + 7.47x10 + 8.92x11 + 0.86x12 + 3.68x13 + 11.99x14 + 17.59x15 + 7.85x16 + 15.55x17 + 10.90x18 + 8.68x19 + 7.94x20 \leq 145.34$

C15:$+6.38x1 + 10.61x2 + 7.94x3 + 18.03x4 + 14.05x5 + 14.11x6 + 19.47x7 + 9.78x8 + 2.76x9 + 13.79x10 + 0.38x11 + 16.81x12 - 0.17x13 - 0.31x14 + 14.19x15 + 14.54x16 + 15.23x17 + 16.38x18 + 9.67x19 + 2.71x20 \leq 80.02$

C16:$+11.53x1 + 7.39x2 + 18.30x3 + 9.12x4 + 16.25x5 + 13.85x6 + 0.02x7 + 4.94x8 + 1.79x9 - 0.94x10 + 12.32x11 + 13.41x12 + 7.99x13 + 10.35x14 + 7.47x15 + 2.11x16 + 9.83x17 + 18.25x18 + 5.87x19 + 3.62x20 \leq 148.28$

C17:$+2.69x1 + 4.45x2 - 0.67x3 + 17.88x4 - 0.99x5 + 15.56x6 + 14.26x7 + 9.67x8 + 19.27x9 - 0.42x10 + 1.21x11 + 6.67x12 + 18.87x13 + 11.33x14 + 2.92x15 + 12.72x16 + 12.36x17 + 8.86x18 + 15.52x19 + 12.41x20 \leq 60.42$

C18:$+8.93x1 + 0.41x2 + 12.54x3 + 17.41x4 + 3.52x5 + 2.38x6 + 15.66x7 + 10.40x8 + 7.00x9 + 15.30x10 + 14.09x11 + 12.46x12 + 15.63x13 + 11.97x14 + 12.46x15 + 11.19x16 + 6.24x17 + 2.14x18 + 10.47x19 + 6.81x20 \leq 75.49$

C19:$+18.15x1 + 5.68x2 + 16.69x3 + 1.08x4 + 19.40x5 + 9.06x6 + 10.95x7 + 14.93x8 + 1.48x9 + 13.14x20 + 3.87x11 + 2.89x12 + 5.68x13 + 1.28x14 + 7.42x15 + 9.07x16 + 17.94x17 + 18.82x18 + 17.07x19 + 13.24x20 \leq 116.28$

C20:$+9.54x1 + 8.88x2 + 4.89x3 + 2.00x4 + 0.08x5 + 12.13x6 + 5.14x7 + 11.55x8 + 19.95x9 + 10.50x20 + 9.71x11 + 5.63x12 + 7.19x13 + 11.79x14 + 5.04x15 + 17.25x16 + 2.75x17 - 0.02x18 + 19.73x19 + 16.89x20 \leq 77.84$

Appendix 2

For a fair comparison between the TSM and the ITSM, the example by Boland et al. (2015) was reconsidered under the same computer environment and program coding developed by us. The results given in Tables 4 were obtained by us and in Table 5 were obtained by Boland et al. (2015). Note again that the number of non-dominated frontiers in Table 4 are 40 whereas in Table 5 are 31 i.e. more number of non-dominated frontiers have been identified.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 3, No. 4, 351–364, 2018
https://dx.doi.org/10.33889/IJMEMS.2018.3.4-025

Table 4. Nondominated frontier points for first instance Boland et al. (2015) solved by ITSM

NDP	(Z_1(x); Z_2(x))	Connected
1	(-417.786, -88.0714)	1
2	(-417.703, -92.743)	1
3	(-410.585, -108.217)	1
4	(-398.148, -123.612)	1
5	(-367.731, -152.37)	1
6	(-345.556, -169.667)	1
7	(-333.472, -174.5)	1
8	(-318.789, -176.598)	0
9	(-318.786, -220.071)	1
10	(-318.647, -227.882)	1
11	(-313.225, -239.67)	1
12	**(-310.647, -242.861)**	**1**
13	(-302.789, -252.587)	0
14	(-302.786, -293.071)	1
15	(-302.647, -300.882)	1
16	(-300.067, -306.492)	1
17	(-289.635, -319.405)	1
18	(-256.395, -350.832)	1
19	(-227.889, -373.067)	1
20	(-215.806, -377.9)	1
21	(-200.769, -380.048)	0
22	(-170.609, -380.052)	1
23	**(-167.979, -380.428)**	**1**
24	**(-167.066, -380.558)**	**1**
25	(-166.776, -380.599)	0
26	(-166.767, -380.608)	1
27	(-166.758, -380.616)	1
28	(-166.74, -380.633)	1
29	**(-166.704, -380.667)**	**1**
30	**(-166.631, -380.736)**	**1**
31	**(-166.486, -380.873)**	**1**
32	**(-166.154, -381.187)**	**1**
33	**(-165.35, -381.947)**	**1**
34	**(-163.345, -383.842)**	**1**
35	**(-156.082, -390.709)**	**1**
36	(-144.81, -401.367)	1
37	(-135.252, -410.403)	1
38	(-110.556, -429.667)	1
39	(-98.4722, -434.5)	1
40	**(-83.4352, -436.648)**	**0**

Table 5: Nondominated frontier points for first instance Boland et al. (2015) solved by TSM

NDP	$(Z_1(x); Z_2(x))$	Connected
1	(-417.781526, -88.307345)	1
2	(-417.702786, -92.743034)	1
3	(-410.584793, -108.216933)	1
4	(-410.569603, -108.235736)	1
5	(-398.147854, -123.612083)	1
6	(-367.731092, -152.369748)	1
7	(-345.555556, -169.666667)	1
8	(-333.472222, -174.500000)	1
9	(-318.798146, -176.596297)	0
10	(-318.782516, -220.251575)	1
11	(-318.647059, -227.882353)	1
12	(-318.640018, -227.897659)	1
13	(-313.224847, -239.669771)	1
14	(-302.788396, -252.587227)	0
15	(-302.782676, -293.242561)	1
16	(-302.647059, -300.882353)	1
17	(-300.066546, -306.492164)	1
18	(-289.634794, -319.405178)	1
19	(-256.394958, -350.831933)	1
20	(-227.888889, -373.066667)	1
21	(-227.830262, -373.090117)	1
22	(-215.805556, -377.900000)	1
23	(-200.795192, -380.044338)	0
24	(-170.633469, -380.048393)	1
25	(-166.775347, -380.599565)	0
26	(-166.773680, -380.601141)	1
27	(-144.796283, -401.379770)	1
28	(-135.252101, -410.403361)	1
29	(-110.555556, -429.666667)	1
30	(-98.472222, -434.500000)	1
31	**(-83.435255, -436.648138)**	**0**