An exact algorithm to find non-dominated facets of Tri-Objective MILPs

Seyyed Amir Babak Rasmi[1], Ali Fattahi[2], Metin Türkay[1]

Department of Industrial Engineering, Koç University, Istanbul, Turkey, 34450
{srasmi14,mturkay}@ku.edu.tr
UCLA Anderson School of Management, 110 Westwood Plaza, Los Angeles, CA 90095
ali.fattahi.1@anderson.ucla.edu

Keywords: *Tri-Objective MILP, Non-Dominated frontier (NDF), Exact solution*

1 Introduction and Literature review

Many decisions in different subjects of science, engineering, society, and environment involve more than a single decision criterion. Modeling and solving these problems have attracted significant attentions from researchers and practitioners in recent years. Specifically, the sustainability considerations are coined by the term Triple Bottom Line.

In the sustainable systems, decision makers approach problems by considering economic, environmental, and social concerns simultaneously. In fact, sustainability is a tri-objective optimization problem that is usually converted to a single objective optimization problem after a scalarization. Few papers study sustainable development with a tri-objective modeling approach [1, 5, 6].

Some papers in the literature design exact optimization methods for Tri-Objective Integer Linear Problems (TOILPs) [2, 3, 4]. To the best of our knowledge, no exact method for finding the exact non-dominated frontier (NDF) of the Tri-Objective MILP (TOMILP) problems has been presented so far. Mixed integer programming is critical for the analysis of both operational and managerial decisions. Therefore, an exact representation of the NDF is significantly important for the decision makers. In this paper, we present an algorithm for determining the exact NDF of TOMILPs.

2 Problem Definition

In this paper, we are interested in TOMILPs of the form:

$$\begin{aligned} \max \quad & z(x,y) = C_C x + C_Z y \\ \text{s.t.} \quad & A_C x + A_Z y \leq b,\ x \in \mathbb{R}^q_+, y \in \mathbb{Z}^q_+, \end{aligned} \tag{1}$$

where x is the vector of continuous decision variables, y is the vector of discrete (integer) decision variables, and C_C and C_Z are $3 \times n$ and $3 \times q$ matrices, respectively. To simplify the presentation, we sometimes denote the feasible region by $\{S(y) : y \in \mathbb{Z}^q_+\}$; therefore, fixing integer decision variables to specific integer values ($\overline{y}$) gives a sub-TOLP with the feasible region $S(\overline{y}) = \{x \in \mathbb{R}^q_+ | A_C x \leq b - A_Z \overline{y}\}$.

This paper was first published in proceedings of the 12th International Conference on Multiple Objective Programming and Goal Programming (MOPGP); 30-31 October 2017; Metz, France. The Laboratory of Design, Optimization and Modelling of Systems, University of Lorraine. Abstract number is 14.

3 Slicing with Adaptive Steps Search method (SASS)

We present an algorithm to generate all ND solutions of TOMILPs. Each ND integer solution corresponds to a polyhedron that may contain 0, 1, or 2−dimensional ND facets for its sub-TOLP (the tri-objective LP after fixing integer decision variable). A facet can be completely/partially ND, and we aim to determine these completely/partially ND facets of TOMILPs.

3.1 The NDF of a TOMILP

Figure 1 shows the objective space of a maximization case of a TOMILP. Three polyhedra are specified by green, grey, and blue colors. Each feasible integer solution corresponds to a polyhedron that is a sub-TOLP in the objective space. In this example, we have three feasible integer solutions which we denote by y_{GR}, y_{GY}, and y_{BL}, and correspond to the green, grey, and blue polyhedra, respectively.

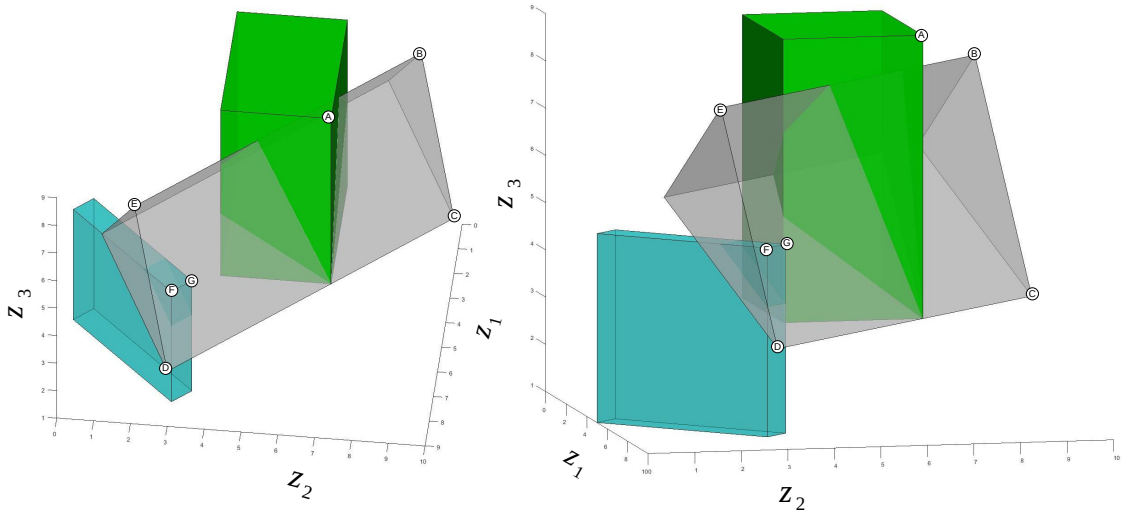

FIG. 1: An illustrative example for TOMILP

A ND solution is a solution that can not be improved in one objective value without worsening another objective value. Regarding Figure 1, an exact algorithm for TOMILP problems should generate ND point (A), and partially ND facet (B)–(C)–(D)–(E) as the NDF. In Figure 1, (A)$= [5, 7, 9]^T$ has the largest z_3-value. In addition, in the green sub-TOLP, (A) is the only ND solution. Facet (B)–(C)–(D)–(E)[1] is a ND facet for the grey sub-TOLP. However, it is a partially ND facet for the main TOMILP. This facet is not completely ND since a subset of its surface is dominated by (A).

In the blue polyhedron, there is no ND solution for the main TOMILP problem. Line segment (F)–(G) is ND for the blue sub-TOLP; however, it is dominated by some points in the facet (B)–(C)–(D)–(E).

3.2 General Design of SASS

SASS starts with a ND solution that is found by a lexicographic optimization. In the lexicographic optimization, we optimize the third objective function, then the second objective function while fixing the third to the optimal value found for z_3, then the first objective function

[1] (B)$= [1, 9, 8]^T$, (C)$= [2, 10, 3]^T$, (D)$= [9, 3, 3]^T$ (E)$= [8, 2, 8]^T$

while fixing the second and third objectives to the found values. In each iteration, the current ND solution is z^{cr}, Y_{exc} is the set of integer solutions that are excluded, and $ExConst$ is the set of constraints that should be added to exclude the known dominated regions. As shown in Figure 2, in each iteration, SASS aims to find the ND solutions in the slice $z_3(x, y) = z_3^{cr}$. If there is no ND solution in the current slice, we move to the next slice with smaller z_3 value.

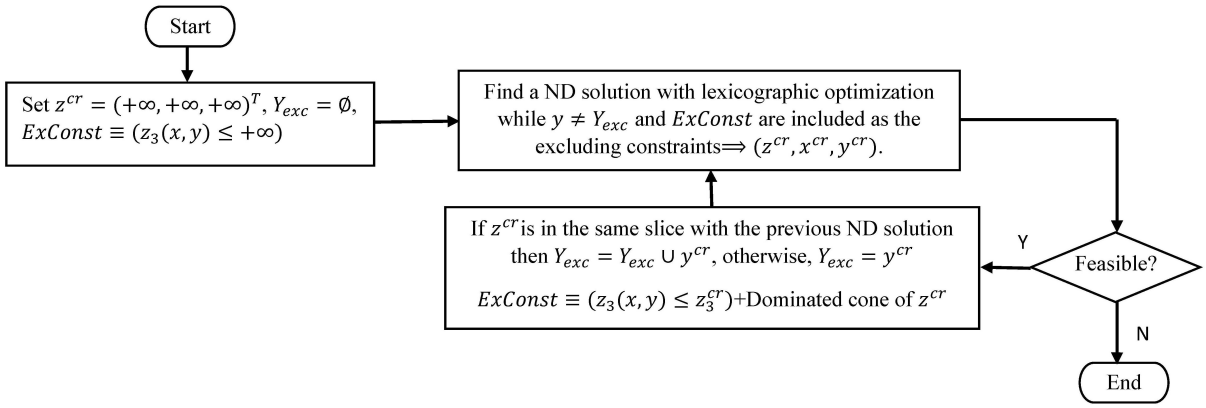

FIG. 2: The general design of SASS

Figure 2 shows the general steps of SASS. We explore the ND solution among the feasible solutions with z_3 value smaller than the z_3 value of the current iteration. In addition, the dominated cone of the found ND solutions should be excluded to avoid finding repeating solutions. SASS generates the ND facets associated with the found ND solutions after generating the ND points of each slice.

3.3 An illustrative example of SASS on a TOMILP

We provide an illustration using the example that we introduced in section 3.1 to explain the main steps of our algorithm. Figure 3 shows the process from different perspectives. In Figure 3(a), we apply a lexicographic optimization with no extra constraint ($Y_{exc} = \phi$ and $ExConst \equiv z_3(x, y) \leq +\infty$). Maximizing $z_3(x, y)$ results in all points in the top facet of the green polyhedron; then, we set $z_3(x, y) = 9$ and maximize $z_2(x, y)$, and so on. We find $z^{cr} = [5, 7, 9]^T$ that is ND in Figure 3(a).

In the next iteration, we exclude $Y_{exc} = y_{GR}$ and the cone generated by $[5, 7, 9]^T$, and add constraint $z_3(x, y) \leq 9$ (see Figure 3(b)). With these constraints, a lexicographic optimization results in $z^{cr} = [1, 9, 8]^T$ shown in Figure 3(c). Since we move to a slice with a z_3-value equals to eight (< 9), then $Y_{exc} = y_{GR}$ in the next iteration. In this iteration, SASS explores for a ND solution in the region $z_(x, y) \leq 8$ while excluding the cone that is dominated by $[1, 9, 8]^T$. In Figures 3(d) and 3(e), $z^A = [5, 7, 8]^T$ is the solution of the lexicographic optimization; however, it is dominated. Therefore, we use $z^{cr} = [1, 9, 8]^T$ as the current point and exclude y_{GR} and y_{GY}, and hence, we find $z^A = [7.5, 3.5, 5]^T$ (Figure 3(f)). This point is dominated.

In the last iteration, we exclude $Y_{exc} = y_{GR} \cup y_{GY} \cup y_{BL}$ and the dominated cone of z^{cr}. The lexicographic optimization is infeasible. Hence, there is no more ND solutions and the algorithm stops.

4 Conclusions

We present SASS method to generate the NDF of the TOMILP problems. SASS iteratively finds the ND points while decreasing the value of the third objective function with adaptive steps between the slices. At each iteration, our algorithm applies lexicographic optimization

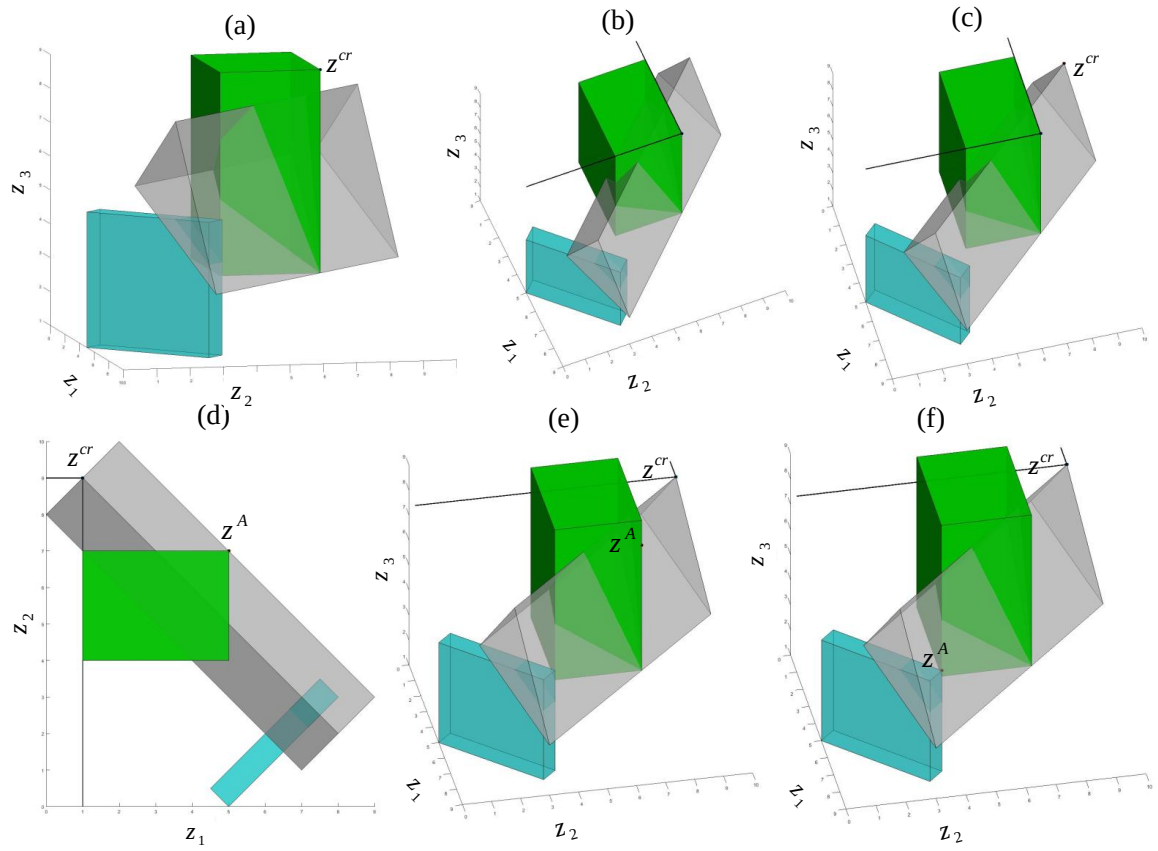

FIG. 3: An illustrative example for SASS

and excluding constraints to avoid finding repeated and dominated solutions. We illustrate our method on a small example.

References

[1] Raunak Bhinge, Raphael Moser, Emanuel Moser, Gisela Lanza, and David Dornfeld. Sustainability optimization for global supply chain decision-making. *Procedia CIRP*, 26:323–328, 2015.

[2] Natashia Boland, Hadi Charkhgard, and M Savelsbergh. A simple and efficient algorithm for solving three objective integer programs. *Optimization Online*, 2014.

[3] Natashia Boland, Hadi Charkhgard, and Martin Savelsbergh. The l-shape search method for triobjective integer programming. *Mathematical Programming Computation*, 8(2):217–251, 2016.

[4] Natashia Boland, Hadi Charkhgard, and Martin Savelsbergh. The quadrant shrinking method: A simple and efficient algorithm for solving tri-objective integer programs. *European Journal of Operational Research*, 260(3):873–885, 2017.

[5] Zhixiang Chen and Svenja Andresen. A multiobjective optimization model of production-sourcing for sustainable supply chain with consideration of social, environmental, and economic factors. *Mathematical Problems in Engineering*.

[6] H Hassine, M Barkallah, A Bellacicco, J Louati, A Riviere, and M Haddar. Multi objective optimization for sustainable manufacturing, application in turning. *International Journal of Simulation Modelling*, 14(1):98–109, 2015.