Scalarization Approaches for Four Objectives Optimization Problems: A Case Study

Jannatul Ferdous[1], M. M. Rizvi[2]

Abstract

We extend the algorithms introduced by Burachik et al. to approximate solutions to four objectives optimization problems. Efficient scalarization approaches are considered to design the algorithms and implement these algorithms for multiobjective mixed-integer programming problems. We show that algorithms with the weighted-constraint scalarization approach efficiently solve the problem even if the set of answers is discrete or disconnected. We test the efficiency of the algorithms for a four objective rocket injector design problem.

Keywords: Multiple objective programming, Mixed integer problems, Scalarization approaches, Pareto fronts, Numerical techniques.

1. Introduction

There are numerous conflicting objectives involved in multi-objective optimization problems whereby enhancing one criterion will reduce the value of others, leading to a exchange-off among answers. Therefore, any model that incorporates multiple objectives with continuous and discrete phenomena involves the consideration of *multi-objective mixed-integer programming* (MOMIP). MOMIP has many real-life applications, including problems in the fields of mining, engineering, and finance. However, as far as we know, very few algorithms (Antunes, Alves & Climaco , 2016; Belotti, Soylu & Wiecek, 2013; Przybylski, Gandibleux & Ehrgott, 2010; Pettersson & Ozlen, 2017) devised for MOMIP consider more than three objectives. Two such algorithms were presented in (Burachik, Kaya & Rizvi, 2014; Mueller-Gritschneder, Graeb & Schlichtmann, 2009) and tested for integer and mixed-integer programming problems. One of the reasons for not having enough literature for three and four objective cases is that the feasible set is not convex and might even be disconnected if the variables are integer or mixed-integer. This poses difficulties for the scalarization techniques to approximate the Pareto front.

[1]Jannatul Ferdous
Lecturer, Department of Mathematics, Premier University, Chittagong, Bangladesh
Email: jannatul.f25@gmail.com

[2]M. M. Rizvi
STEM, University of South Australia, Mawson Lakes, SA 5095, Australia, and
Department of Mathematics, University of Chittagong, Chittagong, Bangladesh
Email: rizmm001@mymail.unisa.edu.au.

In this paper, we analyse algorithms for four-objective mixed integer problems for approximating Pareto points. It is desirable that these algorithms should have the following practical attributes: (a) The method should approximate all discrete and continuous Pareto points when the problem has mixed-integer variables. This is an essential characteristic as many real-world applications require the consideration of both discrete and continuous variables (such as production levels and fixed charges).

(b) The method should generate the Pareto points in reasonable computational time. This is an important attribute as solving each scalarization problem can be very costly.

In this paper, we extend the Burachik et al. algorithm which is used to solve the rocket injector mixed-integer design problem. First, we make constraints more complicated by partitioning the hydrogen flow angle $x_1 = \{0,0.2,0.4,0.6,0.8,1\}$ into six equal sections, whereas in the Burachik et al., the author considered the variable into four segments ($x_1 = \{0,0.2,0.4,0.6\}$). Hence, the problem turns into a more challenging mixed-integer problem, which is tricky to solve. To solve this problem, we needed to modify the algorithms for the proposed new set of constraints and the associated coding in MATLAB. Moreover, the algorithm used to implement the Pascaloti approach to approximate the Pareto front is a novel work in this analysis.

2. Preliminaries

In this segment, we deal the key notions, terminologies, and ideas which can be used in our analysis. Those are popular notation and devices for multiobjective optimization, and we are using the classic notation here in the literature (Chankong & Haimes, 1983; Miettinen, 1999). Let $\mathbb{R}^n$ be the n-dimension Euclidean area. Let it be a set of nonnegative actual values. A set of numbers and strictly high quality numbers are denoted by $\mathbb{R}_+$ and $\mathbb{R}_{++}$ respectively. Define ℓ, m, n_1, n_2 belongs to $\mathbb{N}$, the multiobjective optimization Problem (P) is difined as follows.

$$\min \ f(x),$$

subject to

$$x \in Q := \{x \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} | h_j(x) \leq 0, \quad j = 1, \dots, \ell\},$$

where $f(x) = [\, f_1(x), \dots, f_m(x)\,]$ and $\mathbb{Z}$ is about of all integer numbers, and the functions f_i: $\mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$, $i = 1, \dots, m$, and h_j: $\mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$, $j = 1, \dots, \ell$

are defined in Q. ssume that the capabilities f_i are bounded under at the constraint set Q, with a recognized smaller bound. We require it to write the condition as follows.

$$\min_{i=1,\ldots,m} \{\min_{x\in Q} \quad f_i(x)\} > 0. \tag{1}$$

The solutions of Problem (P) are called *efficient points* (Burachik et al., 2014; Yu, 1985), or *Pareto points* (Miettinen, 1999). A more general concept of solution of (P) is the one of a *weak efficient* point. We use the following standard definitions to derive efficient points and weak efficient points.

Definition 2.1 (Chankong & Haimes, 1983 ; Miettinen, 1999)
(i) A point $\bar{x} \in Q$ is called an efficient for Problem (P) if and only if there exists no x in Q, and $x \neq \bar{x}$ such that $f(x) \leqq f(\bar{x})$.

(ii) A point $\bar{x} \in Q$ is called a weak efficient for Problem (P) if and only if there exists no x belongs to Q such that $f(x) < f(\bar{x})$ holds. Throughout the paper, we express weak efficient points of Problem (P) as WE(P).

Next, we introduce a reference vector, called an *emphutopia* vector. Utopia vectors are exactly superior to any efficiency point defined as.

Definition 2.2 Let, $u_i^* := f_i^* - \delta_i$, where $\delta_i > 0$ for all $i=1,\ldots,m$.
If f_i^* is the optimal value of the optimization problem

$$\min_{x\in Q} \quad f_i(x), \qquad\qquad (P_i)$$

for i=1,…,m, then $u^* \epsilon\ \mathbb{R}^l$ is called a utopia vector of problem (P).

3. Classical Scalarization Approaches

Two scalarization techniques are recalled in this section. These are known as the Pascoletti-Serafini approach and the Weighted-Constraint approach. The former approach is popular when $n_1 = 0$ (no integer variable). We also recall here the weighted-constraint approach this is green in approximating Pareto points while the front is disconnected. Even though these techniques are known, for the ease of the reader we supply right here a quick description. For extra information on these techniques, (Burachik, Kaya & Rizvi, 2017) and references therein.

Sketch a set of valid weights and

$$W := \left\{ w \quad is\ \ in \quad \mathbb{R}^m | w_i > 0\,, \qquad \sum_{i=1}^{\ell} w_i = 1 \right\}.$$

The Pascoletti-Serafini approach: The parameter-based scalarization approach, introduced by Pascoletti and Serafini (Pascoletti & Serafini, 1984) is widely used for approximating Pareto points in the front. This method is also referred to as goal-attainment method (Miettinen, 1999 ; Dutta & Kaya, 2011 ; Collette & Siarry, 2004). Let w be chosen from W, and $(\beta, x) \in \mathbb{R} \times \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}$, the scalarization is stated as follows.

$$\min_{(\beta,x)} \beta, \qquad s.t. \quad w_i\,(f_i(x) - u_i) \leq \beta\,, \quad \forall\ i = 1, \ldots, m\,, \tag{PS}$$

where $\beta \in \mathbb{R}$ and the variable and u is utopia vector in question (P). In (Eichfelder, 2008), it's miles proved that each answer of Problem (PS) were weak and efficient coefficient used in reverse to generate an approximation of. in use previous Pareto.

Weighted Constraint Approach: One of the most efficient methods for approximating weak efficient solutions for disconnected Pareto front is the Weighted-Constraint approach. In this method, one of the k targets is consider for minimization and uses the other $k-1$ weighted goals for constraints: The main advantage of this method, illustrated in (Burachik et al., 2014), is that it is able to generate Pareto points of nonconvex issues with separate Parate fronts and separate executable sets. The math version of the problem is,

$$\begin{cases} \min\limits_{x \in Q} & f_k(x) \\ subject\ \ to & w_i\, f_i(x) \leq w_k\, f_k(x)\,, \quad i = 1, \ldots, m, \quad i \neq k\,, \end{cases} \tag{P_w^k}$$

General results can be found in (Burachik et al., 2014, Theorem 3.1), based on the fact that for fixed w is in W the solution of P_w^k for all $k \in \{1, \ldots, m\}$ is a weakly efficient point. This means that if the solution of each k sub-problems is the same, then the solution is a weak green factor. On the other hand, if the answers are exceptional, a assessment is made among answers to differentiate the ruled factors from the true solutions. The proposition below plays a crucial position inside the weighted-constraint technique. We use it in the Algorithms given in (Burachik, Kaya & Rizvi, 2021) for removing dominated points. For the convenience of the reader, we recall it below.

Suppose the solution set of Problem (P^{k}_{w}) as

$$Q^k_w := \{x \in Q | x \text{ solves } (P^{k}_{w})\}, \quad k = 1,2,\dots,m.$$

The following proposition is recalled from (Burachik et al., 2014, Proposition 3.3).

Proposition 3.1 *Let* $\exists\, w \in W$ *such that* $Q^j_w \neq \emptyset,\ \forall\ j = 1,\dots,m.$ *Assume that, for a few* $k \in \{1,\dots,m\}$, $\exists\, \bar{x}_k \in Q^k_w$ *such that* $\forall r \in \{1,2,\dots,m\},\ r \neq k,\ \exists\, \bar{x}_r \in Q^r_w,$ *was*

$$f_r(\bar{x}_r) \geq f_r(\bar{x}_k)\,. \tag{2}$$

Then $\bar{x}_k \in WE(P)$.

4. Algorithm

Our analysis uses two grid technology strategies to design the algorithms for approximating Pareto factors. In this paper, we reported only the results obtained by SBG grids as CHIM failed to produce some parts of the Pareto front of rocket injector design problem. The reader can see the geometric interpretation of these grid generation techniques in (Burachik et al., 2017, Sections 1,2 ; Burachik et al., 2014).

Individual Minimum Convex Hull (IMCH): IMCH grid era technique become delivered with the aid of Das and Dennis (Das & Dennis, 1998).This technique was employed in the proposed Normal Boundary Intersection method (NBI) in (Das & Dennis, 1998), which is possibly the most popular approach for approximating the Pareto front. In this technique, the minimum values of every goal function are taken, and then the convex hull of these man or woman minima are produced.

The Sequential Boundary Generation (SBG): SBG became proposed by Mueller-Gritschneder et al (Mueller-Gritschneder et al., 2009). We check in our experiments that the SBG works while the IMCH mesh cannot generate some parts of the Pareto the front. The sequential SBG technique builds the Pareto before the trouble. Construction of Pareto front by SBG method requires solving additional linear programming issues. As a result, it needs greater computational time than the IMCH approach. Detailed descriptions of this grid era technique, along with their geometric depictions, can be found in (Burachik et al., 2017, Sections 1.2 ; Burachik et al., 2014). However, for the completeness of the paper, we recall these geometric interpretations below.

Figure 1 illustrates the construction of SBG grid for three objectives case. First the individual minima are approximated by minimizing f_1, f_2 and f_3, separately. Then the Pareto front of two-objective subproblems $[f_1, f_2]$, $[f_1, f_3]$, and $[f_2, f_3]$ are computed. The combination of arbitrary Pareto fronts of these subproblems provides a distinct boundary of the real Pareto front. Next, the grid of the weights of the interior part of the boundary is calculated by solving a linear programming problem. In the final step, Pareto front of the interior part is constructed by minimizing $[f_1, f_2, f_3]$.

In the algorithms we use the weighted-constraint scalarization (Problem (P (P $_w^k$)), and the Pascoletti-Serafini scalarization (Problem (PS)).

4.1 A four-objective set of rules: the use of the SBG grid

The description of the four-goal set of rules is given in (Burachik et al., 2021, Appendix A) This analysis is the extension of Algorithm given in (Burachik et al., 2021, Appendix A) (with the SBG grid) considering hydrogen flow angles $x_1 = \{0,0.2,0.4,0.6,0.8,1\}$ of the rocket injector design problem. Note that the IMCH grid generation technique does not work for the rocket injector layout issues (stated in Section 5.1) due to the complex pareto boundaries in front of the issue. To solve this problem, one needs to

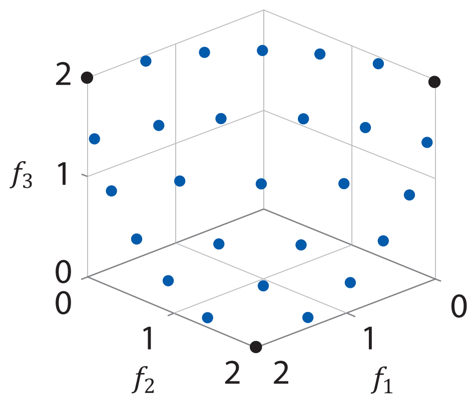


Figure 1: (Burachik et al., 2017, Figure~2) SBG communication network. The Corners of the small circles indicate the maximum value of the person, the circles indicate the Pareto boundary coefficients, and the interval coefficients of the grid are displayed.

use SBG grid generation. We use the scalarization approaches (P $_w^k$) and (PS) in the algorithm.

The Pascoletti-Serafini approach and the Weighted Constraint Approach are used in the algorithm to approximate Pareto fronts of the rocket injector design problem. In this section we presented algorithms designed to write code in MATLAB and solve the problem. Description of the algorithm-steps are as follows.

Step 2 of Algorithms: All four objectives are minimized individually subject to the constraints given in rocket injector design problem presented in Section 5.1.

Step 3 of Algorithms: These individual optimum solutions are used to form weighted grids which are used for both Pascoletti-Serafini and Weighted Constraint Approaches. Each grid point corresponds to a weight vector which is in $\mathbb{R}^4$.

Step 4 of Algorithms: Four sub-problems are solved at each grid point to generate Pareto points. In this step we calculate efficient and weak efficient points based on both scalarization properties.

The rocket injector problem typically encountered with the disconnected Pareto fronts. Therefore, effcient algorithms are required to program the problem in MATLAB, and appropriate solver are used to approximate the Pareto points.

Step 1 (Input parameters)
Set all parameters that introduced in the rocket injector design problem.

Step 2 (Calculate optimum point of individual objective function)
Find optimum solution of $\min f_i$, $i = 1, 2, 3, 4$, subject to the constraints of the rocket injection problem introduced in Section 5.1, that provide the solutions $(\bar{x}_1^i, \bar{x}_2^i, \bar{x}_3^i, \bar{x}_4^i)$, for $i = 1,2,3,4$, respectively.

Step 3 (Generate weighted parameters)
Weighted grids are created in this step. We follow the same ways to generate grids as introduced in (Burachik et al., 2017, Step 3 of Algorithm 3). Find x that solves auxiliary problems $\min_{x \in X} \; w_k f_k(x)$, s.t. $w_i f_i(x) = w_k f_k(x)$, for some k and find $\bar{x}$ that solves $\min_{(\beta,x)} \beta$, s.t. $w_i \, (f_i(x) - u_i) \leq \beta \;\; \forall \; i = 1, \dots, m$.

Step 4 Choose $w = (w_1, w_2, w_3, 1 - w_1 - w_2 - w_3)$, which generated from Step 3.

Case: I Implemented Weighted Constraint Approach (P_w^k)
(a) Find $\hat{f}_k = (\bar{f}_1, \bar{f}_2, \bar{f}_3, \bar{f}_4),$ that solves Problems $(P_w^k),\ k = 1,2,3,4.$

(b) Determine weak efficient points :
(i) If $\bar{x}_1 = \bar{x}_2 = \bar{x}_3 = \bar{x}_4,$ then set $\bar{x} = \bar{x}_i$ for some $i = 1,2,3,4.$
(an efficient point) and record the points.

(ii) If $\bar{x}_1 = \bar{x}_2 = \bar{x}_3 = \bar{x}_4$ does not hold, then, any dominated point is discarded by comparing the above four solutions.
Record non dominated points.

Case: II Implemented Pascoletti-Serafini (PS)

(a) Find $\hat{f}_k = (\bar{f}_1, \bar{f}_2, \bar{f}_3, \bar{f}_4),$ that solves Problem (PS), for all $i = 1,2,3,4.$

(b) Determine weak efficient points and then any dominated point is discarded by comparing the above solutions.
Record the non dominated points.

Step 5 (Output)
All recorded points are Pareto points of the rocket injector design problem.

Codes are written in MATLAB. The range of solvers are tested in Steps 3 and 4(a), these include smooth and non-smooth solvers such as fmincon with sequential quadratic programming algorithm, SCIP and SolvOpt.

5. Numerical Results

This section tests and compares four objective algorithms for the rocket injector layout issues. We aim to understand the algorithm's capacity to approximate the Pareto points when two well-known scalarization approaches are considered. The assignment of approximating the Pareto factors is especially difficult for four-goal instances.

We applied the set of rules and modelled the rocket injector problem using MATLAB. We have used BARON and SCIP in solving the scalarized integer issues with default alternatives. Within SBG Lattice technology , MATLAB's linear programming solver is used, with standard alternatives to address related linear programming (LP) issues. The calculation was performed on a Dell Core i7 laptop with 16 GB RAM.

5.1 Application to rocket injector design

Liquid Rocket Injector layout trouble turned into formerly analyzed with multiple criteria objectives in (Goel, Vaidyanathan, Haftka, Shyy, Queipo & Tucker, 2007; Vaidyanathan, Tucker, Papila & Shyy, 2004). Two primary objectives are related with the injector design, one is the performance improvement of the injector and another is its survivability. The injector performance is enhanced through the ejector chamber's shaft life while the injector's life is related to the thermal element within the ejector chamber. For version and the injector layout visual representation, invites the reader to (Goel et al., 2007). There are conflicting interaction: high temperatures enhance the overall performance of the injector and shorten the service life of the component. Four design variables have been introduced in (Goel et al., 2007).to build a mathematical version of the rocket injector layout hassle. The modified rocket injector layout hassle is taken into consideration as a combined integer multiobjective optimization hassle defined as follows.

$$\min \quad [f_1, f_2, f_3, f_4]$$

where

$$\begin{aligned} f_1 = {} & 0.692 + 0.477x_1 - 0.687x_4 - 0.08x_3 - 0.065x_2 - 0.167x_1^2 - 0.0129x_1x_4 \\ & + 0.0796x_4^2 - 0.0634x_1x_3 - 0.0257x_3x_4 + 0.0877x_3^2 - 0.0521x_1x_2 \\ & + 0.00156x_2x_4 + 0.00198x_2x_3 + 0.0184x_2^2 , \end{aligned}$$

$$\begin{aligned} f_2 = {} & 0.37 - 0.205x_1 + 0.0307x_4 + 0.108x_3 + 1.019x_2 - 0.135x_1^2 + 0.0141x_1x_4 \\ & + 0.0998x_4^2 + 0.208x_1x_3 - 0.0301x_3x_4 - 0.226x_3^2 + 0.353x_1x_2 - 0.0497x_2x_3 \\ & - 0.423x_2^2 + 0.202x_1^2x_4 - 0.281x_1^2x_3 - 0.342x_1x_4^2 - 0.245x_3x_4^2 + 0.281x_3^2x_4 \\ & - 0.184x_1x_2^2 + 0.281x_1x_3x_4, \end{aligned}$$

$$\begin{aligned} f_3 = {} & 0.153 - 0.322x_1 + 0.396x_4 + 0.424x_3 + 0.0226x_2 + 0.175x_1^2 + 0.0185x_1x_4 \\ & - 0.0701x_4^2 - 0.251x_1x_3 + 0.179x_3x_4 + 0.015x_3^2 + 0.0134x_1x_2 + 0.0296x_2x_4 \\ & + 0.0752x_2x_3 + 0.0192x_2^2, \end{aligned}$$

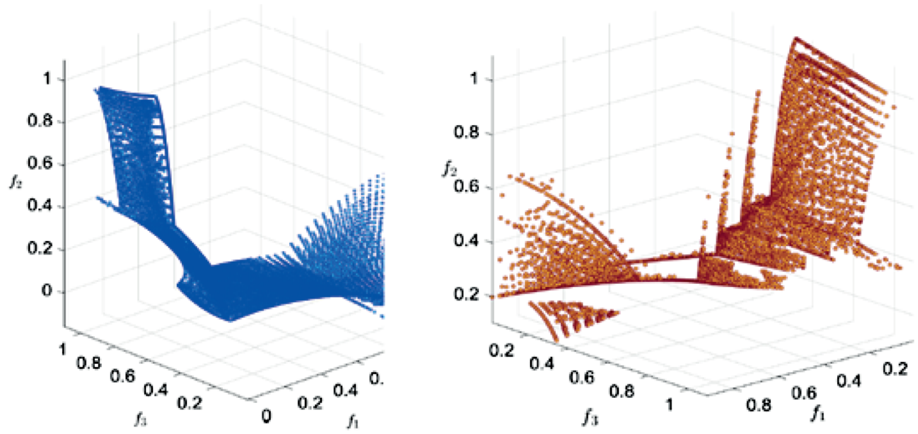
$$\begin{aligned} f_4 = {} & 0.758 + 0.358x_1 - 0.807x_4 + 0.0925x_3 - 0.0468x_2 - 0.172x_1^2 + 0.0106x_1x_4 \\ & + 0.0697x_4^2 - 0.146x_1x_3 - 0.0416x_3x_4 + 0.102x_3^2 - 0.0694x_1x_2 \\ & - 0.00503x_2x_4 + 0.0151x_2x_3 + 0.0173x_2^2 , \end{aligned}$$

subject to

$$x_1 = 0.2\tilde{x}_1, \quad 0 \le \tilde{x}_1 \le 5 , \quad \tilde{x}_1 \quad integer ,$$

$$0 \le x_1, x_2, x_3, x_4 \le 1 .$$

where x_1, x_2, x_3, x_4 are hydrogen go with the flow attitude, oxidizer submit tip thickness, decreases when admiring for the baseline cross-sectional region of the tube sporting oxygen, and increases with



(a) Projection of the Pareto Front for P_w^k scalarization approach .

(b) Projection of the Pareto Front (PS) scalarization approach .

Figure 2: Pareto Fronts for P_w^k and PS.

appreciate to the baseline go-segment region of the tube sporting hydrogen, respectively.

In our experiments, we impose a new equality constraint $x_1 = 0.2\,\tilde{x}_1$ and an integer variable $\tilde{x}_1$ which control the hydrogen flow angle (x_1) to fit in a set of angles. In our test, we optimize the rocket injector design problem for considering hydrogen flow angles $x_1 = \{0,0.2,0.4,0.6,0.8,1\}$.

Four goal capabilities are identified and indexed in (Goel et al., 2007) as: $f_1(x)$, $f_1(x)$, $f_1(x)$ and $f_1(x)$ represents face temperature, tip temperature, combustion length, and wall temperature, respectively.

We have used Algorithms given in (Burachik et al., 2021) ,which implements the weighted-constraint scalarization and Pascoletti-Serafini approach with the SBG grid in obtaining 4D approximation of the Pareto front. We compare the results obtained for both scalarization approaches when mixed-integer case is considered. Select prediction display $(f_1, f_2, f_3, f_4) \mapsto (f_1, f_2, f_3, 0)$ in the $f_1 f_3 f_2$ -spaces to compare the solutions found for mixed-integer variables that are shown in Figures 2(a) and (b), respectively.

Figure 2(b) shows the Pareto projection front into space $f_1 f_3 f_2$. (PS) approach approximated 10025 points including 85235 Pareto point. The elapsed CPU time was about 65 hours. On the other hand, the prediction approximation of Figure 2(a) obtained 90,000 Pareto points for continuous variables in 55 hours CPU time as reported in (Burachik et al., 2017) .Note that solving mixed integer problems required more computer memory and more CPU time. We have used BARON and SCIP in Algorithm 7 to solve subproblems as these solvers are quite efficient for dealing with non-linear mixed integer problems.

6 Conclusion

In this article, we implemented an algorithm that was added to Burachik et al. for approximating the Pareto points of multiobjective mixed-integer programming problems. Both CHIM and SBG grids were used in the algorithms. We compared both results using the same algorithms with the same number of grids. Our analysis shows that algorithms used the weighted-constraint scalarization approach, which is more efficient than the Pascoletti-Serafini approach in terms of computational time. It has also been shown that the Pareto fronts obtained by weighted-constraint scalarization approach approximated a reasonable Pareto front compared to the Pascoletti-Serafini approach.

References

Antunes, C. H., Alves, M. J., & Climaco, J. (2016). *Multiobjective Linear and Integer Programming*. Springer.

Belotti, P., Soylu, B., & Wiecek, M. M. (2013). A Branch-and-Bound Algorithm for Biobjective Mixed-Integer Programs. *http://www.optimization-online.org/DB HTM L/2013/01/3719.html.*

Burachik, R. S., Kaya, C. Y., & Rizvi, M. M. (2014). A New Scalarization Technique to Approximate Pareto Fronts of Problems with Disconnected Feasible Sets. *Journal of Optimization Theory and Applications*, *162*(2), 428–446.

Burachik, R. S., Kaya, C. Y., & Rizvi, M. M. (2017). A New Scalarization Technique and New Algorithms to Generate Pareto Fronts. *SIAM Journal on Optimization*, *27*(2), 1010–1034.

Burachik, R. S., Kaya, C. Y., & Rizvi, M. M. (2021). Algorithms for Generating Pareto Fronts of Multiobjective Integer and Mixed-Integer Programming Problems. *Engineering Optimization, https://doi.org/10.1080/0305215X.2021.1939695.*

Chankong, V., & Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology*. Amsterdam: North-Holland.

Collette, Y., & Siarry, P. (2004). *Multiobjective Optimization*. Berlin, Heidelberg: Springer.

Das, I., & Dennis, J. E. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, *8*, 631–657.

Dutta, J., & Kaya, C. Y., (2011). A New Scalarization and Numerical Method for Constructing Weak Pareto Front of Multi-Objective Optimization Problems. *Optimization*, *60*(8-9), 1091–1104.

Eichfelder, G. (2008). Adaptive Scalarization Methods in Multiobjective Optimization. Berlin, Heidelberg: Springer.

Goel, T., Vaidyanathan, R., Haftka, R. T., Shyy, W., Queipo, N. V., & Tucker, K. (2007). Response Surface Approximation of Pareto Optimal Front in Multi-Objective Optimization. *Computer Methods in Applied Mechanics and Engineering*, *196*(4-6), 879–893.

Miettinen, K. M. (1999). *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic.

Mueller-Gritschneder, D., Graeb, H., & Schlichtmann, U. (2009). A Successive Approach to Compute the Bounded Pareto Front of Practical Multiobjective Optimization Problems. *SIAM Journal on Optimization*, *20*(2), 915–934.

Pascoletti, A., & Serafini, P. (1984). Scalarizing Vector Optimization Problems. *Journal of Optimization Theory and Applications*, *42*(4), 499–524.

Pettersson, W., & Ozlen, M. (2017). A Parallel Approach to Bi-objective Integer Programming. *ANZIAM Journal (to appear)*.

Przybylski, A., Gandibleux, X., & Ehrgott, M. (2010). A Recursive Algorithm for Finding All Nondominated Extreme Points in the Outcome Set of a Multiobjective Integer Programme. *INFORMS Journal on Computing*, *22*(3), 371–386.

Vaidyanathan, R., Tucker, P. K., Papila, N., & Shyy, W. (2004). CFD-Based Design Optimization for a Single Element Rocket Injector. *Journal of Propulsion and Power*, *20*(4), 705–717.

Yu, P. L. (1985). *Multicriteria Decision Making: Concepts, Techniques, and Extensions*. New York: Plenum Press.