Communication in Physical Sciences, 2026, 13(8):1230-1252

Solution Approaches to Multiple Objective Linear Programming

Paschal Bisong Nyiam and Abdellah Salhi

Received: 18 April 2026/Accepted: 19 July 2026/Published: 28 July 2026

https://doi.org/10.4314/cps.v13i8.2

Abstract: *Multiple Objective Linear Programming (MOLP) has become an important optimization framework for solving decision-making problems involving multiple and often conflicting objectives. Over the years, several algorithms have been developed to generate efficient or nondominated solutions to MOLP problems, each exhibiting distinct computational characteristics and solution capabilities. This study reviews and evaluates four widely used MOLP solution approaches: the Evans–Steuer Multiobjective Simplex Algorithm (MSA), the Parametric Simplex Algorithm (PSA), the Affine Scaling Interior MOLP Algorithm (ASIMOLP), and Benson's Outer Approximation Algorithm (BOA). The algorithms are compared using ten benchmark and randomly generated MOLP test problems ranging from small to medium dimensions. Computational performance is assessed using CPU execution times, while solution quality is evaluated based on the nondominated solutions generated by each method. The results indicate that ASIMOLP consistently achieves the shortest computation times and is therefore the most computationally efficient among the algorithms considered. The PSA also demonstrates competitive computational performance, whereas the MSA becomes increasingly inefficient as problem size grows. In contrast, BOA provides the highest-quality solutions by generating a more complete representation of the nondominated frontier, although at a higher computational cost than ASIMOLP. The findings highlight the trade-off between computational efficiency and solution quality and provide guidance for selecting appropriate MOLP solution methods based on problem characteristics and decision-making requirements.*

Keywords: *Multiple Objective Linear Programming; Multi-objective Simplex Algorithm; Interior Point Methods; Parametric Simplex Algorithm; Objective Space Method.*

Paschal Bisong Nyiam
Department of Statistics, University of Calabar, Calabar, Nigeria
Email: nyiampaschal@unical.edu.ng
ORCID: 0000-0003-2535-3526

Abdellah Salhi
School of Mathematics & Actuarial Sciences, University of Essex, Colchester, United Kingdom
Email: as@essex.ac.uk

1.0 Introduction

Multiple Objective Linear Programming (MOLP), also known as Multi-Objective Linear Programming, is concerned with the simultaneous optimization of two or more conflicting linear objective functions subject to a common set of linear constraints. Unlike conventional linear programming, which seeks a single optimal solution, MOLP addresses situations where several objectives must be considered concurrently, often leading to trade-offs among competing goals (Ehrgott, 2006; Löhne, 2011). Such problems arise naturally in numerous fields, including engineering design, production planning, resource allocation, transportation, environmental management, finance, and

Also available @ https://doi.org/10.5281/zenodo.21738126

management science, (Alves *et al*., 2015; Yan *et al*., 2005). An MOLP problem can be formally express as:

$$\text{Min} \quad {c_1}^T x = f_1$$
$${c_2}^T x = f_2$$
$$\vdots$$
$${c_q}^T x = f_q$$

Subject to $x \in X = \{x \in \mathbb{R}^n : Ax \leq b, b \in \mathbb{R}^m, x \geq 0\}$. (1)

or alternatively as a linear vector optimization problem

$$\begin{array}{ll} \min & Cx \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \quad (2)$$

with $q \times n$ criterion matrix C consisting of the rows c_k, $k = 1, .., q$, $m \times n$ constraint matrix A and the right hand side vector $b \in \mathrm{R}^m$. The feasible set in the decision space is $X = \{x \in \mathrm{R}^n : Ax = b, x \geq 0\}$ and the feasible set in the objective space is $Y = \{Cx : x \in X\}$. The set Y is also referred to as the image of X. (Benson, 1998).

In MOLP, it is generally impossible to optimize all objective functions simultaneously because of their conflicting nature (Pandian & Jayalakshmi, 2013). Consequently, the classical concept of optimality is replaced by the notions of efficiency and Pareto optimality (Ehrgott, 2006; Löhne, 2011). The principal objective of MOLP is therefore to identify all efficient solutions, all nondominated points, a representative subset of these solutions, or a most preferred solution depending on the solution approach employed.

A feasible solution x^∈ X is said to be efficient (Pareto optimal) if there exists no other feasible solution x ∈ X such that $Cx \leq C\ddot{x}$ with at least one strict inequality. Similarly, x^\hat{x}x^ is weakly efficient if there exists no feasible solution x ∈ X satisfying $Cx < C\ddot{x}$

The image $y = C\ddot{x}$ of an efficient solution is called a nondominated point, while the image of a weakly efficient solution is referred to as a weakly nondominated point (Ehrgott *et al*., 2011). The sets of efficient and weakly efficient solutions are denoted by X_E and X_{WE}, respectively, whereas Y_N={Cx : x ∈ X_E} and Y_{WN}={Cx: x ∈ X_{WE}} represent the nondominated and weakly nondominated sets in the objective space. The collection of maximal efficient faces constitutes the efficient[1] frontier or Pareto frontier of the problem (Benson, 1998a; Ehrgott, 2006).

Research in MOLP dates back to the early 1960s and has evolved into one of the most active areas of operations research. During this period, numerous algorithms have been developed for generating efficient solutions, nondominated points, or preferred compromise solutions. These methods can broadly be classified into decision-space approaches and objective-space approaches. The major classes include the Multi-Objective Simplex Algorithm (MSA), Parametric Simplex Algorithm (PSA), Interior-Point Algorithms (IPA), and Objective-Space Algorithms (OSA). While the MSA, PSA, and IPA operate primarily in the decision space to generate efficient solutions, objective-space algorithms work directly in the outcome space to identify nondominated points and efficient frontiers (Benson, 1998b; Rudloff *et al*., 2015).

The present study focuses on four prominent solution approaches for MOLP: the Multi-Objective Simplex Algorithm (MSA) of Evans & Steuer (1973), the Parametric Simplex Algorithm (PSA) of Rudloff *et al.* (2015), the Affine Scaling Interior MOLP Algorithm (ASIMOLP) based on the weighted-gradient approach of Arbel (1993), and Benson's Outer Approximation (BOA) Algorithm (Benson, 1998b). These algorithms are among the most influential exact approaches for solving MOLP problems and represent different paradigms of multi-objective optimization. Their computational performance and solution quality are evaluated using randomly generated benchmark problems as well as instances obtained from the recently developed Multi-Objective Problem Library (MOPLIB) (Nyiam, 2019).

The remainder of this paper is organized as follows. Section 2 presents a comprehensive review of the literature on MOLP solution methods and comparative studies. Section 3 describes the Multi-Objective Simplex Algorithm, Parametric Simplex Algorithm, Affine Scaling Interior MOLP Algorithm, and Benson's Outer Approximation Algorithm. Section 4 presents computational experiments and comparative results. Finally, Section 5 summarizes the major findings and conclusions of the study.

2.0 Related Literature

Multiple Objective Linear Programming (MOLP) has remained an active area of research since the early 1960s. Over the years, numerous algorithms have been developed for generating efficient solutions, nondominated points, subsets of the Pareto frontier, or a most preferred solution based on the decision maker's preferences. The major classes of solution approaches include the Multi-Objective Simplex Algorithm (MSA), Parametric Simplex Algorithm (PSA), Interior-Point Algorithms (IPA), and Objective-Space Algorithms (OSA). While the MSA, PSA, and IPA operate primarily in the decision space to identify efficient solutions and their corresponding nondominated points, objective-space algorithms work directly in the outcome space to generate the nondominated set (Benson, 1998; Ehrgott, 2006; Löhne, 2011).

This study focuses on four prominent MOLP solution approaches: the Multi-Objective Simplex Algorithm (MSA) developed by Evans & Steuer (1973), the Parametric Simplex Algorithm (PSA) proposed by Rudloff *et al*. (2015), the Affine Scaling Interior MOLP Algorithm (ASIMOLP), which is based on the weighted-gradient interior-point approach of Arbel (1993), and Benson's Outer Approximation (BOA) algorithm (Benson, 1998). These algorithms represent four major paradigms in MOLP solution methodology and are compared in terms of computational efficiency and solution quality using randomly generated test problems as well as benchmark instances obtained from the Multi-Objective Problem Library (MOPLIB) developed by Nyiam (2019).

The remainder of this paper is organized as follows. Section 2 reviews the evolution of MOLP solution approaches and comparative studies reported in the literature. Section 3 presents the theoretical foundations and computational procedures of the MSA, PSA, ASIMOLP, and BOA algorithms. Section 4 reports the numerical experiments and comparative results, while Section 5 concludes the study and highlights future research directions.

2.1 Historical Development of MOLP

The earliest documented contribution to MOLP is generally attributed to Schönfeld (1964), who proposed a parametric programming approach for determining efficient solutions. In addition, Schönfeld extended the duality concepts introduced by Gale *et al*. (1951) and subsequently generalized these ideas to nonlinear vector optimization problems (Schönfeld, 1970). Around the same period, Geoffrion (1967) investigated bicriteria linear programming problems using parametric linear programming techniques and demonstrated that efficient solutions are not necessarily restricted to extreme points of the feasible region. This observation later influenced the development of interior-point and objective-space methods (Nyiam, 2019).

The development of interactive MOLP methods gained significant momentum during the 1970s. Benayoun *et al*. (1971) introduced the Step Method (STEM), one of the earliest interactive approaches for multi-objective optimization. Shortly thereafter, Geoffrion *et al*. (1972) proposed an interactive procedure that incorporated decision-maker preferences into the optimization process. Evans & Steuer (1973) developed the Multi-Objective Simplex Algorithm, which became one of the most influential decision-space approaches for generating efficient extreme points. Additional advances were made through the interactive methods of Zionts & Wallenius (1976, 1983) and Steuer (1977), which further enhanced decision-maker involvement in identifying preferred solutions.

2.2 Comparative Studies of Interactive MOLP Methods

Aksoy *et al*. (1996) noted that the selection of algorithms for comparative evaluation is often influenced by two primary factors: the familiarity of researchers with a particular method and the popularity of the method within the research community. Despite the large number of algorithms proposed for MOLP, relatively few studies have conducted comprehensive comparative analyses.

One of the earliest comparative investigations was undertaken by Wallenius (1975), who evaluated the STEM procedure of Benayoun *et al*. (1971), the interactive method of Geoffrion *et al*. (1972), and the trial-and-error approach proposed by Dyer (1972). The results indicated that Dyer's method outperformed the other procedures across most evaluation criteria. The study also observed that decision makers generally placed greater emphasis on solution quality than on computational time.

Brockhoff (1985) compared the STEM procedure with the interactive MOLP algorithm developed by Steuer (1977). The findings showed that STEM was effective for relatively small MOLP problems, whereas Steuer's algorithm demonstrated superior performance on larger problems involving up to six objective functions.

Malakooti & Ravindran (1986) proposed an Interactive Paired Comparison Simplex Method and compared it with the interactive procedure of Zionts & Wallenius (1976). Their computational experiments revealed that the proposed method outperformed the Zionts–Wallenius procedure across all selected performance criteria.

Buchanan & Daellenbach (1987) conducted a comparative evaluation of four interactive MOLP methods: the Surrogate Worth Trade-Off Method (Haimes & Hall, 1974), the interactive procedure of Zionts and Wallenius (1983), the Weighted Tchebycheff Procedure of Steuer & Choo (1983), and the Simplistic Naïve Method of Posner and Wu (1981). Their results showed that the Weighted Tchebycheff Procedure consistently produced superior solutions, although it required relatively longer computational times. The Zionts–Wallenius procedure produced acceptable CPU times but generally performed poorly with respect to solution quality.

Similarly, Gibson *et al*. (1987) compared the Weighted Tchebycheff Procedure of Steuer & Choo (1983) with the interactive goal programming approach developed by Franz (1980). Their study found that Steuer and Choo's procedure produced more accurate solutions, whereas Franz's method required fewer iterations to reach convergence.

Reeves & Gonzalez (1989) compared the Simplified Interactive Multiple Objective Linear Programming (SIMOLP) procedure developed by Reeves & Franz (1985) with the Weighted Tchebycheff Procedure of Steuer and Choo (1983). The results demonstrated that SIMOLP generated slightly higher-quality solutions while requiring less computational effort because only one linear program was solved per iteration. Subsequently, Haksever & Ringuest (1990) investigated four variants of the SIMOLP procedure and showed that using the optimal basis from the previous iteration significantly reduced both CPU time and the number of iterations required for convergence.

Dell & Karwan (1990) proposed the K–D Procedure, an interactive weight-space reduction method based on a Tchebycheff utility function. When compared with the method of Zionts & Wallenius (1976), the K–D Procedure generated slightly better-quality solutions, although the Zionts–Wallenius method remained computationally more efficient.

Michalowski & Szapiro (1992) developed a Bi-Reference Procedure and compared it with the methods of Benayoun *et al*. (1971), Geoffrion *et al*. (1972), and Zionts & Wallenius (1983). Their numerical experiments demonstrated that the Bi-Reference Procedure consistently reached preferred solutions in fewer iterations than the competing methods.

Lotfi *et al*. (1997) introduced the Aspiration-Based Search Algorithm (ABSALG), an interactive procedure that incorporates decision-maker aspiration levels through a Tchebycheff utility function. Comparative analyses involving ABSALG, the SIMOLP procedure of Reeves and Franz (1985), and the Weighted Tchebycheff Procedure of Steuer & Choo (1983) showed that ABSALG produced solutions of competitive quality. However, SIMOLP exhibited superior computational efficiency. The authors also observed that the relative ranking of solution quality differed from that reported by Reeves & Gonzalez (1989), possibly because their study considered linear, quadratic, and Tchebycheff utility functions rather than only linear and nonlinear utility functions.

Objective-Space and Interior-Point Approaches

The limitations associated with decision-space methods motivated the development of objective-space algorithms. A significant breakthrough was achieved by Benson (1998), who proposed an Outer Approximation Algorithm capable of generating all nondominated extreme points directly in the outcome space. Comparative experiments with the Evans–Steuer Multi-Objective Simplex Algorithm (Evans & Steuer, 1973) demonstrated that objective-space approaches are particularly effective for large-scale MOLP problems and often outperform decision-space methods in terms of computational performance.

Further developments occurred in the area of interior-point methods. Lin *et al*. (2006) proposed a modified Interior-Point Algorithm and compared it with the interior-point method of Wen & Weng (1998) and conventional simplex-based approaches. Their results demonstrated that the modified interior-point algorithm was superior to the earlier method of Wen and Weng and significantly more efficient than simplex-based methods for large-scale optimization problems.

More recently, Rudloff *et al*. (2015) developed a Parametric Simplex Algorithm (PSA) for linear vector optimization problems and compared its performance with Benson's Outer Approximation Algorithm and the Evans–Steuer Multi-Objective Simplex Algorithm. Their computational experiments revealed that PSA outperformed Benson's algorithm on non-degenerate problems. However, Benson's algorithm remained more effective for highly degenerate instances. Furthermore, PSA exhibited substantially better computational performance than the traditional multi-objective simplex algorithm.

Overall, the literature demonstrates that no single MOLP solution method consistently outperforms all others across different classes of problems. The relative performance of an algorithm depends on factors such as problem size, degeneracy, computational complexity, and the desired quality of the generated efficient solutions. Consequently, comparative evaluations remain essential for understanding the strengths and limitations of existing methods and for guiding the selection of appropriate algorithms for specific MOLP applications (Aksoy *et al*., 1996; Ehrgott, 2006; Nyiam, 2019). Table 1 summarizes the major comparative studies conducted on interactive and exact solution methods for Multi-Objective Linear Programming (MOLP), highlighting the methods compared, evaluation criteria, and principal findings reported in the literature.

3.0 Solution Algorithms for Multiple Objective Linear Programming

This section presents the major solution algorithms that are widely recognized and frequently applied in solving Multiple Objective Linear Programming (MOLP) problems. As discussed earlier, these algorithms represent the principal classes of exact solution methods available in the MOLP literature. They include the Multi-Objective Simplex Algorithm (MSA) and the Parametric Simplex Algorithm (PSA), which are simplex-based approaches operating in the decision space; the Affine Scaling Interior MOLP Algorithm (ASIMOLP), which is based on interior-point methodology; and Benson's Outer Approximation Algorithm (BOA), which is an objective-space approach.

Table 1: Comparative Studies of Interactive Multi-Objective Linear Programming (MOLP) Methods

S/N	Study	Methods Compared	Criteria for Comparison	Findings
1	Wallenius (1975)	Step Method of Benayoun *et al*. (1971), Geoffrion *et al*. (1972), and Trial-and-Error Method of Dyer (1973)	DM's confidence in the final solution, ease of use, CPU time, solution time, and distance of the final solution from the efficient frontier	Numerical results showed that the Trial-and-Error Method outperformed the other two methods in most evaluation criteria.
2	Brockhoff (1985)	Benayoun *et al*. (1971) (STEM) and Steuer (1977)	DM's confidence in the final solution, number of objective functions, and ease of use	Steuer's algorithm was found to be superior to STEM for larger and more complex problem instances.
3	Malakooti and Ravindran (1986)	Proposed Interactive Paired Comparison Simplex Method and Zionts and Wallenius (1976)	CPU time, number of iterations, number of LPs solved, and total number of questions asked of the DM	The Interactive Paired Comparison Simplex Method outperformed the Zionts and Wallenius Method.
4	Buchanan and Daellenbach (1987)	Surrogate Worth Trade-Off Method (Haimes and Hall, 1974), Zionts and Wallenius (1976, 1983), Simplistic Naïve Method (Posner and Wu, 1981), and Steuer and Choo (1983)	DM's confidence in the most preferred solution, ease of use, CPU time, and time taken to identify the most preferred solution	Experimental results indicated that the method of Steuer and Choo converged more effectively to the most preferred solution than the other methods.
5	Gibson *et al*. (1987)	Steuer and Choo (1983) and Franz (1980)	Number of iterations required for convergence and accuracy of optimal solutions	Steuer and Choo's method was preferred for solution accuracy, whereas Franz's method required fewer iterations.
6	Reeves and Gonzalez (1989)	SIMOLP Procedure (Reeves and Franz, 1985)	Number of iterations and quality of solutions	The SIMOLP procedure generated slightly higher-quality solutions

		and Steuer and Choo (1983)	obtained	and was computationally more efficient than the method of Steuer and Choo.
7	Dell and Karwan (1990)	Proposed K-D Procedure and Zionts and Wallenius (1976)	Quality of solutions obtained and average CPU time	The K-D procedure produced slightly better-quality solutions, while the Zionts and Wallenius method was computationally more efficient.
8	Haksever and Ringuest (1992)	Four variants of the SIMOLP Procedure (Reeves and Franz, 1985)	Number of iterations and CPU time	Results showed that solving each successive LP using the optimal basis of the previous problem reduced both CPU time and the number of iterations.
9	Michalowski and Szapiro (1992)	Proposed Bi-Reference Procedure, Zionts and Wallenius (1983), Geoffrion *et al*. (1972), and Benayoun *et al*. (1971)	Number of iterations and optimal utility value	The proposed Bi-Reference Procedure outperformed the established methods considered in the study.
10	Lotfi *et al*. (1997)	Proposed Aspiration-Based Search Algorithm (ABSALG), SIMOLP (Reeves and Franz, 1985), and Steuer and Choo (1983)	Quality of solutions, number of questions asked of the DM, and average CPU time	SIMOLP was computationally more efficient than both ABSALG and Steuer and Choo's method. However, ABSALG compared favorably in terms of solution quality.
11	Benson (1998)	Proposed Benson's Outer Approximation (BOA) and Evans-Steuer (1973)	Average number of efficient extreme points	Results suggested that outcome-set-based methods are superior to decision-set-based approaches and show promise for solving large MOLP instances.
12	Lin *et al*.	Proposed	CPU time and	The proposed IPA

	(2006)	Interactive Pareto Algorithm (IPA), Wen and Weng (1998), and a Simplex-Based Method	accuracy of optimal objective values	outperformed the algorithm of Wen and Weng and was superior to the Simplex-Based Method for large problem instances.
13	Rudloff *et al.* (2015)	Proposed Parametric Simplex Algorithm (PSA), Benson (1998), and Evans-Steuer (1973)	CPU time	

Each of these algorithms employs a different strategy for generating efficient solutions or nondominated points. While simplex-based and interior-point methods primarily operate in the decision space to identify efficient solutions, objective-space algorithms work directly in the outcome space to construct the nondominated frontier. The theoretical foundations and computational procedures of these algorithms are presented in the following subsections.

3.1 Multi-Objective Simplex Algorithm (MSA)

The Multi-Objective Simplex Algorithm (MSA) considered in this study is the algorithm originally proposed by Evans and Steuer (1973) and subsequently presented in detail by Ehrgott (2006). The algorithm is one of the most widely used decision-space approaches for solving MOLP problems and is designed to generate all efficient extreme-point solutions of a linear multi-objective optimization problem.

The MSA operates in the decision space by systematically exploring adjacent efficient bases through simplex pivot operations. The algorithm maintains three principal data structures throughout its execution:

- **L_1** : a list of efficient bases awaiting processing;
- **L_2** : a list of efficient bases that have already been processed and accepted for output; and
- **N_E**: a list of efficient nonbasic variables identified during the search process.

The algorithm begins with an initialization phase in which two auxiliary linear programming problems are solved. These preliminary computations are used to determine whether the MOLP problem is feasible and whether efficient solutions exist. If the feasible region, $X= \{x \in R^n : Ax=b, x \geq 0\}b,x\geq 0\}$, is non-empty and the efficient solution set X_E exists, the algorithm proceeds to enumerate all efficient extreme points.

The enumeration process is accomplished through a sequence of simplex pivot operations between adjacent efficient bases. At each iteration, the algorithm identifies efficient nonbasic variables capable of generating new efficient bases and subsequently explores these bases. Newly discovered efficient bases are added to the processing list L_1 while previously examined efficient bases are transferred to the output list L_2. This procedure continues until all efficient bases have been examined and no additional efficient extreme points remain to be generated.

Because the MSA systematically traverses the set of efficient bases, it guarantees the identification of all efficient extreme-point solutions of the MOLP problem. However, its computational performance may deteriorate for large-scale problems due to the potentially large number of efficient extreme points that must be enumerated (Evans & Steuer, 1973; Ehrgott, 2006).

The detailed computational steps of the Multi-Objective Simplex Algorithm are presented below.

Algorithm 1 Multiobjective Simplex Algorithm (*Evans & Steuer, 1973*)

Input: A, b, C: *Problem data*
Initialize: *Set* $L_1 \leftarrow \varnothing,\ L_2 \leftarrow \varnothing$;
    Phase I: *Solve the LP* $\min\{e^T z : Ax + Iz = b, x, z \geq 0\}$. *If the optimal value of this*
        LP is nonzero, STOP, $X = \varnothing$;
        Otherwise let x^o *be a basic feasible solution of MOLP*
    Phase II: *Solve the LP* $\min\{u^T b + w^T C x^o : u^T A + w^T C \geq 0, w \geq e\}$. *If this LP is*
        infeasible, STOP, $X_E = \varnothing$;
        Otherwise let $(\hat{u}, \hat{w})$ *be an optimal solution*;
        Find an optimal basis B of the LP $\min\{\hat{w}^T Cx : Ax = b, x \geq 0\}$;
        Set $L_1 \leftarrow \{B\},\ L_2 \leftarrow \varnothing$.;
while $L_1 \neq \varnothing$ **do**: *The list of efficient bases to be processed*
    Choose $B \in L_1$, $L_1 \leftarrow L_1 \setminus \{B\}$, $L_2 \leftarrow L_2 \cup \{B\}$;
    Compute $\tilde{A}$, $\tilde{b}$, *and R according to B*;
    $N_E \leftarrow N$;
    for *all* $j \in N$
        Solve the LP $\max\{e^T v : Ry - r^j \sigma + Iv = 0; y, \sigma, v \geq 0\}$.
        If this LP is unbounded $N_E \leftarrow N_E \setminus \{j\}$;
        for *all* $j \in N_E$
            for *all* $i \in B$
                if $B' \leftarrow (B \setminus \{i\}) \cup \{j\}$ *is feasible*, $B' \notin L_1 \cup L_2$ **then**;
                $L_1 \leftarrow L_1 \cup B'$;
                endif;
            endfor;
        endfor;
    endfor;
endwhile.
Output: L_2: *The list of efficient bases.*

3.2 Parametric Simplex Algorithm (PSA)

The Parametric Simplex Algorithm (PSA) considered in this study is the algorithm developed by Rudloff, Ulus, and Vanderbei (2015) for solving linear vector optimization problems and multiple objective linear programming (MOLP) problems. Similar to the Multi-Objective Simplex Algorithm (MSA), the PSA operates in the decision space. However, unlike the MSA, which seeks to generate all efficient extreme-point solutions, the PSA aims to construct a solution concept in the sense of Benson (1998), namely, a subset of efficient extreme points that is sufficient to generate the entire efficient frontier.

The PSA was developed to address some of the computational limitations associated with the exhaustive enumeration strategy employed by traditional simplex-based MOLP algorithms. Rather than identifying every efficient extreme point, the algorithm focuses on generating only those efficient solutions that are necessary for describing the complete nondominated frontier. Consequently, the number of pivots and basis transitions required during the solution process is significantly reduced, resulting in improved computational efficiency (Rudloff *et al*., 2015).

The implementation of the PSA maintains several important data structures throughout the optimization process. These include:

- **Boundary Dictionaries (BD):** a collection of current simplex dictionaries representing candidate efficient regions in the parameter space;
- **Explored Directions (ED):** a set containing all pivot directions that have already been investigated for a given dictionary, typically represented by index pairs (i, j), where (i) denotes a leaving variable and (j) denotes an entering variable; and
- **Visited Dictionaries (VD):** a collection of dictionaries that have already been explored during the search process.

The algorithm begins with an initial optimal dictionary corresponding to a weighted-sum scalarization of the MOLP problem. At each iteration, the current dictionary is examined to identify candidate pivot directions capable of generating adjacent efficient dictionaries. Unlike the MSA, which may explore multiple leaving variables associated with a given entering variable, the PSA performs pivoting using only a single leaving variable selected from the set of admissible candidates. This strategy substantially reduces the number of simplex pivots required and contributes to the algorithm's superior computational performance (Rudloff *et al*., 2015).

As the algorithm progresses, it systematically moves from one dictionary to another while recording the corresponding basic feasible solutions. The efficient basic solutions obtained during this process are stored in a set of efficient extreme points, denoted by $\dot{X}$, which constitutes the algorithm's output. The search continues until all relevant boundary dictionaries have been explored and no unexplored efficient pivot directions remain.

An important characteristic of the PSA is that it does not attempt to enumerate all efficient extreme points of the MOLP problem. Instead, it identifies a minimal representative subset of efficient solutions that can be used to reconstruct the entire efficient frontier in the objective space. This feature makes the algorithm particularly attractive for solving large-scale MOLP and linear vector optimization problems where the total number of efficient extreme points may be prohibitively large (Benson, 1998; Löhne, 2011; Rudloff *et al*., 2015).

Computational studies conducted by Rudloff *et al*. (2015) demonstrated that the PSA is generally more efficient than the Multi-Objective Simplex Algorithm of Evans and Steuer (1973), especially for non-degenerate problems. The authors also reported that the PSA often requires fewer pivot operations and less computational time while still generating a complete representation of the efficient frontier. However, for highly degenerate problems, objective-space methods such as Benson's Outer Approximation Algorithm may exhibit superior performance (Benson, 1998; Rudloff *et al*., 2015).

Overall, the Parametric Simplex Algorithm represents a significant advancement in simplex-based MOLP solution methodologies by combining the strengths of traditional simplex procedures with a more efficient mechanism for generating the efficient frontier. As a result, it has become one of the most important decision-space approaches for solving contemporary multi-objective linear programming problems (Löhne, 2011; Rudloff *et al*., 2015).

3.2.1 Computational Procedure of the Parametric Simplex Algorithm (PSA)

The Parametric Simplex Algorithm (PSA) proposed by Rudloff *et al*. (2015) is a simplex-based method designed for solving linear vector optimization and multiple objective linear programming problems. Unlike the Multi-Objective Simplex Algorithm (MSA), which attempts to generate all efficient extreme-point solutions, the PSA seeks to construct a finite solution consisting of a representative subset of efficient extreme points and efficient directions that fully describe the efficient frontier. This approach significantly reduces the computational burden associated with exhaustive enumeration (Benson, 1998; Rudloff *et al*., 2015).

The algorithm operates by traversing adjacent dictionaries in the parameter space through simplex pivot operations. Throughout the solution process, it maintains a set of boundary dictionaries awaiting exploration, a collection of visited dictionaries, and a record of explored pivot directions. Efficient basic solutions and efficient directions generated during the search are stored and subsequently used to construct a complete representation of the efficient frontier (Löhne, 2011; Rudloff *et al*., 2015).

Algorithm 2: Parametric Simplex Algorithm (PSA)
Source: Rudloff *et al*. (2015)

Input: A, b, P
Initialize: Find D^0 and the index set of entering variables J^{D0}; BD$\leftarrow\{D^0\}$, $\bar{X}\leftarrow\{X^0\}$, R$\leftarrow\emptyset$ VS $\leftarrow\emptyset$, $\bar{X}^h\leftarrow\emptyset$, $E^{D0}\leftarrow\emptyset$
while BD $\neq\emptyset$ do
  Let D$\in$BD with nonbasic variables N and index set of entering variables J^D;
  for $j\in J^D$ do
    Let x_j be the entering variable;
    if $B^{-1}Ne^j\leq 0$ then
      Let x^h be such that $x^h{}_B=-B^{-1}Ne^j$ and $x^h{}_N=e^j$;
      $\bar{X}^h\leftarrow\bar{X}^h\cup\{x^h\}$
      $P^T[\bar{X}^h]\leftarrow P^T[\bar{X}^h]\cup\{-Z^T{}_N e^j\}$
    else
      Pick$i\in\text{argmin}_{\{i\in B,\,(B^{-1}N)ij>0\}}$ $[(B^{-1}b)_i/(B^{-1}N)_{ij}]$
      if $(j,i)\notin E^D$ then
        Perform the pivot with entering variable x_j and leaving variable x_i;
        Call the new dictionary $\bar{D}$ with nonbasic variables $\bar{N}=(N\cup\{i\})\backslash\{j\}$;
        if $\bar{D}\notin$ VS then
          if $\bar{D}\in$ BD then
            $E^{\bar{D}}\leftarrow E^{\bar{D}}\cup\{(i,j)\}$
          else
            Let $\bar{x}$ be the basic solution for $\bar{D}$;
            $\bar{X}\leftarrow\bar{X}\cup\{\bar{x}\}$
            $P^T[\bar{X}]\leftarrow P^T[\bar{X}]\cup\{P^T\bar{x}\}$
            Compute the index set of entering variables $J^{\bar{D}}$ of $\bar{D}$;
            Let $E^{\bar{D}}=\{(i,j)\}$;
            BD$\leftarrow$ BD$\cup\{\bar{D}\}$
          end if
        end if
      end if
    end if
  end for
  VS $\leftarrow$ VS$\cup\{D\}$, BD $\leftarrow$ BD$\backslash\{D\}$
end while

32: Output:

- $(\bar{X}, \bar{X}^h)$: A finite solution of P;
- $(P^T[\bar{X}], P^T[\bar{X}^h] \cup \{y_1, ..., y_t\})$: V**-representation of P**

3.3 Affine Scaling Interior MOLP Algorithm (ASIMOLP)

The Affine Scaling Interior MOLP Algorithm (ASIMOLP) considered in this study is the interior-point approach developed by Arbel (1993) for solving Multiple Objective Linear Programming (MOLP) problems. Unlike simplex-based methods such as the Multi-Objective Simplex Algorithm (MSA) and the Parametric Simplex Algorithm (PSA), which generate either all efficient extreme points or a representative subset thereof, the ASIMOLP algorithm is designed to identify a single efficient solution that reflects the decision maker's preferences (Arbel, 1993). Consequently, the algorithm converges to an efficient face of the feasible region rather than attempting to enumerate the entire efficient frontier.

Interior-point methods have attracted considerable attention in multi-objective optimization because of their computational efficiency, particularly when dealing with large-scale optimization problems (Lin *et al.*, 2006). Among the various interior-point approaches proposed in the literature, the affine scaling algorithm developed by Arbel (1993) remains one of the most widely cited and applied methods. Its popularity stems from its ability to combine the computational advantages of interior-point methods with a structured mechanism for incorporating decision-maker preferences into the optimization process (Arbel, 1993; Junior & Lins, 2009). The algorithm begins with the selection of a strictly feasible interior solution, $x^0 \in X$, which serves as the initial iterate. At each iteration, the algorithm generates q feasible search-direction vectors, $dx_{i,I} = 1,2,\ldots,q$, corresponding to the q objective functions of the MOLP problem. These direction vectors represent potential improvement paths within the interior of the feasible region and are computed using the affine scaling framework proposed by Arbel (1993).

To determine the relative importance of the competing objective functions, the algorithm incorporates the Analytic Hierarchy Process (AHP) developed by Saaty (1980). The decision maker is required to perform pairwise comparisons of the objective functions, resulting in a comparison matrix from which a priority vector, $p = \left(p_1, p_2, p_3. \dots \dots . p_g\right)^T$ is derived. The components of this vector represent the relative preferences assigned to the objectives and satisfy the condition

$$\sum_{i=1}^{q} p_i = 1$$

The preference vector is subsequently used to form a convex combination of the individual search directions, yielding a composite search direction

$$dx = \sum_{i=1}^{q} p_i dx = 1$$

This combined direction reflects both the geometric structure of the feasible region and the decision maker's preferences among the competing objectives (Arbel, 1993; Saaty, 1980).

The algorithm then moves from the current interior point to a new feasible point along the direction dx while maintaining feasibility with respect to the problem constraints. The process is repeated iteratively, with new search directions and preference-weighted movements being generated at each step. Convergence is achieved when a predefined stopping criterion is satisfied, such as negligible improvement in objective values or attainment of a stable efficient solution.

A notable advantage of the ASIMOLP algorithm is its ability to reach a preferred efficient solution without requiring the exhaustive enumeration of efficient extreme points. This characteristic significantly reduces computational effort and makes the algorithm particularly attractive for large-scale MOLP problems (Arbel, 1993; Lin *et al*., 2006). Furthermore, by explicitly incorporating decision-maker preferences through the Analytic Hierarchy Process, the method facilitates the identification of solutions that are not only efficient but also aligned with the decision maker's priorities (Saaty, 1980; Arbel, 1993).

Overall, the Affine Scaling Interior MOLP Algorithm provides an effective integration of interior-point optimization techniques and multi-criteria decision analysis, making it a valuable alternative to simplex-based and objective-space approaches for solving multiple objective linear programming problems (Arbel, 1993; Junior & Lins, 2009).

Algorithm 3: Affine Scaling Interior MOLP Algorithm (ASIMOLP)
Source: Nyiam and Salhi (2021)

Input: *Constraint matrix A, right-hand-side vector b, criterion matrix C*
Initialize:
  Choose $x_0 > 0$*, stopping tolerance* σ $(0 < \sigma < 1)$*, step size factor* ρ $(0 < \rho < 1)$*;*
  Converged $\leftarrow 0$*,* $k \leftarrow 0$
while Converged $\neq 1$ *do*
  $k \leftarrow k + 1$
  $D \leftarrow diag(x^0)$
  $y_i(k) \leftarrow (A D^2 A^T)^{-1} A D c_i$
  $dx_i(k) \leftarrow D^2(c_i - A^T y_i(k))$
  if $dx_i(k) \geq 0$*,* $1 \leq i \leq q$*, stop,*
  else
    $\alpha_i = Min[-x_i/dxi(k), \forall dxi(k) < 0]$
    $x_i(k) \leftarrow x(k+1) + \rho\, \alpha_i\, dx\_i(k)$
  $dx = \Sigma\, p_i dx_i,\ 1 \leq i \leq q$
  $x_{new} \leftarrow x(k) + \rho\, dx(k)$
    $dx(k) \leftarrow \sum p_i \alpha_i dx_i(k),\ 1 \leq i \leq q$
    $x_{new} \leftarrow x(k+1) + \rho\, dx(k)$
    $v_{new} \leftarrow C^T x_{new}$
    $x_{end} \leftarrow x(k+1) + dx(k)$
    $dx_{end} \leftarrow x_{end} - x(k+1)$

else
  $dx(k) \leftarrow \sum p_i\alpha_i dx_i(k) + p_{end}\, dx_{end}$
  $x_{new} \leftarrow x^{(k-1)} + \rho\, dx(k)$
  $v_{new} \leftarrow C^T x\_new$
  $x_{end} \leftarrow x^{(k-1)} + dx(k)$
  $v_{end} \leftarrow C^T xend$
  $dx_{end} \leftarrow x_{end} - x_{new}$
end if
if $\|dx\_{end}\| \leq \sigma$ *then*
  Converged $\leftarrow 1$
else
  $x_o \leftarrow x_{new}$
end if
end while

Output:
- *xend : Most preferred efficient solution*
- *vend : Values of the objective functions*

3.4 Benson's Outer Approximation Algorithm (BOA)

In this study, the modified version of Benson's Outer Approximation Algorithm (BOA) presented by Löhne (2011) is considered. Unlike the Multi-Objective Simplex Algorithm (MSA), the Parametric Simplex Algorithm (PSA), and the Affine Scaling Interior MOLP Algorithm (ASIMOLP), which operate primarily in the decision space, the BOA is an **objective-space algorithm**. The method is designed to generate the complete set of nondominated extreme points together with the extreme

3.4.1 Computational Procedure of Benson's Outer Approximation Algorithm (BOA)

The Benson Outer Approximation Algorithm (BOA) is one of the most influential objective-space methods for solving Multiple Objective Linear Programming (MOLP) and Linear Vector Optimization Problems (LVOPs). Originally proposed by Benson (1998) and subsequently refined and presented in a computationally efficient form by Löhne (2011), the algorithm generates the complete set of nondominated extreme points and extreme directions by constructing successive outer approximations of the upper image in the objective space. Unlike decision-space methods, BOA operates directly on the geometric structure of the objective space and therefore avoids the exhaustive enumeration of efficient bases (Benson, 1998; Löhne, 2011).

The algorithm may be viewed as a primal–dual procedure because it simultaneously constructs solutions to both the primal and dual vector optimization problems. However, in this study, emphasis is placed on the primal solution, namely the generation of all nondominated extreme points and extreme directions that define the efficient frontier (Löhne, 2011).

Algorithm 4: Benson's Outer Approximation Algorithm (BOA)
Source: Benson (1998); Löhne (2011)

1. **Input:** A, b, P.
   a finitely generated solution $(\{0\}, \bar{X}_1)$ to P^h;

a finitely generated solution $\bar{\mathcal{T}}^{h}$ to D^{*h}

2. Initialize:

$\hat{p} \leftarrow P(\text{solve}(P_1(0))) + \varepsilon$

$\bar{T} \leftarrow \{(\text{solve}(D^{h}(w)), w) \mid (u, w) \in \bar{T}^{h}\}$

3. **while** $z \neq 0$ **do**

4. $\mathcal{X}^{d} \leftarrow \{D^{*}(u, w) \mid (u, w) \in \bar{T}\}$;

5. $\mathcal{X}^{p} \leftarrow vert(\mathcal{X}_1)$;

6. $\bar{X} \leftarrow \emptyset$

7. **for** $i = 1$ **to** $/\mathcal{X}_{o}|$ **do**

8. $v \leftarrow \mathcal{X}^{p}[i]$

9. $(x, z) \leftarrow \text{solve}(P2(v))$;

10. $\bar{X} \leftarrow \bar{X} \cup \{x\}$;

11. **if** $z > 0$ **then**

12. $(x, \delta) \leftarrow \text{solve}(R(v))$;

13. $y \leftarrow \delta v + (1 - \delta)\hat{p}$;

14. $(u, w) \leftarrow \text{solve}(D2(y))$;

15. $\bar{T} \leftarrow \bar{T} \cup \{(u, w)\}$;

16. Break

17. **end if**

18. **end for**

19. **end while**

20. **Output:** $(\bar{\mathcal{X}}, \bar{\mathcal{X}}^{h})$ a finitely generated solution to P;

$(\bar{\mathcal{T}})$ a finitely generated solution to D^{*};

4.0 Results and Discussion

This section presents the computational results obtained from the numerical experiments conducted to evaluate and compare the performance of the four MOLP solution approaches considered in this study, namely the Multi-Objective Simplex Algorithm (MSA) of Evans and Steuer (1973), the Parametric Simplex Algorithm (PSA) of Rudloff *et al*. (2015), the Affine Scaling Interior MOLP Algorithm (ASIMOLP) of Arbel (1993), and Benson's Outer Approximation Algorithm (BOA) (Benson, 1998; Löhne, 2011).

The comparison focuses primarily on computational efficiency, measured in terms of CPU execution time, and the quality of the solutions generated by each algorithm. The numerical results for ten test problems are summarized in Tables 2a and 2b.

Problems 1 and 2 were obtained from the literature, while Problems 3 and 4 were selected from the Interactive MOLP Explorer problem set developed by Alves *et al*. (2015). Problems 5–7 were randomly generated with fully dense objective and constraint matrices. For Problems 8–10, only the constraint matrices were generated as fully dense while the objective matrices retained a less dense structure.

The implementation of ASIMOLP was carried out in MATLAB based on the algorithm proposed by Arbel (1993). For BOA, the MATLAB implementation provided by **Bensolve 1.2** (Löhne, 2012) was employed. The results for MSA and PSA were obtained using the MATLAB implementations reported by Rudloff *et al*. (2015), thereby ensuring that all algorithms were evaluated under a common computational environment.

In Table 2, mmm denotes the number of constraints, nnn the number of decision variables, and qqq the number of objective functions. For each problem, the most preferred solution returned by each algorithm together with the corresponding CPU time (in seconds) is reported.

4.1 Computational Performance Analysis

The results presented in Table 2 reveal substantial differences in computational performance among the four algorithms. The MSA exhibited the poorest scalability as problem size increased. Although the algorithm performed satisfactorily on small problems, its CPU time increased dramatically with increasing problem dimensions. For example, the execution time increased from 0.146 seconds for Problem 1 to 125,416.24 seconds for Problem 5. Furthermore, for Problems 6–10, involving at least 35 variables and 20 constraints, the algorithm failed to terminate within seven days of continuous execution and was therefore manually stopped.

Table 2. Comparative Results of MSA, ASIMOLP, BOA, and PSA for Individual MOLP Problems

Prob.	m	N	q	MSA CPU (s)	ASIMOLP CPU (s)	BOA CPU (s)	PSA CPU (s)
1	6	3	3	0.146	0.074	0.155	0.152
2	8	4	3	0.244	0.076	0.190	0.155
3	15	10	3	1.041	0.171	0.243	0.194
4	25	15	3	91.450	0.211	0.584	0.266
5	30	15	3	125,416.240	0.253	0.775	0.323
6	35	20	3	*	0.335	1.001	0.688
7	20	40	3	*	0.275	0.712	0.536
8	15	45	3	*	0.120	0.471	0.287
9	30	50	3	*	0.251	0.958	0.613
10	30	60	3	*	0.159	0.689	0.414

Table 2a. Most Preferred Solutions Returned by MSA, ASIMOLP, BOA, and PSA

Prob.	Algorithm	Objective Function Values
1	MSA	(−2.00, 10.00, −5.00)
	ASIMOLP	(−1.98, 5.80, −2.89)
	BOA	(−2.00, 10.00, −5.00)
	PSA	(−2.00, 10.00, −5.00)
2	MSA	(−42.50, −42.50, −10.00)
	ASIMOLP	(−34.98, −38.58, −28.50)
	BOA	(−15.00, 15.00, −75.00)
	PSA	(−15.00, 15.00, −75.00)
3	MSA	(223.09, −496.23, −346.64)

	ASIMOLP	(59.42, −357.21, −356.48)
	BOA	(223.83, −501.78, −357.22)
	PSA	(269.75, −449.58, −359.66)
4	MSA	(−363.82, −33.70, −136.71)
	ASIMOLP	(−107.15, −169.94, −166.26)
	BOA	(−363.82, −33.70, −136.71)
	PSA	(−111.87, −39.15, −11.18)
5	MSA	(−260.62, −33.70, −136.71)
	ASIMOLP	(−247.63, −208.93, −270.71)
	BOA	(−236.52, −26.73, −137.31)
	PSA	(−70.41, −119.70, −96.84)
6	ASIMOLP	(−163.40, −265.94, −266.47)
	BOA	(−363.82, −47.30, −136.71)
	PSA	(−111.87, −41.10, −11.18)
7	ASIMOLP	(−72.23, −63.77, −41.14)
	BOA	(−108.70, −78.35, −17.77)
	PSA	(−45.50, 40.50, −14.50)
8	ASIMOLP	(−39.52, −93.22, −92.02)
	BOA	(−44.80, −23.60, −200.80)
	PSA	(−44.80, −23.60, −200.80)
9	ASIMOLP	(−56.74, −68.66, −65.62)
	BOA	(−160.22, −92.15, −23.77)
	PSA	(−21.44, 30.02, −26.54)
10	ASIMOLP	(−23.12, −69.40, −203.25)
	BOA	(45.11, −24.58, −283.42)
	PSA	(45.11, −24.58, −283.43)

****MSA did not produce solutions for Problems 6–10 because execution was terminated after seven days.**

This behaviour is consistent with the findings of Benson (1998) and Rudloff *et al*. (2015), who noted that decision-space enumeration methods such as the Evans–Steuer algorithm become computationally expensive when the number of efficient extreme points grows rapidly. Since the MSA attempts to generate all efficient extreme-point solutions, its computational burden increases significantly for medium- and large-scale MOLP problems.

In contrast, the ASIMOLP algorithm demonstrated remarkable computational efficiency across all test problems. The CPU times remained consistently low, ranging from 0.074 to 0.335 seconds regardless of problem size. The superior performance of ASIMOLP can be attributed to its interior-point search strategy, which avoids the exhaustive enumeration of efficient extreme points and instead converges directly to a preferred efficient solution (Arbel, 1993). These findings agree with previous observations by Lin *et al*. (2006), who reported that interior-point approaches are generally more efficient than simplex-based methods for large optimization problems.

The PSA also exhibited excellent computational performance. Although its execution times were generally higher than those of ASIMOLP, they remained relatively low for all problem instances considered. CPU times ranged from 0.152 seconds for the smallest problem to 0.688 seconds for the largest problem. The efficiency of the PSA can be attributed to its strategy of generating only a representative subset of efficient extreme points rather than enumerating all efficient solutions (Rudloff *et al*., 2015). Consequently, the PSA emerged as the second most computationally efficient algorithm among the four methods investigated.

The performance of BOA was intermediate between ASIMOLP and PSA on one hand and MSA on the other. Although BOA required more computational time than ASIMOLP for most instances, it consistently outperformed the MSA and remained computationally tractable even for the largest problems. This observation supports the conclusions of Benson (1998) and Löhne (2011), who demonstrated that objective-space methods are particularly effective for solving large-scale MOLP problems because their performance depends more strongly on the number of objectives than on the number of efficient extreme points.

4.2 Solution Quality Analysis

In addition to computational efficiency, the quality of the solutions generated by the algorithms was also examined. The results indicate that BOA consistently produced the most comprehensive representation of the efficient frontier because it generates all nondominated extreme points and extreme directions of the MOLP problem (Benson, 1998; Löhne, 2011). Consequently, BOA provides decision makers with the largest set of efficient alternatives from which a preferred solution can be selected.

The PSA also produced high-quality solutions because it generates a subset of efficient extreme points sufficient to reconstruct the entire efficient frontier (Rudloff *et al*., 2015). However, unlike BOA, it does not explicitly generate all nondominated points.

The MSA similarly identifies efficient extreme-point solutions but suffers from severe computational limitations as problem size increases. Therefore, although the quality of the solutions generated by MSA is theoretically comparable to that of BOA for smaller problems, its practical applicability is limited for large-scale MOLP instances.

ASIMOLP differs fundamentally from the other algorithms because it is designed to generate a single preferred efficient solution rather than the entire efficient frontier (Arbel, 1993). Consequently, while it is the most computationally efficient method among those investigated, it provides less information regarding the overall structure of the nondominated set.

4.3 Summary of Findings

Overall, the numerical experiments demonstrate that ASIMOLP is the most computationally efficient algorithm among the four approaches investigated. The PSA also exhibits excellent computational performance and consistently produces results close to those obtained by ASIMOLP. BOA offers a good compromise between computational efficiency and solution quality by generating the complete nondominated set while remaining computationally tractable for larger problems. In contrast, the MSA exhibits poor scalability and becomes impractical for medium- and large-scale MOLP problems.

Based on the results obtained, ASIMOLP is recommended when rapid identification of a preferred efficient solution is required, whereas BOA is the preferred approach when a complete characterization of the efficient frontier is desired. PSA represents an attractive alternative when both computational efficiency and accurate representation of the efficient frontier are important considerations.

4.4 Discussion of Results

As shown in Table 2, the computational time required by the Multiobjective Simplex

Algorithm (MSA) increased significantly as the size of the problem grew. For Problems 1–5, the CPU time rose from 0.146 seconds to 125,416.24 seconds as the number of variables and constraints increased. Furthermore, for Problems 6–10, which involved larger dimensions, the MSA was unable to produce a solution even after seven days of continuous execution and was therefore terminated. This behaviour indicates that the algorithm is highly sensitive to problem size and may not be suitable for medium- to large-scale MOLP problems.

The computational performance of the remaining algorithms—ASIMOLP, BOA, and PSA—appears to depend not only on problem size but also on the structural characteristics of the constraint and objective matrices. Among the four methods, the Affine Scaling Interior MOLP Algorithm (ASIMOLP) consistently recorded the lowest CPU times across all test instances, demonstrating superior computational efficiency. The Parametric Simplex Algorithm (PSA) also exhibited excellent performance and produced CPU times that were generally closest to those obtained by ASIMOLP. In contrast, Benson's Outer Approximation Algorithm (BOA) required slightly longer computation times than ASIMOLP and PSA but remained substantially more efficient than the MSA for the majority of the test problems.

With respect to solution quality, the results indicate that BOA generally produced the most desirable nondominated solutions. This outcome is expected because BOA operates directly in the objective space and generates the complete set of nondominated extreme points, thereby providing a more comprehensive representation of the efficient frontier. Consequently, BOA offers greater flexibility for identifying high-quality solutions and is superior to MSA, PSA, and ASIMOLP in terms of the overall quality of the solutions returned.

5.0 Conclusion

This study presented and compared four of the most widely used solution approaches for Multi-Objective Linear Programming (MOLP): the Multiobjective Simplex Algorithm (MSA), the Parametric Simplex Algorithm (PSA), the Affine Scaling Interior MOLP Algorithm (ASIMOLP), and Benson's Outer Approximation Algorithm (BOA). A review of the relevant literature was provided, followed by a computational comparison of the algorithms using ten MOLP test instances, comprising both benchmark and randomly generated problems with three objective functions.

The numerical results revealed substantial differences in computational efficiency among the algorithms. The MSA performed satisfactorily on small-scale problems but experienced a rapid increase in computational time as problem size increased, ultimately becoming impractical for larger instances. In contrast, ASIMOLP consistently exhibited the best computational performance, producing solutions in the shortest CPU times across all test problems considered. The PSA also demonstrated strong computational efficiency and emerged as a competitive alternative, particularly for medium-sized problems. BOA, while generally requiring more computation time than ASIMOLP and PSA, remained considerably more efficient than MSA.

In terms of solution quality, BOA produced the most comprehensive and representative nondominated solutions because it operates directly in the objective space and generates the complete set of nondominated extreme points and directions. Consequently, BOA proved superior to MSA, PSA, and ASIMOLP with respect to the quality and completeness of the solutions obtained.

Overall, the results suggest that ASIMOLP is the preferred choice when computational speed is the primary concern, whereas BOA is more appropriate when the objective is to obtain high-quality and comprehensive representations of the efficient frontier. The PSA offers a balanced compromise between computational efficiency and solution quality, while the MSA is best suited to relatively small MOLP instances due to its sensitivity to problem dimensionality.

6.0 References

Aksoy, Y., Butler, T. W., & Minor, E. D. (1996). Comparative studies in interactive multiple objective mathematical programming. *European Journal of Operational Research*, 89, 2, pp. 408–422.

Alves, M. J., Antunes, C. H., & Clímaco, J. (2015). Interactive MOLP explorer: A graphical-based computational tool for teaching and decision support in multi-objective linear programming models. *Computer Applications in Engineering Education*, 23, 2, pp. 314–326.

Arbel, A. (1993). A weighted-gradient approach to multi-objective linear programming problems using the analytic hierarchy process. *Mathematical and Computer Modelling*, 17, 4, pp. 27–39.

Benayoun, R., De Montgolfier, J., Tergny, J., & Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, 1, 1, pp. 366–375.

Benson, H. P. (1998). An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13, 1, pp. 1–24.

Benson, H. P. (1998). Hybrid approach for solving multiple-objective linear programs in outcome space. *Journal of Optimization Theory and Applications*, 98, 1, pp. 17–35.

Brockhoff, K. (1985). Experimental test of MCDM algorithms in a modular approach. *European Journal of Operational Research*, 22, 2, pp. 159–166.

Buchanan, J. T., & Daellenbach, H. G. (1987). A comparative evaluation of interactive solution methods for multiple objective decision models. *European Journal of Operational Research*, 29, 3, pp. 353–359.

Dell, R. F., & Karwan, M. H. (1990). An interactive MCDM weight space reduction method utilizing a Tchebycheff utility function. *Naval Research Logistics*, 37, 2, pp. 263–277.

Dyer, J. S. (1972). *An empirical investigation of a man-machine interactive approach to the solution of the multiple criteria problem*. Western Management Science Institute, University of California.

Ehrgott, M. (2006). *Multicriteria optimization*. Springer Science & Business Media.

Ehrgott, M., Shao, L., & Schöbel, A. (2011). An approximation algorithm for convex multi-objective programming problems. *Journal of Global Optimization*, 50, 3, pp. 397–416.

Evans, J. P., & Steuer, R. E. (1973). A revised simplex method for linear multiple objective programs. *Mathematical Programming*, 5, 1, pp. 54–72.

Franz, L. (1980). *Bicriteria decision maker preference structure for multicriteria optimization through interactive goal programming* (Doctoral dissertation). University of Nebraska–Lincoln.

Gale, D., Kuhn, H. W., & Tucker, A. W. (1951). Linear programming and the theory of games. In T. C. Koopmans (Ed.), *Activity analysis of production and allocation* (pp. 317–329). John Wiley & Sons.

Geoffrion, A. M. (1967). Solving bicriterion mathematical programs. *Operations Research*, 15, 1, pp. 39–54.

Geoffrion, A. M., Dyer, J. S., & Feinberg, A. (1972). An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 19, 4, pp. 357–368.

Gibson, M., Bernardo, J. J., Chung, C., & Badinelli, R. (1987). A comparison of interactive multiple-objective decision making procedures. *Computers & Operations Research*, 14, 2, pp. 97–105.

Haimes, Y. Y., & Hall, W. A. (1974). Multiobjectives in water resource systems analysis: The surrogate worth trade-off method. *Water Resources Research*, 10, 4, pp. 615–624.

Haksever, C., & Ringuest, J. L. (1990). Computational efficiency and interactive MOLP algorithms: An implementation of the SIMOLP procedure. *Computers & Operations Research*, 17, 1, pp. 39–50.

Junior, H., & Lins, M. P. E. (2009). A win-win approach to multiple objective linear programming problems. *Journal of the Operational Research Society*, 60, 5, pp. 728–733.

Lin, C., Chen, C., & Chen, P. (2006). On the modified interior point algorithm for solving multi-objective linear programming problems. *International Journal of Information and Management Sciences*, 17, 1, pp. 107–120.

Löhne, A. (2011). *Vector optimization with infimum and supremum*. Springer Science & Business Media.

Löhne, A. (2012). *Bensolve: VLP solver (Version 1.2)*. Retrieved from www.bensolve.org

Lotfi, V., Yoon, Y. S., & Zionts, S. (1997). Aspiration-based search algorithm (ABSALG) for multiple objective linear programming problems: Theory and comparative tests. *Management Science*, 43, 8, pp. 1047–1059.

Malakooti, B., & Ravindran, A. (1986). Experiments with an interactive paired comparison simplex method for MOLP problems. *Annals of Operations Research*, 5, 1–4, pp. 575–597.

Michalowski, W., & Szapiro, T. (1992). A bi-reference procedure for interactive multiple criteria programming. *Operations Research*, 40, 2, pp. 247–258.

Nyiam, P. B. (2019). *Multi-objective linear programming revisited: Exact and approximate approaches* (PhD thesis). University of Essex.

Nyiam, P. B. & Salhi A. (2021). On the Simplex, Interior Point and Objective Space Approaches to Multi-objective linear programming. *Journal of Algorithms & Computational Technology*, 15 (2021), 17483026211008414.

Pandian, P., & Jayalakshmi, M. (2013). Determining efficient solutions to multiple objective linear programming problems. *Applied Mathematical Sciences*, 7, 26, pp. 1275–1282.

Posner, M. E., & Wu, C.-T. (1981). Linear max-min programming. *Mathematical Programming*, 20, 1, pp. 166–172.

Reeves, G. R., & Franz, L. S. (1985). A simplified interactive multiple objective linear programming procedure. *Computers & Operations Research*, 12, 6, pp. 589–601.

Reeves, G. R., & Gonzalez, J. J. (1989). A comparison of two interactive MCDM procedures. *European Journal of Operational Research*, 41, 2, pp. 203–209.

Rudloff, B., Ulus, F., & Vanderbei, R. (2015). A parametric simplex algorithm for linear vector optimization problems. *Mathematical Programming*, pp. 1–30.

Saaty, T. L. (1980). *The analytic hierarchy process: Planning, priority setting, resource allocation*. McGraw-Hill.

Schönfeld, K. P. (1964). *Effizienz und Dualität in der Aktivitätsanalyse* (Doctoral dissertation). Freie Universität Berlin, Germany.

Schönfeld, K. P. (1970). Some duality theorems for the non-linear vector maximum problem. *Unternehmensforschung*, 14, 1, pp. 51–63.

Steuer, R. E. (1977). An interactive multiple objective linear programming procedure. *TIMS Studies in the Management Sciences*, 6, pp. 225–239.

Steuer, R. E., & Choo, E.-U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26, 3, pp. 326–344.

Wallenius, J. (1975). Comparative evaluation of some interactive approaches to multicriterion optimization. *Management Science*, 21, 12, pp. 1387–1396.

Wen, U.-P., & Weng, W.-T. (1998). An interior algorithm for solving multiobjective linear programming problem. *Proceedings of the Institute for Operations Research and the Management Sciences International Meeting*, Tel Aviv, Israel.

Yan, H., Wei, Q., & Wang, J. (2005). Constructing efficient solutions structure of multiobjective linear programming. *Journal of Mathematical Analysis and Applications*, 307, 2, pp. 504–523.

Zionts, S., & Wallenius, J. (1976). An interactive programming method for solving the multiple criteria problem. *Management Science*, 22, 6, pp. 652–663.

Zionts, S., & Wallenius, J. (1983). An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. *Management Science*, 29, 5, pp. 519–529.

Declarations

Consent for Publication

Not applicable.

Availability of Data

The data used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare that they have no competing interests and no conflict of interest regarding the publication of this manuscript.

Ethical Consideration

Not applicable. This study did not involve human participants, human data, human tissues, or animals.

Funding

The authors declare that no specific funding was received from any public, commercial, or not-for-profit funding agency for the conduct of this research.

Authors' Contributions

Paschal Bisong Nyiam and Abdellah Salhi jointly contributed to the conception, methodology, analysis, interpretation of results, manuscript preparation, review, and approval of the final version of the manuscript.