APPROACHES FOR MULTI-OBJECTIVE COMBINATORIAL OPTIMIZATION PROBLEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BANU LOKMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JUNE 2007

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Çağlar Güven
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Murat Köksalan
Supervisor

Examining Committee Members

Prof. Dr. Ömer Kırca	(METU, IE)	
Prof. Dr. Murat Köksalan	(METU, IE)	
Prof. Dr. Levent Kandiller	(Çankaya Univ., IE)	
Asst. Prof. Dr. Esra Karasakal	(METU, IE)	
Asst. Prof. Dr. Ayten Türkcan	(METU, IE)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Banu Lokman

Signature :

ABSTRACT

APPROACHES FOR MULTI-OBJECTIVE COMBINATORIAL OPTIMIZATION PROBLEMS

LOKMAN, Banu

M.S., Department of Industrial Engineering

Supervisor: Prof. Dr. Murat KÖKSALAN

June 2007, 85 pages

In this thesis, we develop two exact algorithms and a heuristic procedure for Multi-objective Combinatorial Optimization Problems (MOCO). Our exact algorithms guarantee to generate all nondominated solutions of any MOCO problem. We test the performance of the algorithms on randomly generated problems including the Multi-objective Knapsack Problem, Multi-objective Shortest Path Problem and Multi-objective Spanning Tree Problem. Although we showed the algorithms work much better than the previous ones, we also proposed a fast heuristic method to approximate efficient frontier since it will also be applicable for real-sized problems. Our heuristic approach is based on fitting a surface to approximate the efficient frontier. We experiment our heuristic on randomly generated problems to test how well the heuristic procedure approximates the efficient frontier. Our results showed the heuristic method works well.

Keywords: Multiple criteria, combinatorial optimization, efficient solution.

ÖZ

ÇOK AMAÇLI BİLEŞİ OPTİMİZASYONU PROBLEMLERİ İÇİN YAKLAŞIMLAR

LOKMAN, Banu
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Murat KÖKSALAN

Haziran 2007, 85 sayfa

Bu tezde, çok amaçlı bileşi problemleri için kesin çözümler veren iki algoritma ve iyi çözümler veren sezgisel bir yöntem geliştirdik. Geliştirdiğimiz iki algoritma tüm etkin çözümleri tam olarak bulmayı garantilemektedir. Algoritmalarımızın performansını rastgele yarattığımız farklı çok amaçlı bileşi problemleri üzerinde (Çok Amaçlı Sırt Çantası Problemi, Çok Amaçlı En Kısa Yol Problemi ve Çok Amaçlı Kapsayan Ağaç Problemi) değerlendirdik. Algoritmalarımızın performansının daha önceden geliştirilen algoritmalardan iyi olduğunu göstermemize rağmen, gerçek hayat büyüklüğündeki problemlerde de uygulanabilir olması için etkin çözümlerin bulunduğu bölgeyi yaklaşık olarak tanımlayan sezgisel bir yöntem geliştirdik. Aynı çok amaçlı bileşi problemleri üzerinde denemeler yaparak sezgisel yaklaşımımızın etkin çözümleri içeren bölgeyi ne kadar iyi tanımladığını deneysel olarak araştırdık ve sezgisel yaklaşımımızın iyi çalıştığını gösterdik.

Anahtar Kelimeler: Çok kriterli, bileşi optimizasyonu, etkin çözüm

To my family and my love

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to Prof. Dr. Murat Köksalan for his encouragement and support throughout this study. Without his brilliant guidance, the thesis would not have been possible.

I would like to thank jury members for their valuable comments on this thesis.

I am deeply grateful to my father Mehmet Tuna, my mother Latife Tuna, my brother Şakir Tuna and my sister Arzu Tuna for their love and support. Without them, this work could not have been completed.

I would like to thank my roommate Baykal Hafızoğlu for his help, support and cheerful presence. I would also like to thank my friends Nihan Görmez, İbrahim Karahan, Mustafa Baydoğan, Tülin İnkaya and Selin Bilgin for their help and encouragement I received during my thesis.

I am also grateful to my dear friend Pınar Güneş for her love, continued moral support and welcomed presence.

Finally, my special thanks go to my husband Utku Lokman for his endless love, patience, understanding and encouragement through the course of this study. Thanks for giving me a shoulder to lean on whenever I need.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Multiobjective Combinatorial Optimization Problems (MOCO) have been a potential research area for the last few decades due to the multicriteria and combinatorial nature of many real-life problems. Although single objective combinatorial problems have been widely studied, the decision makers (DMs) usually have to deal with conflicting objectives. However, generalizing the results of single objective problems to multiple objectives is not easy. The computational complexity may increase substantially.

Since the number of efficient solutions may be exponential in the problem size and the problem may become computationally intractable, to determine all efficient solutions is not practical especially for realistically large-sized MOCO problems. Therefore, instead of generating all efficient solutions, heuristics are developed in order to approximate the efficient frontier. Furthermore, with the help of the heuristic procedures, it may be more meaningful to find the preferred solutions incorporating decision maker's preferences.

Köksalan (1999) developed a heuristic approach that quickly finds a good hypothetical solution for the DM. The approach is based on fitting several arcs to represent possible locations of efficient solutions. A sample of points are taken on these arcs and based on a known utility function. An efficient solution close to the best hypothetical solution is proposed as a heuristic solution for the problem. Since the procedure utilizes several arcs simultaneously, it is sufficient for each efficient solution to be represented by at least one of the chosen arcs. The proposed heuristic procedure is implemented on a bicriteria scheduling problem and yields good results in negligibly small computational time. Furthermore, the approach is not restricted to bicriteria scheduling problems. It can be generalized for problems with two or more objectives and applied to other MOCO problems. According to the preference information obtained from the DM, a localized search can also be conducted in order to find the preferred efficient solution.

We develop a heuristic method to approximate the efficient frontier for MOCO problems. Our procedure is based on fitting a surface to approximate the efficient frontier similar to the approach developed by Köksalan (1999). We experiment on various MOCO problems, including the Multiobjective Knapsack Problem (MOKP), Multiobjective Shortest Path (MOSP) and Multiobjective Spanning Tree (MOST) Problems.

In order to test how well the heuristic procedure approximates the efficient frontier, we developed two exact algorithms to generate all efficient solutions for MOCO problems. Our first method finds the efficient solutions iteratively by solving a model with increasing number of variables and constraints at each iteration. Our method proposes an improvement to the algorithm developed by Sylva and Crema (2004) by decreasing the number of additional constraints and binary variables. However, the improved algorithm still requires substantial computational effort as the number of efficient solutions increase. Our second method deals with this computational complexity and only two additional constraints are inserted to our model without adding new constraints or binary variables at each iteration. We solve more models but models are much easier in complexity..

Different from many of the previous exact methods, the proposed methods are not restricted to bicriteria problems and can be used for MOCO problems with two or more objectives. Our second exact algorithm to generate all efficient solutions has been tested on a number of random instances of two, three and four-objective problems including knapsack, minimum spanning tree and shortest path problems. Since all efficient solutions of these problems are generated, we also test the performance of our heuristic method of approximating the efficient frontier on these MOCO problems and demonstrate that it works well.

In Chapter 2, we provide background information on approaches to MOCO problems. We develop two different exact methods and demonstrate their performances in Chapter 3. We propose a heuristic procedure to approximate the efficient frontier and we report the experimental results in Chapter 4. We will present our conclusions in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

A number of exact and approximation methods have been developed to solve MOCO problems. The early papers in MOCO usually focused on finding supported efficient solutions. As Ehrgott and Gandibleux (2000) argues, weighted linear combination of objectives, the most popular exact method, can be used to generate all supported efficient solutions by means of varying the weight factors.

The computational complexity of MOCO problems gets worse due to the unsupported solutions. These solutions cannot be obtained by using a weighted linear combination of objectives. Furthermore, generating all supported efficient solutions may not be easy especially for large-sized MOCO problems.

The two phase methods provide a general framework for the problem of generating all efficient solutions of the biobjective MOCO problem as Ehrgott and Gandibleux (2000) argues. In the first phase, all supported efficient solutions are generated using the weighted sum scalarization. In the second phase, all unsupported efficient solutions are obtained by employing problem specific techniques. The two phase method has been modified and applied to several biobjective combinatorial problems.

Visée et al. (1998) proposed a two phase method and branch and bound procedures for the biobjective knapsack problem. Ramos et al. (1998) developed a two phase method to generate all efficient trees for the biobjective MOST problem. Steiner and Radzik (2003) also proposed a two phase algorithm for the biobjective MOST. According to Ehrgott and Gandibleux (2000), the majority of exact methods employed to generate all unsupported efficient solutions as well as the supported efficient solutions utilize the same idea with the two phase method except from the algorithms developed for shortest path problem. They also point out the fact that most of these exact methods are restricted to two objectives and cannot be adapted to multiple objectives.

There also exist exact algorithms especially for the multiobjective shortest path (MOSP) problem adapted from the single objective methods. Martins (1984) proposed an algorithm based on the label setting method to generate all efficient paths of MOSP problem. Martins (1984) tested the performance of the algorithm on MOSP problem with two and four objectives. Tung and Chew (1992) developed an exact algorithm for MOSP problem which is a generalization of the label correcting method for the classical shortest path problem. Tung and Chew (1992) applied their algorithm on MOSP problem with three objectives. Guerriero and Musmanno (2001) also developed a label correcting method to generate the entire set of efficient paths. Guerriero and Musmanno (2001) implemented the algorithm on MOSP with two, three and four objectives. Corley (1985) proposed an algorithm for MOST problem, which is a generalization of Prim's algorithm. However, these proposed algorithms are problem specific since they generalize the classical shortest path methods for the shortest path problem. Therefore, these exact methods to generate all efficient paths cannot be applied to other MOCO problems.

Sylva and Crema (2004) developed an exact algorithm for generating all efficient solutions for multiple objective integer linear programs (MOILP). The process of generating all efficient solutions starts with the selection of a positive weight vector. Taking the linear combination of objectives by using this weight vector, the ILP problem is solved. For each efficient solution found, the model is revised by adding new constraints and binary variables and solved to obtain a new efficient solution. Since addition of new constraints and binary variables for each incoming efficient solution increase the complexity of the problem considerably, they also propose a method to generate a subset of efficient solutions for relatively large-scaled problems and they state that it can also be useful for interactive methods.

The algorithm of Sylva and Crema (2004) includes the full enumeration of the set of efficient solutions which may be impossible especially for large-sized problems. Therefore, Sylva and Crema (2007) have proposed a new algorithm in order to find a well-dispersed subset of efficient solutions for multiple objective mixed integer linear programs (MOMILP). The approach is based on the procedure developed by Sylva and Crema (2004).

Due to the computational complexity of the all proposed exact methods, the last decade witnessed a growing interest in the development and improvement of the approximation methods, heuristics and metaheuristics, as discussed in the bibliography by Ehrgott and Gandibleux (2000).

Phelps and Köksalan (2003) proposed an interactive evolutionary metaheuristic for MOCO problems which is tested on the MOST problem and MOKP with two, three and four objectives. The proposed method handles the computational complexity of MOCO by interacting with the DM to guide the search effort toward the preferred solutions.

Zitzler and Thiele (1999) developed an evolutionary algorithm (EA) for MOCO by combining some features belonging to previously proposed EA's. Zitzler and Thiele (1999) also provided a comparison of some selected EA's by taking the MOKP as a basis. Zitzler and Thiele (1999) tested the performance of the EA on the MOKP with two, three or four objectives.

Ulungu et al. (1999) proposed a multiobjective simulated annealing method to approximate the efficient frontier of MOCO problems. Ulungu et al. (1999) implemented the algorithm on the biobjective knapsack problem. The adaptation of the proposed algorithm to other MOCO problems requires some problem specific preliminary work.

Hamacher and Ruhe (1994) developed a heuristic procedure based on the two phase procedure to approximate the efficient frontier of the biobjective spanning tree problem. After obtaining efficient supported efficient trees in the first phase, they employed neighborhood search to generate representative solutions. However, the generalization of the proposed algorithm for MOST problem with more than two objectives and application of the method on other MOCO problems may not be possible.

Zhou and Gen (1999) developed a genetic algorithm (GA) for MOST problem. The proposed GA approach aims to obtain a subset of efficient solutions close to ideal point as much as possible. The genetic algorithm also tries to generate solutions distributed along the Pareto frontier to provide enough alternatives for the DM. Zhou and Gen (1999) tested the performance of the GA on the biobjective spanning tree problem.

Hamacher et al. (2006) proposed two algorithms to determine a representative subset of the efficient solution set for discrete bicriterion problem considering several quality measures. Although the algorithm may be applied to many biobjective combinatorial problems to approximate the efficient frontier, the extension of the algorithm for the problems with more than two objectives may be very difficult.

Hansen (1997) also developed a multiobjective tabu search (MOTS) procedure to generate efficient solutions for MOCO problems. Hansen (1997) tested the performance of the algorithm on the knapsack problem with three objectives.

Shukla and Deb (2006) classified some of the previously proposed classical methods to generate multiple efficient solutions according to their working principles. They compared the performance of these classical methods with the evolutionary generating methods on a number of test problems with two, three and four objectives.

Ehrgott and Gandibleux (2000) presented a review for MOCO while Ehrgott and Gandibleux (2004) presented a review of approximation methods for MOCO problems.

Deb (2001 pp. 306-324) proposed several performance metrics to evaluate and compare the quality of approximations of the efficient frontier. Deb (2001) categorized the performance metrics into groups of metrics evaluating closeness to the efficient frontier, metrics evaluating diversity among the efficient solutions and metrics evaluating closeness and diversity in a combined manner.

CHAPTER 3

EXACT ALGORITHMS TO GENERATE ALL NONDOMINATED SOLUTIONS

We propose two exact algorithms, Algorithm-1 and Algorithm-2, to find all nondominated solutions of MOCO problems. After we discuss our propositions and findings corresponding to these methods, we present the steps of each algorithm. We test the performance of Algorithm-1 and Algorithm-2 on MOKP, MOSP and MOST problems.

3.1 Definitions and Theorems

Without loss of generality, any MOCO problem can be stated as:

$$
\begin{aligned}
&(P)\\
&\text{"}Max\text{"}\left\{z_1(x), z_2(x), \ldots, z_q(x)\right\}\\
&\textit{subject to}\\
&x \in X
\end{aligned}
$$

$$
\begin{aligned}
&\textit{where}\\
&z_i(x) = i^{th}\ \textit{objective function}\\
&x : \textit{decision vector}\\
&X : \textit{solution space}\\
&q : \textit{the number of objective functions}
\end{aligned}
$$

Problem (P) usually does not have a unique solution due to the conflicting objective vectors. $(z_1', z_2', \ldots, z_q')$ is said to dominate $(z_1'', z_2'', \ldots, z_q'')$ if $z_i' \geq z_i''$ for all i and $z_i' > z_i''$ for at least one i. If there does not exist a decision vector x' satisfying the above conditions, then $(z_1'', z_2'', \ldots, z_P'')$ is said to be nondominated and the corresponding decision vector x'' is said to be an efficient solution.

The following theorem is a well known theorem.

Theorem 1. If $\lambda_i > 0$ for all $i = 1, 2, ..., q$ and $(z_1*, z_2*, ..., z_q*)$ is the objective vector corresponding to the optimal solution $x*$ of (P_λ), then $x*$ is an efficient solution to problem (P) and $(z_1*, z_2*, ..., z_q*)$ is a nondominated objective vector of (P).

$$
\begin{aligned}
&(P_\lambda) \\
&Max \sum_{i=1}^{q} \lambda_i z_i(x) \\
&\textit{subject to} \\
&x \in X
\end{aligned}
$$

The efficient solutions that are optimal to the problem with weighted linear combination of objectives are said to be supported efficient solutions.

Algorithm of Sylva and Crema (2004)

The algorithm of Sylva and Crema (2004) generating all nondominated solutions starts with the selection of a positive weight vector $\lambda > 0$. The algorithm terminates if problem (P_λ) is infeasible which implies the problem does not have any efficient solution. If the problem is feasible, model (P_λ) is revised by adding q binary variables and $(q+1)$ constraints which forbid the feasible solutions dominated by the nondominated solution obtained from (P_λ). If we have n nondominated solutions, then we solve problem $P_{\lambda(n)}$ in order to find $(n+1)^{st}$ nondominated solution.

$(P_{\lambda(n)})$

$$Max \sum_{j=1}^{q} \lambda_j z_j(x)$$

subject to

$$z_j(x) \geq \left(z_j^k(x)+1\right)t_{jk} - M_k\left(1-t_{jk}\right) \ \forall j \ \ \forall k$$

$$\sum_{j=1}^{q} t_{jk} \geq 1 \quad \forall k$$

$$t_{jk} \in \{0,1\}$$

$$x \in X$$

$$k = 1,...,n$$

$$j = 1,...,q$$

In problem $P_{\lambda(n)}$, $(z_1^k, z_2^k,..., z_j^k,..., z_q^k)$ denotes the k^{th} nondominated objective vector and M_k denotes the lower bound for $z_k(x)$ and t_{jk} is a binary variable such that:

$$t_{jk} = \begin{cases} 1, & if \quad z_j(x) \geq z_j^k(x)+1 \\ 0, & otherwise \end{cases}$$

The constraint "$\sum_{j=1}^{q} t_{jk} \geq 1$" guarantees that for at least one criterion the optimal solution will have a larger value than k^{th} nondominated solution. That is, the new solution will not be dominated by any of the existing nondominated solutions.

The algorithm keeps adding binary variables and constraints until the problem becomes infeasible. If the problem is infeasible, then we conclude that the number of all nondominated solutions is equal to n.

Propositions for Algorithm 1

Our first algorithm proposes an improvement to this algorithm by decreasing the number of binary variables and constraints inserted to the model iteratively.

We demonstrate our propositions and algorithms on an example. We will work on the following knapsack problem with 15 items and three objectives, $3D-KP_{15}$, which has 29 nondominated solutions as seen in Table 3.1.

$$
\begin{aligned}
&3D-KP_{15}\\
&\text{"}Max\text{"}\left\{z_1(x), z_2(x), z_3(x)\right\}\\
&subject\ to\\
&\sum_{j=1}^{15} w_{ij}x_j \le C_i \quad i=1,2,3\\
&x_j \in \{0,1\}
\end{aligned}
$$

$$
\begin{aligned}
&where\\
&z_i(x)=\sum_{j=1}^{15} p_{ij}x_j\\
&p_{ij} : profit\ of\ item\ j\ for\ knapsack\ i\\
&w_{ij} : weight\ of\ item\ j\ for\ knapsack\ i\\
&C_i : capacity\ of\ knapsack\ i\\
&x_j=\begin{cases}1, & if\ item\ j\ is\ selected\\ 0, & otherwise\end{cases}\\
&C_i=\frac{\sum_{j=1}^{15} w_{ij}x_j}{2}
\end{aligned}
$$

In our first proposition, we claim that we can obtain the nondominated solution with the best value of a selected criterion by selecting the weights properly.

Table 3.1 Nondominated vectors corresponding to the $3D-KP_{15}$

	z_1	z_2	z_3
1	402	469	521
2	420	393	508
3	318	477	487
4	459	455	486
5	376	536	476
6	425	511	475
7	476	426	473
8	469	534	456
9	443	538	446
10	490	464	443
11	482	497	440
12	472	499	429
13	499	468	427
14	422	543	421
15	471	518	420
16	395	551	415
17	500	543	413
18	492	576	410
19	510	428	406
20	508	571	402
21	518	467	400
22	510	500	397
23	515	526	384
24	534	482	381
25	450	587	376
26	536	594	348
27	557	504	346
28	543	549	330
29	569	509	236

Proposition 1. For a sufficiently small $\varepsilon > 0$, the optimal solution to problem (P_i) will give an efficient solution to problem (P) with maximum z_i value.

(P_i)

$$Max\ z_i(x) + \sum_{j \neq i} \varepsilon z_j(x)$$

subject to

$x \in X$

Proof. We know (P_i) will yield an efficient solution to problem (P) since $\varepsilon > 0$ by using Theorem 1. Furthermore, since ε is sufficiently small, problem (P_i) will not sacrifice from the maximum value of z_i, the i^{th} objective function . Therefore, the optimal solution to problem (P_i) will give the efficient solution to the problem (P) with maximum z_i value. $\square$

We should note that ε value is not unique; it is problem dependent. How the value of epsilon should be selected to obtain the nondominated solution with maximum z_i value is discussed by (see Steuer 1986 pp.425) for a Tchebycheff program. It is similar in our case.

We denote this maximum value of z_i as $z_{i(1)}$ and the corresponding nondominated objective function vector as:

$$e_{i(1)} = \left\{ z_{1[i(1)]}, z_{2[i(1)]}, \ldots, z_{i-1\,[i(1)]}, z_{i(1)}, z_{i+1[i(1)]}, \ldots, z_{q[i(1)]} \right\}.$$

Similarly, we denote the z_i value obtained in the k^{th} solution of the problem in the nonincreasing order of z_i values as $z_{i(k)}$ such that $z_{i(k+1)} \le z_{i(k)} \quad \forall k = 1, 2, \ldots, N$ where N is the number of all nondominated vectors. $e_{i(k)} = \left\{ z_{1[i(k)]}, z_{2[i(k)]}, \ldots, z_{i-1\,[i(k)]}, z_{i(k)}, z_{i+1[i(k)]}, \ldots, z_{q[i(k)]} \right\}$ denotes the corresponding objective vector. We denote the set of the nondominated vectors found until iteration n as $S_i(n) = \left\{ e_{i(k)} : 1 \le k \le n \right\}$.

In our example, we select the main objective function's index as three such that $i = 3$. If we take $\varepsilon = 0.0001$, then the model (P_3) will give the nondominated objective vector $e_{3(1)}$=(402, 469, 521). The nondominated vector has maximum z_3 value among all nondominated vectors as seen in Table 3.2, which demonstrates our proposition. If we selected $\varepsilon = 0$, then it would be possible to find an inefficient solution different from the solutions in Table 3.1.

According to our definition, $e_{3(k)}$ denotes the nondominated solution with k^{th} best value of z_3 as also indicated in Table 3.2. Furthermore, we define the set $S_3(n) = \left\{ e_{3(k)} : 1 \le k \le n \right\}$ such that it includes n nondominated solutions with the best z_3 values. For instance, the set $S_3(2)$ includes the solutions $\left\{ e_{3(1)}, e_{3(2)} \right\}$ in the following table such that $S_3(2) = \left\{ (402, 469, 521), (420, 393, 508) \right\}$.

In proposition 1, we showed that we can obtain the nondominated solution with the best value of the selected criterion. On the other hand, Proposition 2 claims that we can find the nondominated solution with $(n+1)^{st}$ best value of the selected criterion by using the nondominated solutions with n best values of the selected criterion.

Table 3.2 Verification of Proposition 1 on $3D-KP_{15}$

	z_1	z_2	z_3	$z_3+\varepsilon z_1+\varepsilon z_2$
$e_{3(1)}$	402	469	521	521.09
$e_{3(2)}$	420	393	508	508.08
$e_{3(3)}$	318	477	487	487.08
$e_{3(4)}$	459	455	486	486.09
$e_{3(5)}$	376	536	476	476.09
$e_{3(6)}$	425	511	475	475.09
$e_{3(7)}$	476	426	473	473.09
$e_{3(8)}$	469	534	456	456.10
$e_{3(9)}$	443	538	446	446.10
$e_{3(10)}$	490	464	443	443.10
$e_{3(11)}$	482	497	440	440.10
$e_{3(12)}$	472	499	429	429.10
$e_{3(13)}$	499	468	427	427.10
$e_{3(14)}$	422	543	421	421.10
$e_{3(15)}$	471	518	420	420.10
$e_{3(16)}$	395	551	415	415.09
$e_{3(17)}$	500	543	413	413.10
$e_{3(18)}$	492	576	410	410.11
$e_{3(19)}$	510	428	406	406.09
$e_{3(20)}$	508	571	402	402.11
$e_{3(21)}$	518	467	400	400.10
$e_{3(22)}$	510	500	397	397.10
$e_{3(23)}$	515	526	384	384.10
$e_{3(24)}$	534	482	381	381.10
$e_{3(25)}$	450	587	376	376.10
$e_{3(26)}$	536	594	348	348.11
$e_{3(27)}$	557	504	346	346.11
$e_{3(28)}$	543	549	330	330.11
$e_{3(29)}$	569	509	236	236.11

Proposition 2. For a sufficiently small $\varepsilon > 0$ and sufficiently large $M > 0$, if all the nondominated vectors in set $S_i(n) = \{e_{i(k)} : 1 \le k \le n\}$ are known, then the optimal solution to $(P_{i(n)})$ will give the nondominated objective function vector, $e_{i(n+1)}$, with $(n+1)^{th}$ best z_i value. If $(P_{i(n)})$ is infeasible, then set $S_i(n) = \{e_{i(k)} : 1 \le k \le n\}$ is the entire set of nondominated vectors.

$$
\begin{aligned}
&(P_{i(n)}) \\
&Max\ z_i + \sum_{j \ne i} \varepsilon z_j \\
&subject\ to \\
&z_j \ge z_{j[i(k)]} + 1 - M\left(1 - t_{jk}\right)\ \forall j \ne i \quad \forall k \qquad (1) \\
&\sum_{j \ne i} t_{jk} = 1\ \forall k \qquad (2) \\
&t_{jk} \in \{0,1\} \\
&x \in X \\
&k = 1,\dots,n \\
&j = 1,\dots,q
\end{aligned}
$$

$$
where \quad t_{jk} = \begin{cases} 1, & if \quad z_j \ge z_j^k(x) + 1 \\ 0, & otherwise \end{cases}
$$

Proof. Let us first consider the case $n = 1$ where we know only the nondominated vector, $e_{i(1)}$. Since summation of t_{j1} is equal to 1, exactly one of the constraints "$z_j \ge z_{j[i(1)]} + 1 - M\left(1 - t_{j1}\right)$" will be binding and the others will be redundant for sufficiently large M value. Therefore, at least one objective function value of the new efficient solution will be strictly greater than its value in $e_{i(1)}$ which guarantees a different nondominated vector. Since our aim is to maximize z_i as much as possible and guarantee to obtain a different efficient solution, we will obtain the nondominated vector, $e_{i(2)}$, with the second best z_i. In case of infeasibility, we conclude that there is only one nondominated objective vector.

Similarly, for $n > 1$, "$z_j \geq z_{j[i(k)]} + 1 - M(1 - t_{jk})$" guarantees that the new efficient solution will be different from all the efficient solutions in set $S_i(n)$. Since the model will try to maximize z_i as much as possible where n-best nondominated vectors of z_i are forbidden by the constraints with "$\sum_{j \neq i} t_{jk} = 1$", we will obtain the nondominated vector, $e_{i(n+1)}$, with the $(n+1)^{th}$ best z_i. If the problem is infeasible, we can conclude that $S_i(n) = \{e_{i(k)} : 1 \leq k \leq n\}$ is the entire set of nondominated vectors. □

Let us go back to our example problem $3D - KP_{15}$. Since we know $S_3(1) = e_{3(1)}$=(402, 469, 521), we can write the corresponding $P_{3(1)}$ model as the following:

$$
\begin{aligned}
&(P_{3(1)})_{3D-KP_{15}} \\
&Max\ z_3 + \varepsilon z_1 + \varepsilon z_2 \\
&subject\ to \\
&z_1 \geq 402 + 1 - M + Mt_{11} \\
&z_2 \geq 469 + 1 - M + Mt_{21} \\
&t_{11} + t_{21} = 1 \\
&t_{11}, t_{21} \in \{0,1\} \\
&x \in X \\
&k = 1, \ldots, n
\end{aligned}
$$

Since exactly one of the t variables will take the value of one, the nondominated vector $e_{3(1)}$=(402, 469, 521) will be infeasible. Then, model $(P_{3(1)})_{3D-KP_{15}}$ will give the feasible nondominated vector $e_{3(2)}$=(420, 393, 508) since it has the largest z_3 value among all other feasible solutions. If we consider the problem $(P_{3(29)})_{3D-KP_{15}}$, then the problem will be infeasible since all the nondominated solutions in set $S_3(29)$ are forbidden by the related constraints which implies our problem $3D - KP_{15}$ has 29 nondominated objective vectors.

Corollary 1. All nondominated solutions to problem (P) can be generated by solving $(P_{i(n)})$ iteratively until the model becomes infeasible.

3.2 Algorithm 1

Step 0. Initialization

Let $W = \varnothing$ and $n = 0$ where W is the set of nondominated vectors and $n = |W|$.

Step 1. Solve model (P_i).

(P_i)

$$Max\ z_i + \sum_{j \neq i} \varepsilon z_j$$

subject to

$x \in X$

If the problem is feasible, denote the optimal objective vector as $e_{i(1)}$ and let W = $\{e_{i(1)}\} = S_i(1)$ and $n \leftarrow n+1$. Go to Step 2. Otherwise, stop. The problem does not have a feasible solution.

Step 2. Solve the model $(P_{i(n)})$.

$(P_{i(n)})$

$$Max\ z_i + \sum_{j \neq i} \varepsilon z_j$$

subject to

$$z_j \geq z_{j[i(k)]} + 1 - M + Mt_{jk} \quad \forall j \neq i \quad \forall k \qquad (1)$$

$$\sum_{j \neq i} t_{jk} = 1 \quad \forall k \qquad (2)$$

$$t_{jk} \in \{0,1\} \quad \forall j \neq i \quad k = 1,..,n$$

$x \in X$

If the problem is feasible, denote its optimal objective vector as $e_{i(n+1)}$ and let $W = \{e_{i(n+1)}\} \cup W = S_i(n+1)$ and $n \leftarrow n+1$. Repeat Step 2.

If the problem $(P_{i(n)})$ is infeasible, go to Step 3.

Step 3. Stop. W $= S_i(n)$ is the entire set of nondominated vectors for the problem (P) and the number of all nondominated vectors is $n = |W|$.

3.3 Algorithm 2

Our first algorithm improves the algorithm of Sylva and Crema (2004) since the number of binary variables and constraints introduced to the model for each new efficient solution is decreased. The algorithm developed by Sylva and Crema (2004) introduce $(q+1)$ additional constraints and q binary variables at each iteration. On the other hand, we introduce (q) additional constraints and $(q-1)$ binary variables to the problem for each nondominated vector. However, the additional constraints and variables still grow and cause computational difficulty. We develop a new algorithm to further improve Algorithm 1.

3.3.1 Three Criteria Case

We first develop the algorithm for the three criteria case.

Definitions and Theorems

Instead of using binary variables and constraints, we develop a new algorithm employing a sorting and searching mechanism to find the nondominated vectors. Let us order the vectors in $S_i(n)$ such that $e_{i(n)}^{r(j)} = \left(z_{1[i(n)]}^{r(j)}, z_{2[i(n)]}^{r(j)}, z_{3[i(n)]}^{r(j)}\right)$ denotes the vector having $(j-1)$ vectors with r^{th} objective function values less than or equal to that of $e_{i(n)}^{r(j)}$. That is $e_{i(n)}^{r(j)}$ denotes the j^{th} vector when these vectors are ordered in the nondecreasing order of z_r such that $z_{r[i(n)]}^{r(j)} \leq z_{r[i(n)]}^{r(j+1)}$ where $1 \leq j \leq n-1$. Let

$S_i^r(n)$ denote the list of these solutions that are in the nondecreasing order of objective r.

If we consider our problem $3D-KP_{15}$ and we take $n=4$, then set $S_3^1(4)$ includes the nondominated vectors in the set $S_3(4)$, which includes the first 4 nondominated objective vectors with the best z_3 value. While $S_3(4)$ includes nondominated solutions in nonincreasing order of z_3, $S_3^1(4)$ includes the same solutions but in nondecreasing order of z_1 as seen in Table 3.3 and Table 3.4.

Table 3.3 The nondominated vectors of $3D-KP_{15}$ in $S_3(4)$

$S_3(4)$	z_1	z_2	z_3
$e_{3(1)}$	402	469	521
$e_{3(2)}$	420	393	508
$e_{3(3)}$	318	477	487
$e_{3(4)}$	459	455	486

Table 3.4 The nondominated vectors of $3D-KP_{15}$ in $S_3^1(4)$

$S_3^1(4)$	z_1	z_2	z_3
$e_{3(4)}^{1(1)}$	318	477	487
$e_{3(4)}^{1(2)}$	402	469	521
$e_{3(4)}^{1(3)}$	420	393	508
$e_{3(4)}^{1(4)}$	459	455	486

Our first algorithm iteratively solves problem $(P_{i(n)})$ which takes the current set of nondominated vectors, $S_i(n)$, as its input and gives the nondominated vector, $e_{i(n+1)} = \left(z_{1[i(n+1)]}, z_{2[i(n+1)]}, z_{3[i(n+1)]}\right)$, as discussed in Chapter 3.2.

$(P_{i(n)})$

$$\begin{aligned}
&Max\ z_i + \varepsilon z_r + \varepsilon z_a \\
&subject\ to \\
&z_r \geq z_{r[3(k)]} + 1 - Mt_k \quad \forall k \\
&z_a \geq z_{a[3(k)]} + 1 - M(1 - t_k) \quad \forall k \\
&t_k \in \{0,1\} \\
&x \in X \\
&k = 1,...,n \\
&i \neq r \neq a \\
&i, r, a \in \{1,2,3\}
\end{aligned}$$

where

$$t_k = \begin{cases} 1, & if \quad z_a \geq z_{a[3(k)]} + 1 \\ 0, & if \quad z_r \geq z_{r[3(k)]} + 1 \end{cases}$$

In this model, i denotes the main objective function's index, r corresponds to the objective function we will use to order the vectors in $S_i^r(n)$ and a is the index of the remaining objective function. Note that we could differentiate between the M values used in different constraints as M_r and M_a. However, it is sufficient to use a single big M value that is large enough to satisfy all constraint requirements.

Instead of solving model $(P_{i(n)})$ iteratively whose complexity increases considerably as n grows, we solve the following $(n+1)$ models.

$(P_{i(n)}^{r(j)})$

$$\begin{aligned}
&Max\ z_i + \varepsilon z_r + \varepsilon z_a \\
&subject\ to \\
&z_r \geq lb_{z_r}(P_{i(n)}^{r(j)}) \\
&z_a \geq lb_{z_a}(P_{i(n)}^{r(j)}) \\
&x \in X
\end{aligned}$$

where

$j = 0,1,\ldots,n$

$$lb_{z_r}(P_{i(n)}^{r(j)}) = \begin{cases} -M, & if\ \ j=0 \\ z_{r[i(n)]}^{r(j)}+1, & \text{otherwise} \end{cases}$$

$$lb_{z_a}(P_{i(n)}^{r(j)}) = \begin{cases} -M, & if\ \ j=n \\ \max\limits_{n \ge h > j}\left\{z_{a[i(n)]}^{r(h)}\right\}+1, & \text{otherwise} \end{cases}$$

We denote the nondominated vector obtained from problem $(P_{i(n)}^{r(j)})$ as $ce_{i(n+1)}^{r(j)} = \left(cz_{1[i(n+1)]}^{r(j)}, cz_{2[i(n+1)]}^{r(j)}, cz_{3[i(n+1)]}^{r(j)}\right)$, which can also be interpreted as the j^{th} candidate solution.

Note that the nondominated solution with the best $(n+1)^{st}$ z_i value is the best candidate with maximum z_i value as discussed in Proposition 3.

Proposition 3. $e_{i(n+1)} = \left\{ ce_{i(n+1)}^{r(m)} : cz_{i[i(n+1)]}^{r(m)} = \max\limits_{0 \le j \le n}\left\{cz_{i[i(n+1)]}^{r(j)}\right\} \right\}$.

Proof. Without loss of generality, let us take $r=1$, $a=2$ and $i=3$. According to Proposition 2, $(P_{3(n)})$ will give the next unknown nondominated vector $e_{3(n+1)}$. Since the contents of $S_3(n)$ is equivalent to the set $S_3^1(n)$, we can rewrite problem $(P_{3(n)})$ as follows:

$(P'_{3(n)})$

$Max\ z_3 + \varepsilon z_1 + \varepsilon z_2$

subject to

$z_1 \ge z_{1[3(n)]}^{1(j)} + 1 - Mt_j \quad \forall j = 1,\ldots,n$

$z_2 \ge z_{2[3(n)]}^{1(j)} + 1 - M(1-t_j) \quad \forall j$

$x \in X$

$t_j \in \{0,1\}$

Without loss of generality, we can say that exactly one of the following cases will hold for $z_{1[3(n+1)]}$, which is z_1 value of the optimal solution of the problem $(P'_{3(n)})$.

Case (a) If $-M \leq \; z_{1[3(n+1)]} \; \leq \; z^{1(1)}_{1[3(n)]}$, then $z_{2[3(n+1)]} \geq \max\limits_{n \geq h} \left\{ z^{1(h)}_{2[3(n)]} \right\} + 1$.

Case (b) If $z^{1(j)}_{1[3(n)]} + 1 \; \leq \; z_{1[3(n+1)]} \; \leq \; z^{1(j+1)}_{1[3(n)]}$, then $z_{2[3(n+1)]} \geq \max\limits_{n \geq h > j} \left\{ z^{1(h)}_{2[3(n)]} \right\} + 1$.

Case (c) If $z^{1(n)}_{1[3(n)]} + 1 \; \leq \; z_{1[3(n+1)]}$, then $z_{2[3(n+1)]} \geq -M$.

According to the solution space, case (a) corresponds to model $(P^{1(0)}_{3(n)})$, case (b) corresponds to the model $(P^{1(j)}_{3(n)})$ and case (c) corresponds to model $(P^{1(n)}_{3(n)})$. Since the aim is to maximize z_3 as much as possible, all of the possible cases are to be compared according to their z_3 values. Since we define the nondominated vector obtained from problem $(P^{1(j)}_{3(n)})$ as $ce^{1(j)}_{3(n+1)}$, we can write the following equation:

$$e_{3(n+1)} = \left\{ ce^{1(m)}_{3(n+1)} : cz^{1(m)}_{3[3(n+1)]} = \max\limits_{0 \leq j \leq n} \left\{ cz^{1(j)}_{3[3(n+1)]} \right\} \right\} \qquad \square$$

The corresponding j value for the best candidate solution gives the position of the new nondominated solution in the list $S^r_i(n+1)$. Therefore, we do not need to sort the solutions at each iteration as discussed in Corollary 2. If we have the best candidate solution for different values of j, then we select the one with the largest index to determine the position.

Corollary 2. If $e_{i(n+1)} = ce^{r(m)}_{i(n+1)}$ and $e_{i(n+1)} \neq ce^{r(m+1)}_{i(n+1)}$, then $J^{S^r_i(n+1)}_{e_i(n+1)} = m+1$ where $J^{S^r_i(n+1)}_{e_i(n+1)}$ denotes the place of $e_i(n+1)$ in the list $S^r_i(n+1)$.

Proof. Since $e_{i(n+1)} = ce^{r(m)}_{i(n+1)}$ and $e_{i(n+1)} \neq ce^{r(m+1)}_{i(n+1)}$, we know $e_{i(n+1)}$ is the nondominated vector obtained from model $(P^{r(m)}_{i(n)})$. As stated in the proof of Proposition 3, we consider three special cases including:

Case (a) If $m = 0$, then $-M \leq z_{r[i(n+1)]} \leq z^{r(1)}_{r[i(n)]}$ which implies $J^{S_i^r(n+1)}_{e_i(n+1)} = 1$.

Case (b) If $m = j$ $(j \neq 0, j \neq n)$, then $z^{r(j)}_{r[i(n)]} + 1 \leq z_{r[i(n+1)]} \leq z^{r(j+1)}_{r[i(n)]}$ which implies $J^{S_i^r(n+1)}_{e_i(n+1)} = j + 1$.

Case (c) If $m = n$, then $z^{r(n)}_{r[i(n)]} + 1 \leq z_{r[i(n+1)]}$ which implies $J^{S_i^r(n+1)}_{e_i(n+1)} = n + 1$.

Considering all possible cases, if $e_{i(n+1)} = ce^{r(m)}_{i(n+1)}$ and $e_{i(n+1)} \neq ce^{r(m+1)}_{i(n+1)}$, then $J^{S_i^r(n+1)}_{e_i(n+1)} = m + 1$. □

Let us work on our example problem $3D - KP_{15}$ and assume we try to find $e_{3(4)}$ by solving problem $(P_{3(3)})_{3D-KP_{15}}$, where the nondominated solutions in $S^1_3(3)$ are forbidden.

$$
\begin{aligned}
&(P_{3(3)})_{3D-KP_{15}} \\
&Max\ z_3 + \varepsilon z_1 + \varepsilon z_2 \\
&subject\ to \\
&z_1 \geq 318 + 1 - M + Mt_1 \\
&z_2 \geq 477 + 1 - M + M(1 - t_1) \\
&\\
&z_1 \geq 402 + 1 - M + Mt_2 \\
&z_2 \geq 469 + 1 - M + M(1 - t_2) \\
&\\
&z_1 \geq 420 + 1 - M + Mt_3 \\
&z_2 \geq 393 + 1 - M + M(1 - t_3) \\
&\\
&t_k \in \{0,1\} \quad \forall k = 1,2,3 \\
&x \in X
\end{aligned}
$$

where

$S_3(3) = \{(402,469,521),(420,393,508),(318,477,487)\}$,

$S_3^1(3) = \{(318,477,487),(402,469,521),(420,393,508)\}$.

The feasible region corresponding to problem $(P_{3(3)})_{3D-KP_{15}}$ is demonstrated in Figure 3.1 where corresponding third criterion values for (z_1, z_2) pairs are encircled. The known nondominated solutions with three best z_3 values are encircled in this figure. The best candidate solution, nondominated solution with the fourth best z_3 value, is also encircled. The shaded area indicates the dominated region we would like to avoid.

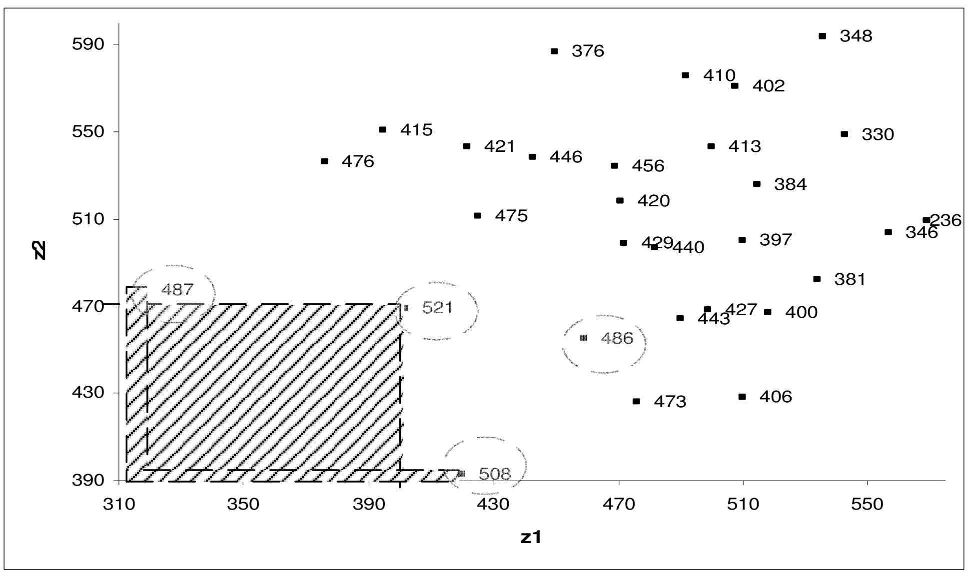


Figure 3.1 Feasible criterion space of $(P_{3(3)})_{3D-KP_{15}}$ and the nondominated solutions

Instead of solving model $(P_{3(3)})_{3D-KP_{15}}$ which has 3 binary variables and 6 constraints, we solve the following 4 models including only two new constraints regardless of the value of n as seen in Figure 3.2

$(P_{3(3)}^{1(0)})_{3D-KP_{15}}$ $Max\ z_3+\varepsilon z_1+\varepsilon z_2$ *subject to* $z_1 \geq -M_1$ $z_2 \geq \max(477,469,393)+1$ $x \in X$	$(P_{3(3)}^{1(1)})_{3D-KP_{15}}$ $Max\ z_3+\varepsilon z_1+\varepsilon z_2$ *subject to* $z_1 \geq 318+1$ $z_2 \geq \max(469,393)+1$ $x \in X$
$(P_{3(3)}^{1(2)})_{3D-KP_{15}}$ $Max\ z_3+\varepsilon z_1+\varepsilon z_2$ *subject to* $z_1 \geq 402+1$ $z_2 \geq \max(393)+1$ $x \in X$	$(P_{3(3)}^{1(3)})_{3D-KP_{15}}$ $Max\ z_3+\varepsilon z_1+\varepsilon z_2$ *subject to* $z_1 \geq 420+1$ $z_2 \geq -M_2$ $x \in X$

Figure 3.2 Problems $(P_{3(3)}^{1(j)})_{3D-KP_{15}}$

We demonstrate the feasible regions corresponding to each problem $(P_{3(3)}^{1(j)})_{3D-KP_{15}}$ in Figures 3.3, 3.4, 3.5 and 3.6. The known nondominated solutions with three best z_3 values and the candidate solutions for each problem are marked on these figures. The shaded areas demonstrate the infeasible regions.

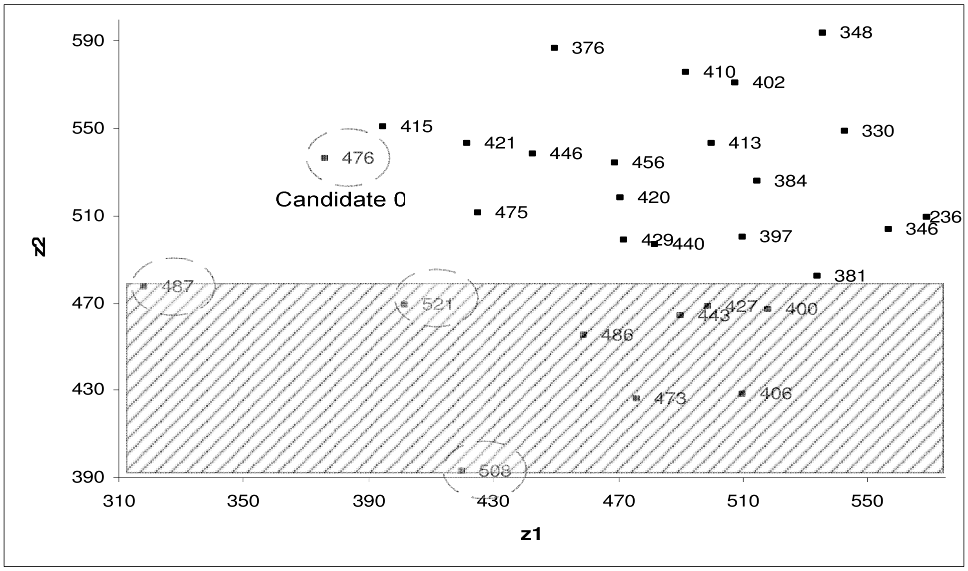


Figure 3.3 Feasible criterion space of $(P_{3(3)}^{1(0)})_{3D-KP_{15}}$ and the candidate solution

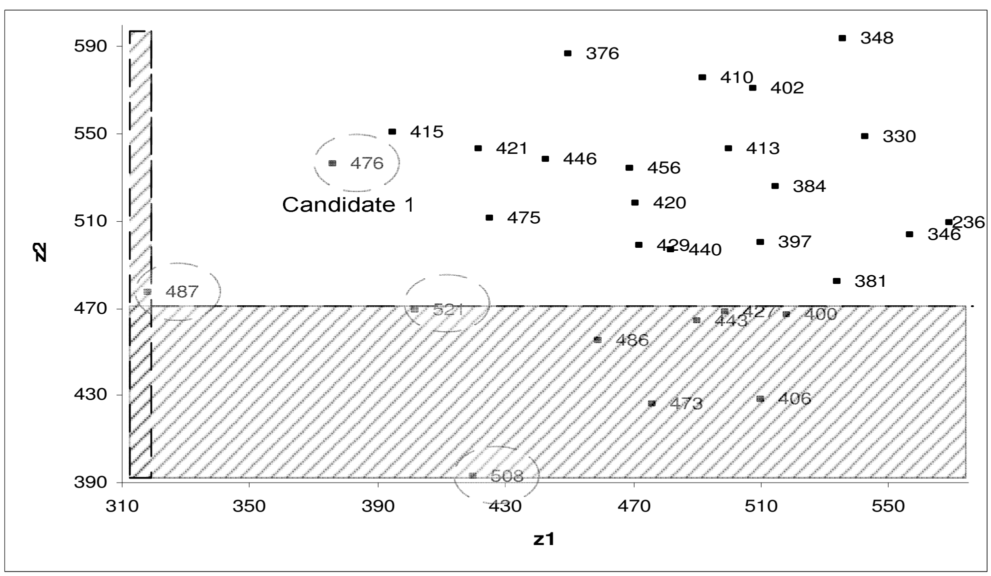


Figure 3.4 Feasible criterion space of $(P_{3(3)}^{1(1)})_{3D-KP_{15}}$ and the candidate solution

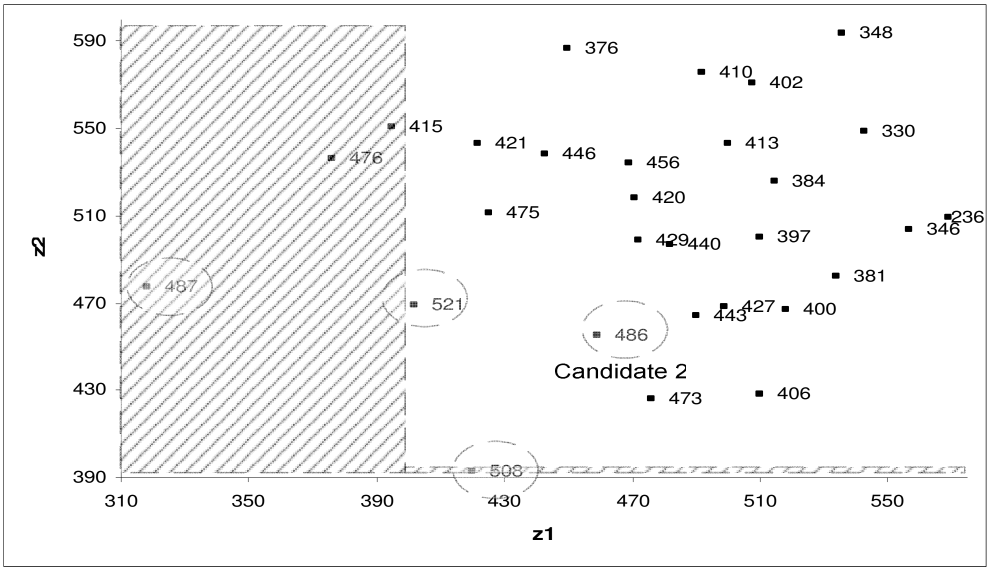


Figure 3.5 Feasible criterion space of $(P_{3(3)}^{1(2)})_{3D-KP_{15}}$ and the candidate solution

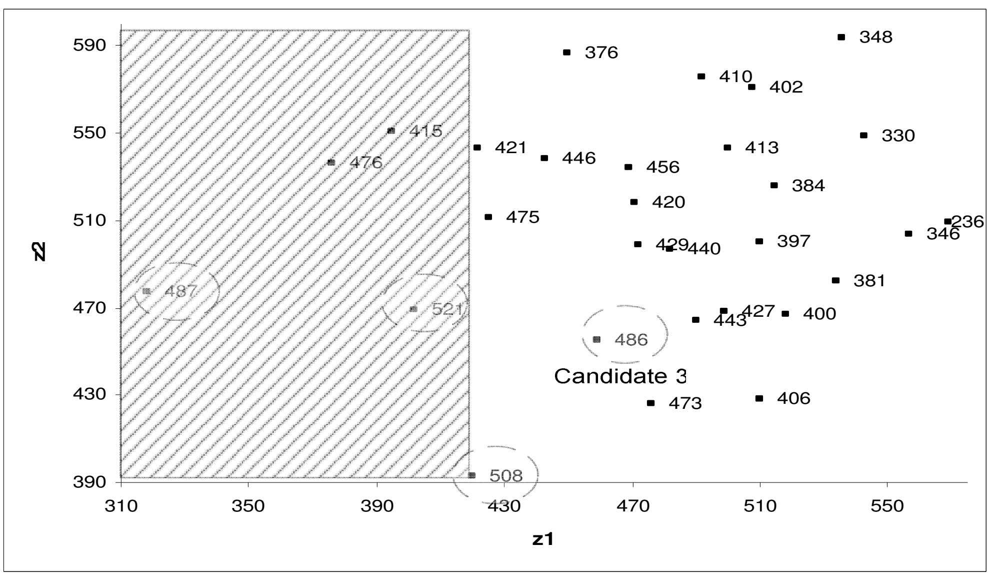


Figure 3.6 Feasible criterion space of $(P_{3(3)}^{1(3)})_{3D-KP_{15}}$ and the candidate solution

Table 3.5 The nondominated vectors corresponding to problems $(P^{1(j)}_{3(3)})_{3D-KP_{15}}$

The problem	Corresponding Nondominated Vector	z_1	z_2	z_3
$(P^{1(0)}_{3(3)})_{3D-KP_{15}}$	$ce^{1(0)}_{3(4)}$	376	536	476
$(P^{1(1)}_{3(3)})_{3D-KP_{15}}$	$ce^{1(1)}_{3(4)}$	376	536	476
$(P^{1(2)}_{3(3)})_{3D-KP_{15}}$	$ce^{1(2)}_{3(4)}$	459	455	486
$(P^{1(3)}_{3(3)})_{3D-KP_{15}}$	$ce^{1(3)}_{3(4)}$	459	455	486

$$e_{3(4)} = \left\{ ce^{1(3)}_{3(4)} : cz^{1(3)}_{3[3(4)]} = \max\{476, 476, 486, 486\} \right\} = (459, 455, 486)$$

Furthermore, $e_{3(4)} = ce^{1(2)}_{3(4)} = ce^{1(3)}_{3(4)}$, $ce^{1(2)}_{3(4)}$ does not satisfy Corollary 2 because $e_{3(4)} = ce^{1(3)}_{3(4)}$ is also true. Then, we set $m = 3$ and insert $e_{3(4)}$ just after the nondominated vector $e^{1(3)}_{3(3)}$ such that $J^{S^1_3(4)}_{e_3(4)} = 3 + 1 = 4$. We should note that this case corresponds to case c since $m = n = 3$ as discussed before. We obtain the following ordered nondominated solutions in $S^1_3(4)$ by using the nondominated vectors in $S^1_3(3)$.

$S^1_3(4) = \{(318, 477, 487), (402, 469, 521), (420, 393, 508), (459, 455, 486)\}.$

Corollary 3. The following algorithm yields all nondominated solutions.

An algorithm for three dimensional problem

Step 0. Initialization

Let $W = \varnothing$ and $n = 0$ where W is the set of nondominated vectors and $n = |W|$.

Step 1. Solve model (P_i).

(P_i)

$Max\ z_i + \sum_{j \neq i} \varepsilon z_j$

subject to

$x \in X$

If the problem is feasible, denote its optimal objective vector as $e_{i(1)} = \left(z^{r(1)}_{1[i(1)]}, z^{r(1)}_{2[i(1)]}, z^{r(1)}_{3[i(1)]}\right)$ and let $W = \{e_{i(1)}\} \cup W$ where $S^r_i(1) = W$ and $e^{r(1)}_{i(1)} = e_{i(1)}$.Go to Step 2.

If the problem is infeasible, stop. The problem does not have any feasible solution.

Step 2. $n \leftarrow n+1$

Step.2.0. Let $j = 0, \quad \max = -M$

Step.2.1. If $j > n$, go to Step 2.5. Otherwise, go to Step 2.2.

Step 2.2. Solve the problem $P^{r(j)}_{i(n)}$. If the problem is feasible, let the objective vector corresponding to the optimal solution as $ce^j_{i(n+1)}$, then go to Step 2.3. Otherwise, go to Step 2.4.

Step 2.3. If $cz^j_{i[i(n+1)]} > \max$, then update $\max = cz^j_{i[i(n+1)]},\ m = j$.

Step 2.4. $j \leftarrow j+1$. Go to Step.2.1.

Step 2.5. If $\max = -M$, go to Step 3. Otherwise, go to Step 2.6.

Step.2.6. Let $e_i(n+1) = ce^{r(m)}_{i(n+1)}$ and insert $e_i(n+1)$ in the position $J^{S_i^r(n+1)}_{e_i(n+1)} = m+1$ of the list $S_i^r(n)$ by changing the positions $J^{S_i^r(n_j)}_{e_i(n_j)} \leftarrow J^{S_i^r(n_j)}_{e_i(n_j)} + 1 \quad for \quad J^{S_i^r(n_j)}_{e_i(n_j)} \geq m+1, 0 < n_j \leq n$. Update $W = e_{i(n+1)} \cup W$ and then repeat Step 2.

Step 3. Stop. $W = S_i^r(n)$ is the entire set of nondominated vectors for the problem (P) and $n = |W|$.

As seen in our algorithm, after solving model (P_i) to find $e_{i(1)}$, we need to solve $n+1$ models in order to find the next nondominated solution $e_i(n+1)$ where $n = 1,...,N-1$. If N is the number of all nondominated vectors of problem (P), then we need to solve $1 + \sum_{n=1}^{N-1}(n+1) = 1 + (2 + ... + N) = \frac{N(N+1)}{2}$ models $(O(N^2))$.

The above algorithm does not transfer any information about the candidate solutions to the following iterations. We may decrease the number of models solved by keeping some information in the memory. In fact, there should be many models yielding the same solution since we have only N nondominated solutions although we solve $\frac{N(N+1)}{2}$ models each of which gives one of the nondominated solutions.

If we go back to our example, both $(P^{1(0)}_{3(3)})_{3D-KP_{15}}$ and $(P^{1(1)}_{3(3)})_{3D-KP_{15}}$ give the same nondominated vector $e_{3(5)}$. However, since our aim is to find $e_{3(4)}$ at that step, we do not keep these eliminated candidate nondominated vectors which will be found again in future iterations.

If the new nondominated solution is inserted to $(m+1)^{st}$ position, then the candidate solutions corresponding to the solutions in j^{th} solution satisfying $j \geq m+1$ will not change in the following iteration.

Proposition 4. If $J_{e_i(n+1)}^{S_i^r(n+1)} = m+1$ and $m+1 \le j$, then $ce_{i(n+2)}^{r(j+1)} = ce_{i(n+1)}^{r(j)}$.

Proof. Without loss of generality, let us take $r=1$ and $i=3$. As stated in previous definitions, the optimal solution of problem $(P_{3(n)}^{1(j)})$ gives the nondominated vector $ce_{3(n+1)}^{1(j)} = \left(cz_{1[3(n+1)]}^{1(j)}, cz_{2[3(n+1)]}^{1(j)}, cz_{3[3(n+1)]}^{1(j)}\right)$.

$$
\begin{aligned}
&(P_{3(n)}^{1(j)}) \\
&Max\ z_3 + \varepsilon z_1 + \varepsilon z_2 \\
&subject\ to \\
&z_1 \ge lb_{z1}(P_{3(n)}^{1(j)}) \\
&z_2 \ge lb_{z2}(P_{3(n)}^{1(j)}) \\
&x \in X
\end{aligned}
$$

where

$$
\begin{aligned}
&j = 0,1,\ldots,n \\
&lb_{z1}(P_{3(n)}^{1(j)}) = \begin{cases} -M, & if\ \ j=0 \\ z_{1[3(n)]}^{1(j)} + 1, & \text{otherwise} \end{cases}
\end{aligned}
$$

$$
lb_{z2}(P_{3(n)}^{1(j)}) = \begin{cases} -M, & if\ \ j=n \\ \max\limits_{n \ge h > j} \left\{ z_{2[3(n)]}^{1(h)} \right\} + 1, & \text{otherwise} \end{cases}
$$

Since $J_{e_1(n+1)}^{S_1^3(n+1)} = m+1$, we know the new solution $e_{3(n+1)}$ will be inserted in the $(m+1)^{st}$ position of $S_3^1(n)$. Therefore, all solutions in position j of the list $S_3^1(n)$ such that $m+1 \le j$ will change their positions such that they will take place in the $(j+1)^{st}$ position of the list $S_3^1(n+1)$. Then we can write $ce_{3(n+1)}^{1(j+1)} = ce_{3(n)}^{1(j)}$ which implies $cz_{1[3(n+1)]}^{1(j+1)} = cz_{1[3(n)]}^{1(j)}$, $cz_{2[3(n+1)]}^{1(j+1)} = cz_{2[3(n)]}^{1(j)}$. Furthermore, we obtain $lb_{z1}(P_{3(n+1)}^{1(j+1)}) = lb_{z1}(P_{3(n)}^{1(j)})$ and $lb_{z2}(P_{3(n+1)}^{1(j+1)}) = lb_{z2}(P_{3(n)}^{1(j)})$ which implies $(P_{3(n+1)}^{1(j+1)})$ is equivalent to model $(P^{1(j)}_{3(n)})$. Then we can conclude that the optimal solutions corresponding to these problems will be the same such that $ce^{r(j+1)}_{i(n+2)} = ce^{r(j)}_{i(n+1)}$. □

The candidate solutions corresponding to the solutions in j^{th} solution, such that $j \le m$, will also not change if z_a value of the new nondominated solution is larger than the corresponding lower bounds for z_a values.

Proposition 5. If $J^{S^r_i(n+1)}_{e_i(n+1)} = m+1$ and $z_{a[r(n+1)]} \le \max_{n \ge h > j} \left\{ z^{i(h)}_{a[r(n)]} \right\} \quad a \ne r, a \ne i$

$m \ge j$, then $ce^{r(j)}_{i(n+2)} = ce^{r(j)}_{i(n+1)}$.

Proof. Without loss of generality, let us take $r = 1$, $a = 2$ and $i = 3$. Since we know the new solution $e_{3(n+1)}$ will be inserted in the $(m+1)^{st}$ position of $S^1_3(n)$, then all solutions in position j of the list $S^1_3(n)$ such that $m+1 \le j$ will not change their positions and we will have $e^{1(j)}_{3(n+1)} = e^{1(j)}_{3(n)}$. Then we can write $z^{1(j)}_{1[3(n+1)]} = z^{1(j)}_{1[3(n)]}$, $z^{1(j)}_{2[3(n+1)]} = z^{1(j)}_{2[3(n)]}$ which implies $lb_{z1}(P^{1(j)}_{3(n+1)}) = lb_{z1}(P^{1(j)}_{3(n)})$. Furthermore, $z^{1(j+1)}_{2[3(n+1)]} = z^{1(j)}_{2[3(n)]}$ for $j \ge m+1$ means

$\max_{n+1 \ge h > j} \left\{ z^{1(h)}_{2[3(n+1)]} \right\} = \max\left(\max_{n \ge h > j} \left\{ z^{1(h)}_{2[3(n)]} \right\}, z_{2[3(n+1)]} \right)$ for $j \le m$. Since we know $z_{2[3(n+1)]} \le \max_{n \ge h > j} \left\{ z^{1(h)}_{2[3(n)]} \right\}$ for $j \le m$, then $\max_{n+1 \ge h > j} \left\{ z^{1(h)}_{2[3(n+1)]} \right\} = \max_{n \ge h > j} \left\{ z^{1(h)}_{2[3(n)]} \right\}$. Then we can write $lb_{z2}(P^{1(j)}_{3(n+1)}) = lb_{z2}(P^{1(j)}_{3(n)})$. Because both $lb_{z1}(P^{1(j)}_{3(n+1)}) = lb_{z1}(P^{1(j)}_{3(n)})$ and $lb_{z2}(P^{1(j)}_{3(n+1)}) = lb_{z2}(P^{1(j)}_{3(n)})$, we know $(P^{1(j+1)}_{3(n+1)})$ is equivalent to model $(P^{1(j)}_{3(n)})$. Then, we can conclude that the optimal solutions corresponding to these problems will be the same such that $ce^{r(j)}_{i(n+2)} = ce^{r(j)}_{i(n+1)}$. □

Specific to our example problem, since the new nondominated vector $e_{3(4)}$ will be placed at the end of the list $S_3^1(4)$ and it has a larger z_2 value than the previous lower bounds for z_2, we should update the candidate nondominated vectors.

We should also note that we can detect if the optimal solution will be identical to any of the previous ones by keeping the lower bounds and corresponding solutions as discussed in Proposition 6.

Proposition 6. If problem $(P_{i(n_1)}^{r(j_1)})$ with the lower bounds $lb_{z_r}(P_{i(n_1)}^{r(j_1)})$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)})$ gives the nondominated vector $ce_{i(n_1+1)}^{r(j_1)} = \left(cz_{1[i(n_1+1)]}^{r(j_1)}, cz_{2[i(n_1+1)]}^{r(j_1)}, cz_{3[i(n_1+1)]}^{r(j_1)}\right)$ and problem $(P_{i(n_2)}^{r(j_2)})$ has the lower bounds $lb_{z_r}(P_{i(n_2)}^{r(j_2)})$ and $lb_{z_a}(P_{i(n_2)}^{r(j_2)})$ such that $lb_{z_r}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_r}(P_{i(n_2)}^{r(j_2)}) \le cz_{r[i(n_1+1)]}^{r(j_1)}$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_a}(P_{i(n_2)}^{r(j_2)}) \le cz_{a[i(n_1+1)]}^{r(j_1)}$, where $r \ne a \ne i$, then $ce_{i(n_1+1)}^{r(j_1)}$ will be also an optimal solution for the problem $(P_{i(n_2)}^{r(j_2)})$.

Proof. Since $lb_{z_r}(P_{i(n_2)}^{r(j_2)}) \le cz_{r[i(n_1+1)]}^{r(j_1)}$ and $lb_{z_a}(P_{i(n_2)}^{r(j_2)}) \le cz_{a[i(n_1+1)]}^{r(j_1)}$, then the nondominated vector $ce_{i(n_1+1)}^{r(j_1)} = \left(cz_{1[i(n_1+1)]}^{r(j_1)}, cz_{2[i(n_1+1)]}^{r(j_1)}, cz_{3[i(n_1+1)]}^{r(j_1)}\right)$ is also feasible for the problem $(P_{i(n_2)}^{r(j_2)})$. To get contradiction, let us assume that $ce_{i(n_1+1)}^{r(j_1)}$ is not an optimal solution for problem $(P_{i(n_2)}^{r(j_2)})$. Then assume that problem $(P_{i(n_2)}^{r(j_2)})$ has an optimal solution $ce_{i(n_2+1)}^{r(j_2)} \ne ce_{i(n_1+1)}^{r(j_1)}$ and $ce_{i(n_2+1)}^{r(j_2)} = \left(cz_{1[i(n_2+1)]}^{r(j_2)}, cz_{2[i(n_2+1)]}^{r(j_2)}, cz_{3[i(n_2+1)]}^{r(j_2)}\right)$. Since $ce_{i(n_2+1)}^{r(j_1)}$ is not an optimal solution for problem $(P_{i(n_2)}^{r(j_2)})$ and both problems try to maximize z_i as much as possible, then $cz_{i[i(n_2+1)]}^{r(j_2)} > cz_{i[i(n_1+1)]}^{r(j_1)}$. Furthermore, we can write $lb_{z_r}(P_{i(n_2)}^{r(j_2)}) \le z_{r[i(n_2+1)]}^{r(j_2)}$ and $lb_{z_a}(P_{i(n_2)}^{r(j_2)}) \le z_{a[i(n_2+1)]}^{r(j_2)}$ in order to provide the feasibility. Since we also know $lb_{z_r}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_r}(P_{i(n_2)}^{r(j_2)})$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_a}(P_{i(n_2)}^{r(j_2)})$, we obtain $lb_{z_r}(P_{i(n_1)}^{r(j_1)}) \le cz_{r[i(n_2+1)]}^{r(j_2)}$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)}) \le cz_{a[i(n_2+1)]}^{r(j_2)}$ which implies $ce_{i(n_2+1)}^{r(j_2)}$ is also a feasible solution for problem $(P_{i(n_1)}^{r(j_1)})$. However, since we obtain $cz_{i[i(n_2+1)]}^{r(j_2)} > cz_{i[i(n_1+1)]}^{r(j_1)}$, implying that $ce_{i(n_2+1)}^{r(j_2)}$ has a better objective function value, then $ce_{i(n_1+1)}^{r(j_1)} = \left(cz_{1[i(n_1+1)]}^{r(j_1)}, cz_{2[i(n_1+1)]}^{r(j_1)}, cz_{3[i(n_1+1)]}^{r(j_1)}\right)$ will not be an optimal solution to the problem $(P_{i(n_1)}^{r(j_1)})$. We obtain a contradiction so we can conclude that $ce_{i(n_1+1)}^{r(j_1)}$ will be also an optimal solution for problem $(P_{i(n_2)}^{r(j_2)})$. □

We can also detect whether the problem is feasible or not by storing the lower bounds that created infeasibility in previous iterations.

Corollary 4. If problem $(P_{i(n_1)}^{r(j_1)})$ with lower bounds $lb_{z_r}(P_{i(n_1)}^{r(j_1)})$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)})$ is infeasible and problem $(P_{i(n_2)}^{r(j_2)})$ has lower bounds $lb_{z_r}(P_{i(n_2)}^{r(j_2)})$ and $lb_{z_a}(P_{i(n_2)}^{r(j_2)})$ such that $lb_{z_r}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_r}(P_{i(n_2)}^{r(j_2)})$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_a}(P_{i(n_2)}^{r(j_2)})$, where $r \ne a \ne i$, then problem $(P_{i(n_2)}^{r(j_2)})$ will also be infeasible. □

Proof. In order to get contradiction, we assume that the problem $(P_{i(n_2)}^{r(j_2)})$ is feasible and it has the optimal solution, $ce_{i(n_2+1)}^{r(j_2)}$. Then, $ce_{i(n_2+1)}^{r(j_2)}$ will also be a feasible solution to problem $(P_{i(n_1)}^{r(j_1)})$ which contradicts the fact that problem $(P_{i(n_1)}^{r(j_1)})$ is infeasible. □

Corollary 5. If problem $(P_{i(n_1)}^{r(j_1)})$ is infeasible and $e_{i(n_1)}^{r(j_1)} = e_{i(n_2)}^{r(j_2)}$, then problems $(P_{i(n_2)}^{r(j_2)})$ will also be infeasible for $n_2 \ge n_1$.

Proof. Problem $(P_{i(n_1)}^{r(j_1)})$ has the lower bounds:

$$lb_{z_r}(P_{i(n_1)}^{r(j_1)}) = \begin{cases} -M, & \text{if } j_1 = 0 \\ z_{r[i(n_1)]}^{r(j_1)} + 1, & \text{otherwise} \end{cases}$$

$$lb_{z_a}(P_{i(n_1)}^{r(j_1)}) = \begin{cases} -M, & \text{if } j_1 = n_1 \\ \max\limits_{n \ge h > j_1} \left\{ z_{a[i(n_1)]}^{r(h)} \right\} + 1, & \text{otherwise} \end{cases}$$

where $e_{i(n_1)}^{r(j_1)} = \left(z_{1[i(n_1)]}^{r(j_1)}, z_{2[i(n_1)]}^{r(j_1)}, z_{3[i(n_1)]}^{r(j_1)} \right)$ and $a \ne i \ne r$.

Since $e_{i(n_1)}^{r(j_1)} = e_{i(n_2)}^{r(j_2)}$, then we can write $lb_{z_r}(P_{i(n_1)}^{r(j_1)}) = lb_{z_r}(P_{i(j_2)}^{r(j_2)})$ and $j_1 \le j_2$ according to the discussions in Propositions 4 and 5. Since both $j_1 \le j_2$ and $n_1 \le n_2$, we obtain $\max\limits_{n1 \ge h > j_1} \left\{ z_{a[i(n_1)]}^{r(h)} \right\} \le \max\limits_{n2 \ge h > j_2} \left\{ z_{a[i(n_2)]}^{r(h)} \right\}$ which implies $lb_{z_a}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_a}(P_{i(n_2)}^{r(j_2)})$. Since problem $(P_{i(n_1)}^{r(j_1)})$ with lower bounds $lb_{z_r}(P_{i(n_1)}^{r(j_1)})$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)})$ is infeasible, then problem $(P_{i(n2)}^{r(j2)})$ will also be infeasible since $lb_{z_r}(P_{i(n_1)}^{r(j_1)}) = lb_{z_r}(P_{i(n_2)}^{r(j_2)})$ and $lb_{z_a}(P_{i(n_1)}^{r(j_1)}) \le lb_{z_a}(P_{i(n_2)}^{r(j_2)})$ according to the Corollary 5. □

If we look at our example, model $(P_{3(3)}^{1(1)})_{3D-KP_{15}}$ with lower bounds $lb_{z_1}\left[P_{3(3)}^{1(1)} \right] = 319$ and $lb_{z_2}\left[P_{3(3)}^{1(1)} \right] = 470$ gives the nondominated vector $ce_{3(4)}^{1(1)} = (376, 536, 476)$. Then, we store these lower bounds and corresponding nondominated vectors in an archive because problems $(P_{3(n)}^{1(j)})_{3D-KP_{15}}$ with lower bounds $319 \le lb_{z_1}(P_{3(3)}^{1(1)}) \le 376$ and $470 \le lb_{z_2}(P_{3(n)}^{1(j)}) \le 536$ will give the same nondominated vector $(376, 536, 476)$. In case of infeasibility, we also keep the lower bounds in a different archive since if our problem is infeasible with lower bounds $lb_{z_1}(P_{3(3)}^{1(1)}) = 319$ and $lb_{z_2}(P_{3(3)}^{1(1)}) = 470$, then problems $(P_{3(n)}^{1(j)})_{3D-KP_{15}}$ with lower bounds $319 \le lb_{z_1}(P_{3(3)}^{1(1)})$ and $470 \le lb_{z_2}(P_{3(n)}^{1(j)})$ will also be infeasible. Furthermore, if the candidate nondominated vector $ce_{3(4)}^{1(1)}$ is infeasible, we keep this information by marking it infeasible. The corresponding candidate vector in the next iteration will also be infeasible since the lower bounds will be at least $(319, 470)$.

Remark. We may not need to solve many of $\frac{N(N+1)}{2}$ models by storing information. We can detect some solutions that will be identical with previous solutions and may not need to solve many of the models.

Algorithm 2 for the three criteria case

Step 0. Initialization

Let $W = \varnothing$, $I = \varnothing$, $C = \varnothing$, $IC = \varnothing$ and $n = 0$,

where

W : The nondominated solutions obtained.

$I = \left\{ \left[lb_{z_r}^f, lb_{z_a}^f \right] \right\}$: The set of lower bound pairs resulting in infeasibility.

$\mathrm{C} = \left\{ \left[lb_{z_r}^c, lb_{z_a}^c, e_c \right] : e_c = \left(z_1^c, z_2^c, z_3^c \right) \right\}$: The set of candidate efficient solutions and corresponding lower bound pairs.

$IC = \left\{ j : (P_{i(n)}^{r(j)}) \text{ is infeasible, } 0 \le j \le n \right\}$: The set of j values for which $P_{i(n)}^{r(j)}$ is infeasible.

n : The current number of nondominated solutions.

$$SOLVE_{i(n)}^{r(j)} = \begin{cases} 1, & \textit{if } P_{i(n)}^{r(j)} \text{ is to be solved to update its candidate solution} \\ 0, & \textit{otherwise} \end{cases}$$

Step 1. Solve model (P_i) .

(P_i)
$Max\ z_i + \varepsilon z_r + \varepsilon z_a$
subject to
$x \in X$

If the problem is feasible, denote its optimal objective vector as $e_{i(1)} = \left(z^{r(1)}_{1[i(1)]}, z^{r(1)}_{2[i(1)]}, z^{r(1)}_{3[i(1)]} \right)$ and update:

$W = e_{i(1)} \cup W$ where $S^r_i(1) = W$ and $e^{r(1)}_{i(1)} = e_{i(1)}$.

$lb_{z_r}(P^{r(0)}_{i(1)}) = -M$, $lb_{z_a}(P^{r(0)}_{a(1)}) = z^{r(1)}_{a[i(1)]} + 1$

$lb_{z_r}(P^{r(1)}_{i(1)}) = z^{r(1)}_{r[i(1)]} + 1$, $lb_{z_a}(P^{r(1)}_{i(1)}) = -M$.

$SOLVE^{r(0)}_{i(1)} = SOLVE^{r(1)}_{i(1)} = 1$. Go to Step.2.

If the problem is infeasible, stop. The problem does not have any feasible solution.

Step 2. $n = n + 1$.

Step 2.0. $j = 0$, $\max = \text{-}M$

Step 2.1. If $j > n$, go to Step 2.11. Otherwise, go to Step 2.2.

Step 2.2. If $SOLVE^j_n = 0$, then go to Step 2.8. Otherwise, go to Step 2.3.

Step 2.3. If $j \in IC$, then go to Step 2.10. Otherwise, go to Step 2.4.

Step 2.4. If $\left[lb^f_{z_r}, lb^f_{z_a} \right] \in I$ such that $lb_{z_r}(P^{r(j)}_{i(n)}) \geq lb^f_{z_r}$ and $lb_{z_a}(P^{r(j)}_{i(n)}) \geq lb^f_{z_a}$ then go to Step 2.9. Otherwise, go to Step 2.5.

Step 2.5. If $\left[lb^{c}_{z_r}, lb^{c}_{z_a}, e_c\right] \in C$ such that $z^{c}_{r} \geq lb_{z_r}(P^{r(j)}_{i(n)}) \geq lb^{c}_{z_r}$ and $z^{c}_{a} \geq lb_{z_a}(P^{r(j)}_{i(n)}) \geq lb^{c}_{z_a}$ then $ce^{j}_{i(n+1)} = e_c$. Go to Step 2.8. Otherwise, go to Step 2.6.

Step 2.6. Solve the problem $P^{r(j)}_{i(n)}$. If the problem is feasible, denote its optimal objective vector as $ce^{j}_{i(n+1)}$ and go to Step 2.7. Otherwise, go to Step 2.9.

Step 2.7. Update:

$$\mathrm{C} \leftarrow \mathrm{C} \cup \left[lb_{z_r}(P^{r(j)}_{i(n)}), lb_{z_a}(P^{r(j)}_{i(n)}), ce^{j}_{i(n+1)}\right]$$

Step 2.8. If $cz^{j}_{i[i(n+1)]} \geq \max$, then update $\max = cz^{j}_{i[i(n+1)]}$, $m = j$. Go to Step 2.10.

Step 2.9. Update $\mathrm{I} \leftarrow \mathrm{I} \cup \left[lb_{z_r}(P^{r(j)}_{i(n)}), lb_{z_a}(P^{r(j)}_{i(n)})\right]$ and $IC \leftarrow \{j\} \cup IC$.

Step 2.10. $j \leftarrow j+1$. Repeat Step 2.1.

Step 2.11. If $\max = \text{-}M$, go to Step 3. Otherwise, go to Step 2.12.

Step 2.12. $e_i(n+1) = ce^{r(m)}_{i(n+1)}$ and insert $e_i(n+1)$ in position $J^{S^r_i(n+1)}_{e_i(n+1)} = m+1$ of the list $S^r_i(n+1)$ by changing the positions $J^{S^r_i(n_j)}_{e_i(n_j)} = J^{S^r_i(n_j)}_{e_i(n_j)} + 1 \quad for \quad J^{S^r_i(n_j)}_{e_i(n_j)} \geq m+1, \quad 0 < n_j \leq n$.

Before insertion, update:

Initialize $SOLVE^{h}_{n+1} = 1,\ 0 \leq \mathrm{h} \leq n+1$.

$ce^{r(h+1)}_{i(n+2)} = ce^{r(h)}_{i(n+1)}$ and change $SOLVE^{h+1}_{n+1} = 0$ if $n+1 \geq h \geq m+1$.

$ce^{r(h)}_{i(n+2)} = ce^{r(h)}_{i(n+1)}$ and $SOLVE^{h}_{n+1} = 0$ if $0 \leq h < m+1$ and $z_{a[r(n+1)]} + 1 \leq lb_{z_a}(P^{r(h)}_{i(n)}) \quad a \neq r, a \neq i$

$$lb_{z_r}(P_{i(n+1)}^{r(h)}) = z_{r[i(n+1)]}^{r(h)} + 1 \ , \ 0 \le \mathrm{h} \le n+1$$

$$lb_{z_a}(P_{i(n+1)}^{r(h)}) = z_{a[r(n+1)]} + 1 \text{ for all } h < m+1 \text{ if } z_{a[r(n+1)]} + 1 > lb_{z_a}(P_{i(n)}^{r(h)}) \quad a \ne r, a \ne i$$

$$lb_{z_r}(P_{i(n+1)}^{r(m+1)}) = z_{r[i(n+1)]}^{r(m+1)} + 1$$

$$lb_{z_a}(P_{i(n+1)}^{r(m+1)}) = lb_{z_a}(P_{i(n)}^{r(m)})$$

$$SOLVE_{n+1}^{m+1} = 1$$

$IC \leftarrow IC \cup \{h+1\} - \{h\}$ for $h \in IC$ and $h \ge m+1$

W = $e_{i(n+1)} \cup W$ and then repeat Step 2.

Step 3. Stop. $W = S_i^r(n)$ is the entire set of nondominated vectors for problem (P) and $n = |W|$.

3.3.2 Generalization of Algorithm 2

We can generalize Algorithm 2 for problems with more than three objectives. Similar to the three criteria case, we employ a sorted list, $S_i^{r_1}(n)$ where the solutions in $S_i(n)$ are in the nondecreasing order of objective r_1. However, we also define a new set of solutions, $S_i^{r_1[j_1]}(n) = \left\{ e_{i(n)}^{r(j)} : e_{i(n)}^{r(j)} \in S_i^{r_1}(n),\ j_1 < j \right\}$, which includes only the nondominated solutions with the index greater than j_1 in $S_i^{r_1}(n)$. Furthermore, we use a second list $S_i^{r_1[j_1],r_2}(n)$, where the solutions in $S_i^{r_1[j_1]}(n)$ are sorted in the nondecreasing order of objective r_2. We denote the nondominated solution in the $j_2{}^{th}$ position of $S_i^{r_1[j_1],r_2}(n)$ as $e_{i(n)}^{r_1[j_1],r_2[j_2]}$. For each different j_1 and j_2 values, we determine the lower bounds corresponding to models $P_{i(n)}^{r_1[j_1],r_2[j_2]}$ as described for the example on Table 3.9 and in Figure 3.2. If we have q objectives, then we solve $P_{i(n)}^{r_1[j_1], r_2[j_2], \ldots, r_{q-2}[j_{q-2}]}$ for each different value of $j_k \, (k = 1, \ldots, n-2)$.

Let us demonstrate the algorithm on a knapsack problem with 10 items and four objectives, $4D - KP_{10}$, which has 14 nondominated solutions as seen in Table 3.6.

Table 3.6 Nondominated vectors corresponding to the $4D - KP_{10}$

	z_1	z_2	z_3	z_4
1	326	344	218	359
2	263	366	229	349
3	304	356	168	338
4	259	366	280	336
5	389	301	194	325
6	237	378	230	315
8	366	319	264	288
7	263	312	297	288
9	382	295	250	281
10	317	306	274	272
11	299	341	326	265
12	315	317	312	258
13	277	353	276	244
14	319	263	329	210

Without loss of generality, let us take $i = 4$, $r_1 = 1$ and $r_2 = 2$. Assume we have 3 nondominated solutions having the largest z_4 values among all the nondominated vectors as demonstrated in Table 3.7 and Table 3.8.

Table 3.7 The nondominated vectors of $4D - KP_{10}$ in the set $S_4(3)$

$S_4(3)$	z_1	z_2	z_3	z_4
$e_{4(1)}$	326	344	218	359
$e_{4(2)}$	263	366	229	349
$e_{4(3)}$	304	356	168	338

Table 3.8 The nondominated vectors of $4D - KP_{10}$ in the set $S_4^1(3)$

$S_4^1(3)$	z_1	z_2	z_3	z_4
$e_{4(3)}^{1(1)}$	263	366	229	349
$e_{4(3)}^{1(2)}$	304	356	168	338
$e_{4(3)}^{1(3)}$	326	344	218	359

By using these nondominated vectors, we solve the following models to find the nondominated vector $e_{4(4)}$.

$$
\begin{aligned}
&(P_{4(3)}^{1[j_1],2[j_2]})_{4D-KP_{10}}\\
&Max\ z_4 + \varepsilon z_1 + \varepsilon z_2 + \varepsilon z_2\\
&subject\ to\\
&z_1 \geq lb_{z_1}(P_{4(3)}^{1[j_1],2[j_2]})_{4D-KP_{10}}\\
&z_2 \geq lb_{z_2}(P_{4(3)}^{1[j_1],2[j_2]})_{4D-KP_{10}}\\
&z_3 \geq lb_{z_3}(P_{4(3)}^{1[j_1],2[j_2]})_{4D-KP_{10}}\\
&x \in X
\end{aligned}
$$

where the lower bounds are given in Table 3.9.

Table 3.9 Lower Bounds for $(P_{4(3)}^{1[j_1],2[j_2]})_{4D-KP_{10}}$

j_1	j_2	lb_{z_1}	lb_{z_2}	lb_{z_3}
$j_1=0$	$j_2=0$	$-M$	$-M$	$\max(229,168,218)+1$
	$j_2=1$	$-M$	344+1	$\max(168,229)+1$
	$j_2=2$	$-M$	356+1	$\max(229)+1$
	$j_2=3$	$-M$	366+1	$-M$
$j_1=1$	$j_2=0$	263+1	$-M$	$\max(168,218)+1$
	$j_2=1$	263+1	344+1	$\max(168)+1$
	$j_2=2$	263+1	356+1	$-M$
$j_1=2$	$j_2=0$	304+1	$-M$	$\max(218)+1$
	$j_2=1$	304+1	344+1	$-M$
$j_1=3$	$j_2=0$	326+1	$-M$	$-M$

These lower bounds are determined according to the sorting mechanism described for $j_1=0$ and $j_2=1$ in Figure 3.7.

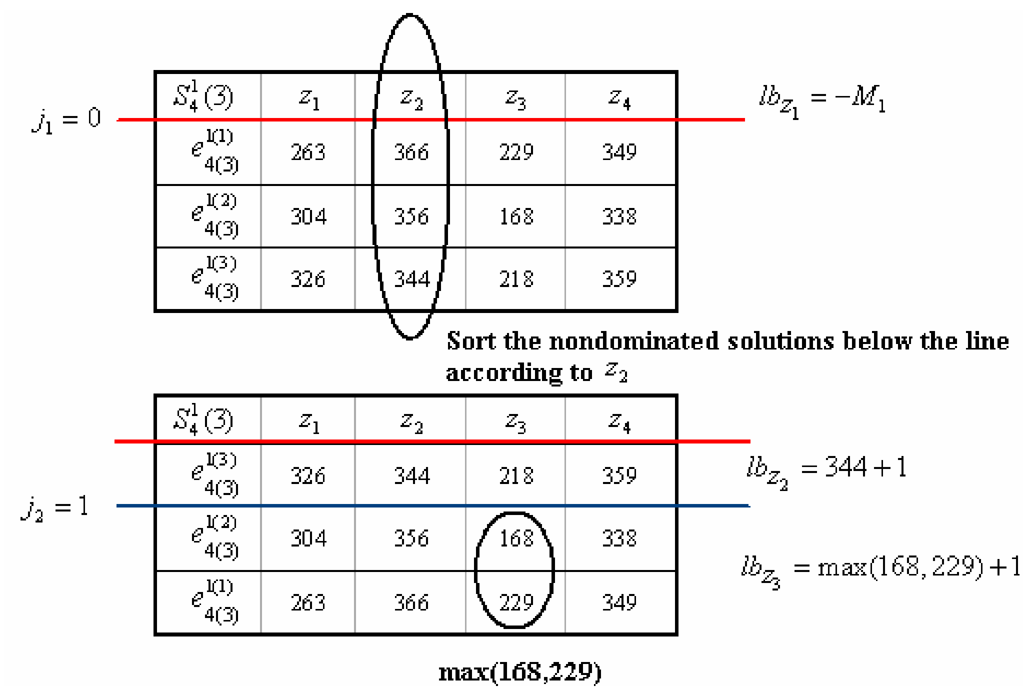


Figure 3.7 Determination of the lower bounds for problem $(P_{4(3)}^{1[0],2[1]})_{4D-KP_{10}}$

Table 3.10 Candidate solutions corresponding to problem $(P_{4(3)}^{1[j_1],2[j_2]})_{4D-KP_{10}}$

j_1	j_2	Corresponding Candidate Solution
$j_1 = 0$	$j_2 = 0$	$ce_{4(4)}^{1[0],2[0]} = (259,366,280,336)$
	$j_2 = 1$	$ce_{4(4)}^{1[0],2[1]} = (259,366,280,336)$
	$j_2 = 2$	$ce_{4(4)}^{1[0],2[2]} = (259,366,280,336)$
	$j_2 = 3$	$ce_{4(4)}^{1[0],2[3]} = (237,378,230,315)$
$j_1 = 1$	$j_2 = 0$	$ce_{4(4)}^{1[1],2[0]} = (389,301,194,325)$
	$j_2 = 1$	$ce_{4(4)}^{1[1],2[1]} = (277,353,276,244)$
	$j_2 = 2$	$ce_{4(4)}^{1[1],2[2]}$ infeasible
$j_1 = 2$	$j_2 = 0$	$ce_{4(4)}^{1[2],2[0]} = (366,319,264,288)$
	$j_2 = 1$	$ce_{4(4)}^{1[2],2[1]}$ infeasible
$j_1 = 3$	$j_2 = 0$	$ce_{4(4)}^{1[3],2[0]} = (389,301,194,325)$

Since our aim is to maximize z_4 as much as possible, we select the candidate with the largest z_4 value. Since all $ce_{4(4)}^{1[0],2[0]}, ce_{4(4)}^{1[0],2[1]}, ce_{4(4)}^{1[0],2[2]}$ have the same largest value, we select the last one such that $e_{4(4)} = ce_{4(4)}^{1[0],2[2]} = (259,366,280,336)$. Since $j_1 = 0$ for $ce_{4(4)}^{1[0],2[2]}$, we insert $e_{4(4)}$ in the first position of the list $S_4^1(3)$. Furthermore, since it is inserted at the beginning, the lower bounds $j_1 > 0$ will not be changed which implies they will give the same nondominated solutions. Then, since their position in the list is changed, we can write $ce_{4(5)}^{1[j_1+1],2[j_2]} = ce_{4(4)}^{1[j_1],2[j_2]}$ for $j_1 > 0$. In case of infeasibility of the model $(P_{4(3)}^{1[j_1],2[j_2]})_{4D-KP_{10}}$, we conclude that $(P_{4(4)}^{j_1+1,j_2})_{4D-KP_{10}}$ is also infeasible for the problems $j_1 > 0$ according to the propositions discussed. For instance, in our problem $(P_{4(3)}^{1[1],2[2]})_{4D-KP_{10}}$ is infeasible as seen in Table 3.10, then we can write $(P^{1[2],2[2]}_{4(3)})_{4D-KP_{10}}$ will also be infeasible without solving the model.

Another observation is the fact that we do not need to solve all the problems corresponding to each (j_1, j_2) pair since there may be equivalent problems giving the same nondominated vector. For instance, since $lb_{z_1}(P^{1[0],2[0]}_{4(3)})_{4D-KP_{10}} \le lb_{z_1}(P^{1[0],2[1]}_{4(3)})_{4D-KP_{10}} \le z^{1[0],2[0]}_{1[4(4)]}$,

$lb_{z_2}(P^{0,0}_{4(3)})_{4D-KP_{10}} \le lb_{z_2}(P^{0,1}_{4(3)})_{4D-KP_{10}} \le z^{0,0}_{2[4(4)]}$ and

$lb_{z_3}(P^{0,0}_{4(3)})_{4D-KP_{10}} \le lb_{z_3}(P^{0,1}_{4(3)})_{4D-KP_{10}} \le z^{0,0}_{3[4(4)]}$, then we obtain $ce^{1[0],2[1]}_{4(4)} = ce^{1[0],2[0]}_{4(4)} = (259,366,280,336)$ without solving the model again.

As seen in Table 3.10, we have

$\sum_{j_1=0}^{n}\sum_{j_2=0}^{n-j_1} 1 = \sum_{j_1=0}^{n}(n+1-j_1) = (n+1)+(n)+(n-1)+...+2+1 = \frac{(n+1)(n+2)}{2}$ models to find $(n+1)^{th}$ solution by using n solutions we know. Therefore, if the number of nondominated vectors is equal to N, then we have $1+\sum_{n=1}^{N-1}\frac{(n+1)(n+2)}{2}$ problems to be solved in the worst case $(O(N^3))$. However, since there should be models giving the same nondominated solutions as discussed before, we may not need to solve many of them. By transferring information to the next iterations, we can determine the candidate solutions without solving the model as shown in our example. When we have q objectives, the number of models to be solved to find $(n+1)^{st}$ solution by using n solutions in the worst case will be equal to $\sum_{j_1=0}^{n}\sum_{j_2=0}^{n-j_1}\sum_{j_3=0}^{n-j_1-j_2}\cdots\sum_{j_{q-2}=0}^{n-j_1-j_2-...-j_{q-3}} 1$. If we have N nondominated solutions, then the number of models to be solved will be

$1+\sum_{n=1}^{N-1}\sum_{j_1=0}^{n}\sum_{j_2=0}^{n-j_1}\sum_{j_3=0}^{n-j_1-j_2}\cdots\sum_{j_{q-2}=0}^{n-j_1-j_2-...-j_{q-3}} 1$ in worst case $(O(N^{q-1}))$.

3.4 Computational Experiments

We compare our two exact algorithms with the algorithm developed by Sylva and Crema (2004) on Multiobjective Knapsack Problem (MOKP). We generate the weights and profits of the items, as integers uniformly distributed between 10 and 100. We take the capacity of the knapsacks as half of total weight. In addition, we take $\varepsilon = 0.001$. As the number of nondominated solutions increase, the complexity of the algorithm proposed by Sylva and Crema increase considerably as seen in the computational times indicated in Table 3.11. Therefore, we have worked on small-sized, 10 and 15 items, knapsack problems with three objectives. ($q = 3$).

$$
\begin{aligned}
&MOKP \\
&\text{"}Max\text{"}\left\{z_1(x), z_2(x), ..., z_q(x)\right\} \\
&subject\ to \\
&\sum_{j=1}^{m} w_{ij} x_j \le C_i \quad i = 1, 2, ..., q \\
&x_j \in \left\{0, 1\right\}
\end{aligned}
$$

where

$$z_i(x) = \sum_{j=1}^{m} p_{ij} x_j$$

p_{ij} : *profit of item j for knapsack i*

w_{ij} : *weight of item j for knapsack i*

C_i : *capacity of knapsack i*

$$x_j = \begin{cases} 1 & \text{if item } j \text{ is selected to put in knapsacks} \\ 0 & \text{otherwise} \end{cases}$$

$$C_i = \frac{\sum_{j=1}^{m} w_{ij} x_j}{2}$$

q : *the number of knapsacks*

m : *the number of items*

Table 3.11 Comparison of Algorithms on MOKP with $q = 3$

Number of items	Problem	Number of nondominated vectors (N)	Solution Time (CPU Time in seconds)		
			Sylva and Crema	Algorithm-1	Algorithm-2
15	1	12	1.06	0.41	0.50
	2	12	1.06	0.39	0.47
	3	23	3.58	1.36	0.98
	4	27	9.78	2.08	1.34
	5	29	12.94	1.69	1.50
25	1	38	188.69	11.14	3.36
	2	49	645.76	43.22	5.28
	3	54	428.73	24.39	5.30
	4	81	38503.59	61.8	8.80
	5	131	55708.38	224.69	16.78

The three algorithms in Table 3.11 are all exact algorithms generating all nondominated solutions. Therefore, we employ corresponding solution times as a performance measure. Algorithm 1 outperforms the algorithm developed by Sylva and Crema as seen in the computational times. This is expected since we decrease the number of binary variables and constraints iteratively inserted for each new nondominated solution. The computational times depend on N because we keep adding new binary variables and constraints until all nondominated solutions (N) are obtained which increases the computational complexity at each iteration. Table 3.11 also indicates that there is a significant increase in the difference in the computational times even when N is slightly increased.

Table 3.12. Comparison of Algorithm-1 and Algorithm-2 on MOKP with $q = 3$

Number of items	Problem	Number of nondominated vectors (N)	Solution Time (CPU Time in seconds)	
			Algorithm-1	Algorithm-2
25	1	76	50.16	7.41
	2	163	243.05	19.45
	3	168	996.42	32.03
	4	182	473.64	28.63
50	1	280	16919.64	184.23
	2	356	14064.81	217.77
	3	519	100670.52	312.17

Although, Algorithm 1 outperforms the algorithm of Sylva and Crema, the additional constraints and variables still grow and cause computational difficulty in Algorithm 1. On the other hand, Algorithm 2 involving a sorting and search mechanism performs better than Algorithm 1. The number of models we solve in Algorithm 2 is larger but each model has the same number of constraints and variables regardless of the solutions on hand. While the computational times of Algorithm 1 and Algorithm 2 for the knapsack problem with 15 items are not much different in Table 3.11, we observe that the relative performance of Algorithm 2 gets much better as the problem size increases as seen in Table 3.12.

We further tested the performance of Algorithm 2 on MOCO problems including the random instances of MOKP, MOST and MOSP problems with three and four objectives ($q = 3,\ q = 4$).

In order to have a mathematical program, we formulate the minimum spanning tree problem as a multicommodity flow problem. Then we can write MOST problem as follows:

$$
\begin{aligned}
& MOST \\
& \text{"}Max\text{"}\ \left\{ z_1(w_{ij}), z_2(w_{ij}), \ldots, z_q(w_{ij}) \right\} \\
& subject\ to \\
& \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} = 1 \\
& \sum_{j=1}^{n} f_{ij}^{k} - \sum_{j=1}^{n} f_{ji}^{k} = \begin{cases} 1 & i = 1 \\ -1 & i = k \\ 0 & otherwise \end{cases} \qquad i = 1,2,\ldots,n \quad k = 2,3,\ldots,n \\
& f_{ij}^{k} \le w_{ij} \quad i, j = 1,2,\ldots,n \quad k = 2,3,\ldots,n \\
& w_{ij} + w_{ji} = x_{ij} \quad \forall i, j \\
& x_{ij} \in \{0,1\}
\end{aligned}
$$

where

$$z_v(w_{ij}) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} \quad v = 1,2,...,q$$

c_{ijv} : unit cost of flow from node i to node j in v^{th} criterion

$$x_{ij} = \begin{cases} 1 & \textit{if} \text{ any flow exists from node } i \text{ to node } j \\ 0 & \textit{otherwise} \end{cases}$$

w_{ij} : total flow from node i to node j in units

f_{ij}^{k} : total flow of commodity k from node i to node j in units

If we have a complete graph with n nodes, we define node 1 as the supply node of n commodities and the remaining nodes as demand nodes where each demand node has a demand for a different commodity of exactly one unit. Therefore, the difference of outflow and the inflow of commodity k will be equal to 1 for the demand node k whereas it will be equal to -1 for the supply node 1. All other nodes will be transshipment nodes for this commodity k. This model will give us a spanning tree since using only one supplier will guarantee a connected graph. In addition, no cycles will occur in this connected graph to minimize the cost. In our experiments, we generate cost parameters as integers uniformly distributed between 10 and 100.

Our preliminary experiments for the MOSP problem showed that the number of nondominated solutions is small when we use a complete graph. Typically, there were several paths from source to sink with relatively small number of arcs and these dominate many other paths. In order to overcome this difficulty, we generate special random graphs instead of a complete graph. We define source and sink nodes as nodes 1 and n respectively as seen in Figure 3.8. Then we iteratively generate a random integer for the number of nodes per stage, $(n_1 - 1), (n_2 - n_1), ..., (n_s - n_{s-1})$ uniformly distributed between $\left[(n-2)*0.08, (n-2)*0.12\right]$ (i.e. between 8% to 12%, and on the average 10% of the number of nodes excluding the source and sink nodes.). We keep on generating as long as the number of nodes left satisfy, $n - 2 - n_{s-1} \leq 1.02(n-2)+1$. Then, we stop and calculate the number of nodes corresponding to the last stage, s, as the number of nodes left such that $n_s - n_{s-1}$ where $n_s = n-1$.

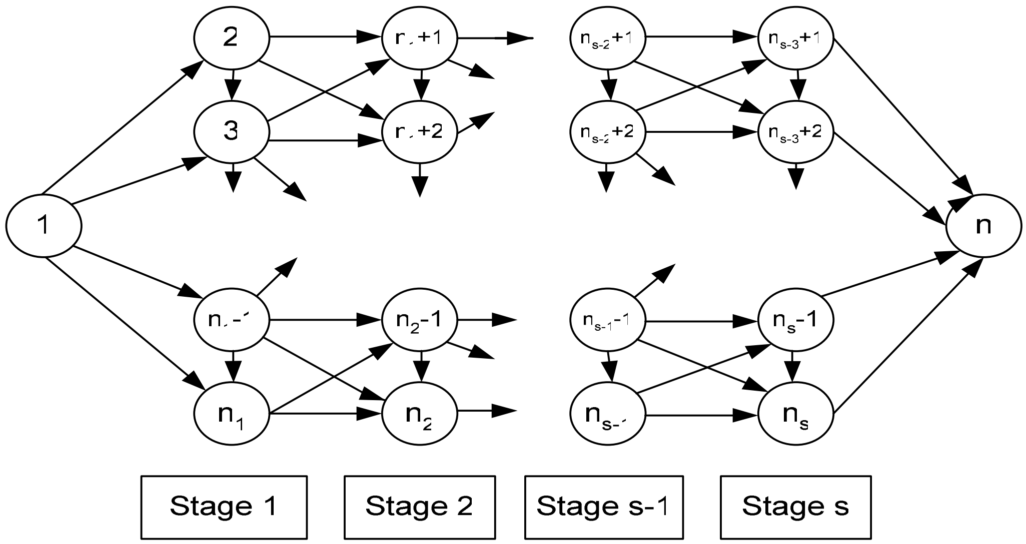


Figure 3.8 Generation of Random Graphs for Shortest Path Problems

After determining the number of nodes for each stage, we define the edges that will be included in our graph and generate corresponding integer costs, c_{ij}, from discrete uniform distribution as below:

$$c_{ij} = \begin{cases} UNIF(10,50), & i, j \in Stage_k \text{ and } i < j \quad k = 1,\dots,s \\ UNIF(30,100), & i \in Stage_k,\ j \in Stage_{k+1} \quad k = 1,\dots,s-1 \\ M, & otherwise \end{cases}$$

M is sufficiently large number to guarantee that the corresponding edge will not be included in the random graph. We allow flows to the adjacent nodes in the same stage or to nodes in the next stage. Then, we formulate MOSP as below:

MOSP

$$\text{"}Max\text{"}\left\{z_1(x_{ij}), z_2(x_{ij}), \ldots, z_q(x_{ij})\right\}$$

subject to

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ij} = \begin{cases} 1 & i = 1 \\ -1 & i = k \\ 0 & otherwise \end{cases} \qquad \forall i, k$$

$$x_{ij} \in \{0,1\}$$

where

$$z_v(x_{ij}) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ijv} x_{ij} \quad v = 1, 2, \ldots, q$$

c_{ijv} : unit cost of arc between node i and node j in v^{th} criterion

$$x_{ij} = \begin{cases} 1 & \textit{if} \text{ arc between node } i \text{ and node } j \text{ is used} \\ 0 & otherwise \end{cases}$$

According to the given formulations, we test the performance of Algorithm-2 on generated random instances of MOKP, MOST and MOSP problems with three and four objectives. The summary of the results are presented in Table 3.13 whose details take place in Appendix A.

Table 3.13 Performance of Algorithm-2 on Random Problems *

Problem	Number of nondominated vectors (N)		Number of Models Solved (MS)		Solution Time (CPU Time) (ST)		Avg. Sol. Time (ST / N)		$\frac{MS}{N}$	
	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev	Avg.	Std. Dev
MOKP 25 items $q = 3$	211.8	150.2	449.2	278.3	22.8	9.8	0.1	0.1	2.2	0.1
MOKP 50 items $q = 3$	570.2	271.7	1224.2	542.7	438.3	280.5	0.7	0.2	2.2	0.1
MOKP 100 items $q = 3$	6786.2	2954.6	12654.6	5264.5	71149.2	60411.1	8.9	4.6	1.9	0.1
MOKP 25 items $q = 4$	425.4	155.1	4293.6	2019.7	677.4	420.6	1.5	0.4	9.8	1.3
MOST Problem 10 nodes $q = 3$	625.4	104.8	1326.2	195.8	941.5	297.4	1.5	0.3	2.1	0.1
MOSP Problem 25 nodes $q = 3$	86.4	55.9	195.2	129.4	5.2	4.7	0.1	0.0	2.2	0.1
MOSP Problem 50 nodes $q = 3$	266.2	38.9	604.8	89.4	82.8	20.5	0.3	0.0	2.3	0.0
MOSP Problem 100 nodes $q = 3$	469.6	98.7	1019.4	205.4	703.2	232.9	1.5	0.2	2.2	0.0
MOSP Problem 150 nodes $q = 3$	731.4	187.5	1538.2	374.2	3761.3	998.0	5.1	0.2	2.1	0.0
MOSP Problem 200 nodes $q = 3$	778.0	180.9	1631.4	357.2	10491.8	2848.6	13.4	0.8	2.1	0.1
MOSP Problem 25 nodes $q = 4$	205.6	71.5	1850.2	890.7	55.5	39.2	0.2	0.1	8.8	1.6
MOSP Problem 50 nodes $q = 4$	376.8	46.5	3217.4	48.5	657.7	56.0	1.8	0.3	8.6	1.0

* Averages or Standard Deviations for 5 problems per cell

If we consider the problems with $q=3$, we solve N increasing-sized models where we insert two new constraints and binary variables at each step of Algorithm-1. On the other hand, we may need to solve $\frac{N(N+1)}{2}$ problems in the worst case of Algorithm 2 (*i.e.* $O(N^2)$) if we cannot predict the optimal solution of any problem without solving the model by using the information kept in our archives. On the other hand, we will solve N models in the best case (*i.e.* $O(N)$) where we always have the opportunity to determine the next nondominated solution by using the solutions kept in the archive after we solve N models. Then, the number of models solved, MS, to find all N nondominated solutions will be in the interval $N \le MS \le \frac{N(N+1)}{2}$. Since all these MS problems are equal-sized in terms of the variables and constraints, we use the average number of models solved per nondominated solution, $\frac{MS}{N}$, as a performance measure. According to the data in the Tables 3.13, we observe that $\frac{MS}{N}$ is in the interval $[1.82, 2.33]$ with an average of 2.14 when $q=3$. That is, we roughly solve 2 models for each nondominated solution in average. This indicates the importance of the information obtained from the archives of Algorithm 2 since $MS << \frac{N(N+1)}{2}$ especially for large N values. The value of MS decreases up to 0.03 % of $\frac{N(N+1)}{2}$ as demonstrated in Table 3.14. Furthermore, the ratio, $\frac{MS}{N}$, is not so sensitive to the value of N which implies that we solve approximately the same number of models for each nondominated solution. We should also note that all models include only two additional constraints regardless of the value of N.

Table 3.14 Percentage of Models solved when $q = 3$ for all problem types

Number of nondominated vectors (N)	$\frac{MS}{N(N+1)/2} * 100\ (\%)$
32	13.45
56	7.52
76	5.81
81	5.42
84	5.49
163	2.71
168	2.72
179	2.54
182	2.47
206	2.20
249	1.81
280	1.60
283	1.60
295	1.52
298	1.54
356	1.27
375	1.18
391	1.13
434	0.99
470	0.83
486	0.92
519	0.84
534	0.81
549	0.78
554	0.76
594	0.74
599	0.70
617	0.70
655	0.66
664	0.64
693	0.61
704	0.55
721	0.58
733	0.59
784	0.54
798	0.52
843	0.49
912	0.46
1022	0.41
1056	0.39
2790	0.14
5652	0.07
6500	0.06
8288	0.04
10701	0.03

If we consider the instances with $p=4$, then $\frac{MS}{N}$ again does not seem to be sensitive to the value of N where the ratio is within the interval $[6.56, 11.50]$ with an average value of 9.06. The value of $\frac{MS}{N}$ is larger compared to the case of $q=3$ for all instances indicating that it increases with the number of objectives. We should also note that the number of models to be solved in the worst case is $1+\sum_{n=1}^{N-1}\frac{(n+1)(n+2)}{2}$ (i.e. $O(N^3)$) for $q=4$ is also larger than $\frac{N(N+1)}{2}$. We can write $N \leq MS \leq 1+\sum_{n=1}^{N-1}\frac{(n+1)(n+2)}{2}$ since the number of models to be solved in the best case is equal to N. As we discuss for $q=3$, if we consider the random instances demonstrated in Table 3.15, we observe $MS << 1+\sum_{n=1}^{N-1}\frac{(n+1)(n+2)}{2}$ especially for large N values .The value of MS decreases up to 0.02 % of $1+\sum_{n=1}^{N-1}\frac{(n+1)(n+2)}{2}$.

Table 3.15 Percentage of Models solved when $q=4$ for all Problem types

Number of nondominated vectors (N)	$\frac{MS}{1+\sum_{n=1}^{N-1}\frac{(n+1)(n+2)}{2}} * 100\,(\%)$
877	0.40
1431	0.08
1499	0.18
1700	0.11
2271	0.11
3166	0.05
3173	0.07
3175	0.02
3216	0.05
3250	0.04
3280	0.02
3581	0.03
4164	0.04
4754	0.02
7269	0.02

Although we develop two exact algorithms, Algorithm 1 and Algorithm 2, to generate all efficient solutions and Algorithm 2 provides substantial decrease in the computational times, determining all nondominated solutions may still not be very practical especially for realistically large-sized MOCO problems. The total number of efficient solutions could be prohibitively large. We also tested the performance of Algorithm 2 on a MOKP with 200 items and three objectives which has 27260 nondominated solutions. We observed that it takes very long time, 184608.70 seconds, to find all these nondominated solutions. Therefore, we propose a heuristic method to approximate the efficient frontier for MOCO problems. We test the performance of our heuristic method on the same random instances utilizing nondominated solutions of these problems we generated.

CHAPTER 4

THE HEURISTIC APPROACH

MOCO problems are typically computationally hard. Finding a single efficient solution may be hard and there may be prohibitively many efficient solutions. We develop a heuristic method to approximate the efficient frontiers of MOCO problems. Our approach is based on fitting a surface similar to the approach developed by Köksalan (1999). Using this approximation, we may search for regions that are preferred by the DM and generate the actual nondominated solutions in those preferred regions. Alternatively, we may use an interactive approach to find the best solution of the DM and the heuristic approach can be utilized to find a starting solution for such an approach.

A variation of this approach for continuous solution space problems is developed by Karasakal and Köksalan (2001) where they try to approximate the efficient surface for problems having a polyhedral solution space. They fit a weighted L_p function and estimate the p and weight values using several representative efficient solutions.

Let $(z_1^{IP}, z_2^{IP}, ..., z_q^{IP})$ and $(z_1^{NP}, z_2^{NP}, ..., z_q^{NP})$ denote the ideal point and nadir point respectively corresponding to problem (P) below:

$$
\begin{aligned}
&(P)\\
&\text{"}Max\text{"} \left\{ z_1(x), z_2(x), ..., z_q(x) \right\}\\
&\textit{subject to}\\
&x \in X
\end{aligned}
$$

where $z_i^{IP} = \max\limits_{j=1,...,N} \left(z_{i(j)} \right)$ and $z_i^{NP} = \min\limits_{j=1,...,N} \left(z_{i(j)} \right)$. We denote z_i value of j^{th} nondominated solution as $z_{i(j)}$ and the number of all nondominated solutions as N.

We should note that if we have a minimization problem, then $z_i^{IP} = \min_{j=1,\ldots,N}\left(z_{i(j)}\right)$ and $z_i^{NP} = \max_{j=1,\ldots,N}\left(z_{i(j)}\right)$.

4.1 Fitting a Surface to Approximate the Efficient Frontier

We scale the values $(z_1, z_2, \ldots,\ z_p)$ corresponding to each nondominated solution and obtain $(z_1', z_2', \ldots, z_3') = \left(\frac{z_1 - z_1^{IP}}{z_1^{NP} - z_1^{IP}}, \frac{z_2 - z_2^{IP}}{z_2^{NP} - z_2^{IP}}, \ldots, \frac{z_q - z_q^{IP}}{z_q^{NP} - z_q^{IP}}\right)$ so that $0 \le z_i' \le 1$ for $i = 1, 2, \ldots, q$. Note that we prefer the values closer to zero in each scaled objective i, since the value of z_i approaches the value of the ideal point z_i^{IP} as z_i' approaches to zero. That is, we minimize the scaled objectives. This observation is valid for both minimization and maximization type problems.

Theorem. Let

(P')
$"Min"\left\{y_1(x), y_2(x), \ldots, y_q(x)\right\}$
subject to
$x \in X$

If (P) and (P') are equivalent problems except for the objective functions where $i = 1, \ldots, q$, $y_i(x) = -z_i(x)$ and $(z_1', z_2', \ldots, z_q')$ and $(y_1', y_2', \ldots, y_q')$ are the scaled nondominated solutions corresponding to the equivalent problems (P) and (P') respectively, then $(z_1', z_2', \ldots, z_q') = (y_1', y_2', \ldots, y_q')$.

Proof. If $(z_1^{IP}, z_2^{IP}, \ldots,\ z_p^{IP})$ and $(z_1^{NP}, z_2^{NP}, \ldots,\ z_p^{NP})$ denote the ideal point and nadir point respectively corresponding to maximization problem (P), then we know $z_i^{IP} = \max_{j=1,\ldots,N}\left(z_{i(j)}\right)$ and $z_i^{NP} = \min_{j=1,\ldots,N}\left(z_{i(j)}\right)$. Multiplying all values by -1, we can write $-z_i^{IP} = \min_{j=1,\ldots,N}\left(-z_{i(j)}\right)$ and $-z_i^{NP} = \max_{j=1,\ldots,N}\left(-z_{i(j)}\right)$. We can also write ideal point and nadir point $y_i^{IP} = \min_{j=1,\ldots,N}\left(y_{i(j)}\right)$ and $y_i^{NP} = \max_{j=1,\ldots,N}\left(y_{i(j)}\right)$ since (P_y) is a minimization problem. Using the relation $y_{i(j)} = -z_{i(j)}$, we obtain $y_i^{IP} = \min_{j=1,\ldots,N}\left(-z_{i(j)}\right) = -z_i^{IP}$ and $y_i^{NP} = \max_{j=1,\ldots,N}\left(-z_{i(j)}\right) = -z_i^{NP}$. Since we scale the nondominated solutions such that $z_i' = \dfrac{z_i - z_i^{IP}}{z_i^{NP} - z_i^{IP}}$ and $y_i' = \dfrac{y_i - y_i^{IP}}{y_i^{NP} - y_i^{IP}}$, then we can write

$z_i' = \dfrac{z_i - z_i^{IP}}{z_i^{NP} - z_i^{IP}} = \dfrac{-z_i - (-z_i^{IP})}{-z_i^{NP} - (-z_i^{IP})} = \dfrac{y_i - y_i^{IP}}{y_i^{NP} - y_i^{IP}} = y_i'$ which means

$(z_1', z_2', \ldots, z_q') = (y_1', y_2', \ldots, y_q')$ □

Our approach is based on the fact that for an efficient solution if we would like to obtain a better value in i^{th} objective function, then we should sacrifice from the other criteria. In other words, as z_i' approaches to zero, then at least one objective z_j' will worsen substantially and may approach to 1. We approximate L_p function such that at the extreme hypothetical nondominated solutions we have the structure that when $z_i' = 0$, $z_j' = 1$ for all $j \neq i$. That is, the L_p curve will pass from the scaled solutions in set $S = \{(0,1,\ldots,1),(1,0,1,\ldots,1),\ldots,(1,1,\ldots,1,0)\}$. Then, we can define a surface including all the solutions in S by using the L_p distance function:

$$(1 - z_1')^p + (1 - z_2')^p + \ldots + (1 - z_q')^p = 1, \quad p \geq 0.$$

If we find a p value such that the L_p surface is close enough to the nondominated solutions, then we can approximate the efficient frontier by fitting this L_p surface. By selecting a scaled nondominated solution as a reference point $(r_1, r_2, \ldots, r_q)$, we can determine the p value such that the corresponding L_p surface will pass through this scaled nondominated solution and satisfy $(1 - r_1)^p + (1 - r_2)^p + \ldots + (1 - r_q)^p = 1$. Alternatively, we may select more than one reference point and find the p value such that the corresponding L_p surface will be at minimum distance to the reference points.

We take a single nondominated solution that is at minimum Tchebycheff distance from the ideal point as a reference point to determine the p value. That is, we select the nondominated solution $\left(r_1, r_2, ..., r_q\right) = \left(z_{1(j^*)}, z_{2(j^*)}, ..., z_{q(j^*)}\right)$ as the reference point of a maximization type problem such that $\min_{j=1,...,N}\left(\max_{i=1,...,q}\left(z_i^{IP} - z_{i(j)}\right)\right) = \max_{i=1,...,q}\left(z_i^{IP} - z_{i(j^*)}\right)$. For minimization problems, we find the reference point $\left(z_{1(j^*)}, z_{2(j^*)}, ..., z_{q(j^*)}\right)$ that satisfies $\min_{j=1,...,N}\left(\max_{i=1,...,q}\left(z_{i(j)} - z_i^{IP}\right)\right) = \max_{i=1,...,q}\left(z_{i(j^*)} - z_i^{IP}\right)$. We can solve model $P_{i(\max)}$ or $P_{i(\min)}$ to obtain reference point $\left(z_{1(j^*)}, z_{2(j^*)}, ..., z_{q(j^*)}\right)$.

$P_{i(\max)}$
$Min \quad \alpha$
subject to
$z_i^{IP} - z_i(x) \leq \alpha \qquad \forall i$
$x \in X$

$P_{i(\min)}$
$Min \quad \alpha$
subject to
$z_i(x) - z_i^{IP} \leq \alpha \qquad \forall i$
$x \in X$

After finding the p value, we find representative points on this L_p surface and rescale to find corresponding nondominated solutions.

4.2 Performance Measures

In order to evaluate how well the heuristic method approximates the efficient frontier, we find a representative point on the L_p surface for each nondominated solution. Then, we assess the corresponding values of performance measures.

4.2.1 Finding Representative Points on the L_p Surface

We determine the point on the L_p surface that is at minimum distance for each nondominated solution $\left(z_{1(j)}, z_{2(j)}, ..., z_{q(j)}\right)$ $j = 1, ..., N$ and we denote the representative point corresponding to this nondominated solution as $\left(rz_{1(j)}, rz_{2(j)}, ..., rz_{q(j)}\right)$. We also denote the scaled nondominated solution and the scaled representative point $E'_j = \left(z'_{1(j)}, z'_{2(j)}, ..., z'_{q(j)}\right)$ and $R'_j = \left(rz'_{1(j)}, rz'_{2(j)}, ..., rz'_{q(j)}\right)$, respectively.

We find the representative point sets using both Euclidean and Tchebycheff distance measures. We define following performance measures to assess how well the efficient frontier is represented.

$$\text{Average Deviation} = \frac{\sum_{i=1}^{q} \frac{\sum_{j=1}^{N} \left| rz'_{i(j)} - z'_{i(j)} \right|}{N}}{q} = \frac{\sum_{i=1}^{q} \sum_{j=1}^{N} \left| rz'_{i(j)} - z'_{i(j)} \right|}{Nq}$$

$$\text{Maximum Tchebycheff Distance} = \max_{j=1,...,N} \left(\max_{i=1,...,q} \left(\left| rz'_{i(j)} - z'_{i(j)} \right| \right) \right)$$

$$\text{Average Tchebycheff Distance} = \frac{\sum_{j=1}^{N} \left(\max_{i=1,...,q} \left(\left| rz'_{i(j)} - z'_{i(j)} \right| \right) \right)}{N}$$

4.2.1.1 Representative Points Using the Euclidean Distance Measure

We solve problem $P_{rep(j)}$ to find scaled representative point $R'_j = \left(rz'_{1(j)}, rz'_{2(j)}, ..., rz'_{q(j)}\right)$ on the L_p surface that is at minimum Euclidean distance from scaled nondominated solution $E'_j = \left(z'_{1(j)}, z'_{2(j)}, ..., z'_{q(j)}\right)$.

$P_{rep(j)}$

$$Min\sqrt{\sum_{i=1}^{q}\left(z'_{i(j)}-rz'_{i(j)}\right)^2}$$

subject to

$$\sum_{i=1}^{q}\left(1-rz'_{i(j)}\right)^p=1$$

$$0\le rz_{i(j)}\le 1$$

This corresponds to solving the following problem for each j using the p value already obtained.

Instead of solving nonlinear optimization models for each nondominated solution, we can use geometric methods to minimize Euclidean distances from points to surfaces. Since we define $R'_j=\left(rz'_{1(j)},rz'_{2(j)},...,rz'_{q(j)}\right)$ as the point on the L_p surface that is at closest Euclidean distance to $E'_j=\left(z'_{1(j)},z'_{2(j)},...,z'_{q(j)}\right)$, then the vector $\overline{E'_jR'_j}=\left(rz'_{1(j)}-z'_{1(j)},rz'_{2(j)}-z'_{2(j)},...,rz'_{q(j)}-z'_{q(j)}\right)$ must be normal to the surface at R_j. Furthermore, the vector $\vec{v}$ obtained by the partial derivatives of the L_p surface at R_j will also be normal to the L_p surface at R_j. Thus, the vector $\vec{v}$ must be parallel to the vector $\overline{E'_jR'_j}$ such that $\overline{E'_jR'_j}=t\vec{v}$ for some scalar t. If we arrange the terms of the equation for the L_p surface, then we can take the partial derivatives and find vector $\vec{v}$ as follows:

$$\sum_{i=1}^{q}\left(1-rz'_{i(j)}\right)^p-1\quad=0\qquad(1)$$

$$\frac{\partial\left(\sum_{i=1}^{q}\left(1-rz'_{i(j)}\right)^p-1\right)}{\partial rz_{i(j)}}=-p\left(1-rz'_{i(j)}\right)^{p-1}\Rightarrow$$

$$\vec{v}=\left(-p\left(1-rz'_{1(j)}\right)^{p-1},-p\left(1-rz'_{2(j)}\right)^{p-1},...,-p\left(1-rz'_{q(j)}\right)^{p-1}\right)$$

$$\overline{E_jR_j}=t\vec{v}\Rightarrow rz'_{i(j)}-z_{i(j)}=tp\left(-\left(1-rz'_{i(j)}\right)^{p-1}\right)=t'\left(-\left(1-rz'_{i(j)}\right)^{p-1}\right),\quad\forall i\qquad(2)$$

We cannot easily determine scaled representative point $R'_j = \left(rz'_{1(j)}, rz'_{2(j)}, ..., rz'_{q(j)}\right)$ from these nonlinear equations. Furthermore, we have a constraint that restricts $rz'_{i(j)}$ to take values only between 0 and 1. Thus, we employ a solver to determine $R'_j = \left(rz'_{1(j)}, rz'_{2(j)}, ..., rz'_{q(j)}\right)$. Therefore, we may still need to solve model $P_{rep(j)}$. Figures 4.1 and 4.2 illustrate the nondominated solutions and representative solutions found by using the Euclidean distance measure corresponding to a MOKP.

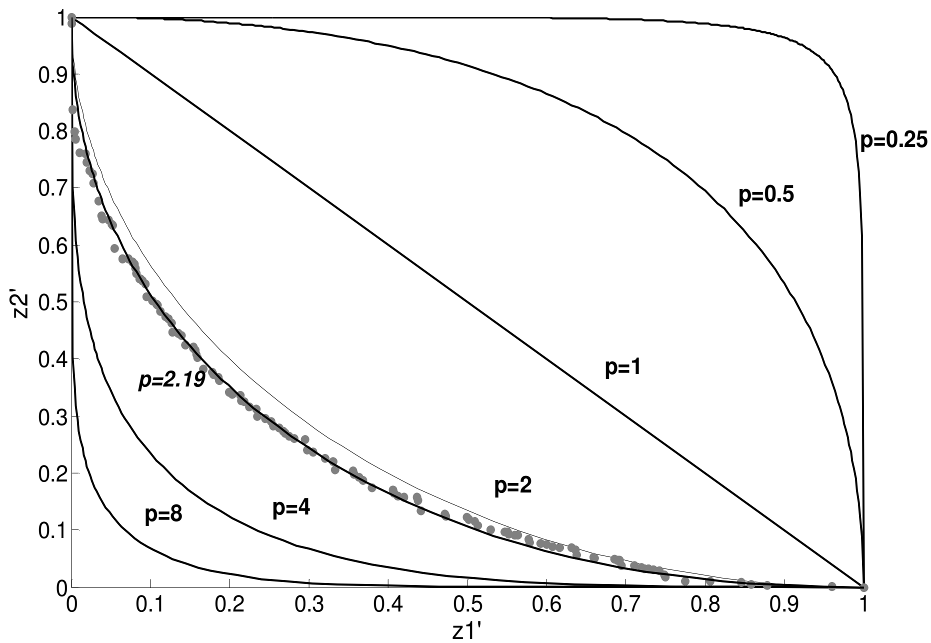
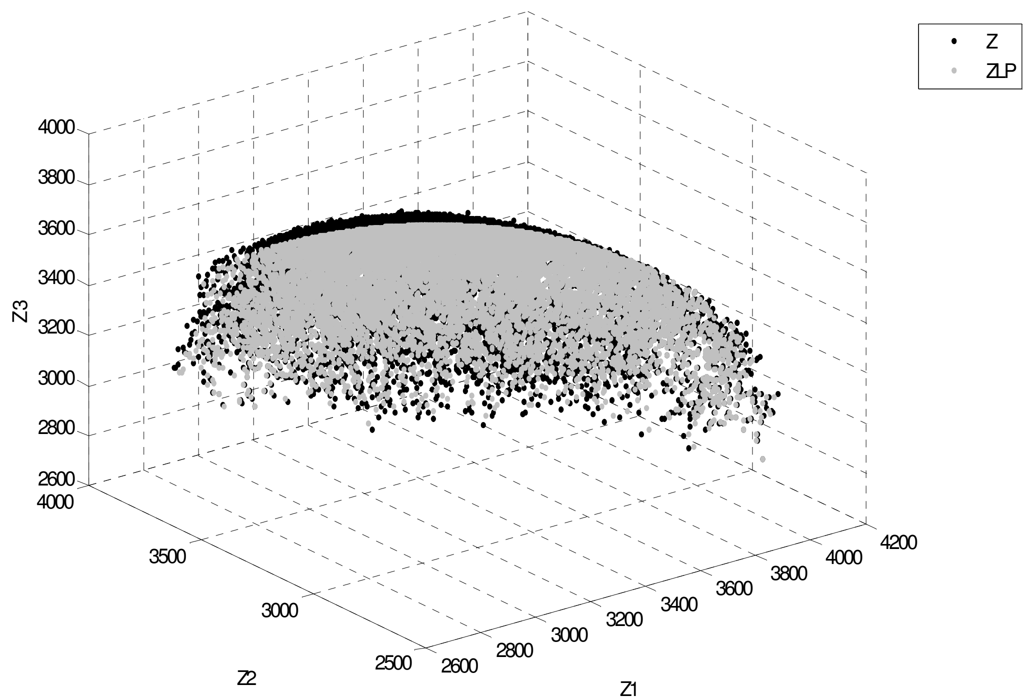


Figure 4.1 Approximation of Efficient Frontier for MOKP (100 items, $N = 126$)



• Nondominated Solutions • Representative Solutions on L_p curve

Figure 4.2 Approximation of the Efficient Frontier for MOKP

(100 items, $N = 10701$)

4.2.1.2 Representative Points Using the Tchebycheff Distance Measure

We solve problem $P'_{rep(j)}$ to determine representative point $R'_j = \left(rz'_{1(j)}, rz'_{2(j)}, ..., rz'_{q(j)}\right)$ on L_p surface that is at minimim Tchebycheff distance from nondominated solution $E'_j = \left(z'_{1(j)}, z'_{2(j)}, ..., z'_{q(j)}\right)$ using the p value already obtained.

$P'_{rep(j)}$

$$\begin{aligned}
&Min \quad \alpha \\
&subject\ to \\
&z'_{i(j)} - rz'_{i(j)} \le \alpha \qquad \forall i \\
&-z'_{i(j)} + rz'_{i(j)} \le \alpha \qquad \forall i \\
&\sum_{i=1}^{q} \left(1 - rz'_{i(j)}\right)^{p} = 1 \\
&0 \le rz'_{i(j)} \le 1 \qquad \forall i
\end{aligned}$$

4.3 Computational Experiments

We tested the performance of the heuristic procedure on the same problems of Section 3.4 for which we obtained all nondominated solutions using Algorithm-2. We found the set of representative points by using both Euclidean and Tchebycheff distances. In order to test the performance of the heuristic, we used the performance measures defined in previous chapter. The results are demonstrated in Appendix B.

We employ the average deviation as our performance measure in evaluating the representative points obtained using the Euclidean distance measure. According to Table 4.1, average deviation is in the interval $[0.002, 0.037]$ with the average value of 0.016. If we consider each problem separately, then we observe that the average deviation for MOKP is in $[0.002, 0.037]$ with the average of 0.014, for MOST problem is in $[0.004, 0.019]$ with the average of 0.012 and for MOSP is in $[0.007, 0.032]$ with the average of 0.018. Since average deviation may depend on the number of objectives and the number of nondominated solutions, we should compare the problems having the same number of objectives and approximately same number of nondominated solutions. According to this observation, we observe that the heuristic approach works well especially for MOKP where the average deviation decreases up to 0.002. We also observe that the heuristic method also works well for relatively large-sized models where even Algortihm 2 takes very long to generate all efficient solutions. For the problem with maximum number of nondominated solutions among our test problems, the average deviation is found as only 0.008 as can be seen in Table B.1 in Appendix B. This allows us to represent the efficient frontier of large-sized problems instead of generating all nondominated solutions.

Table 4.1 Performance of the Heuristic Method on Random Problems when representative points are found by using Euclidean distance measure *

Problem	Number of nondominated vectors (N)		p value		Average Deviation **	
	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP 100 items $q=2$	153.80	25.51	2.112	0.089	0.005	0.002
MOKP 200 items $q=2$	370.60	76.14	2.140	0.221	0.004	0.001
MOKP 25 items $q=3$	211.80	150.25	2.340	0.388	0.023	0.006
MOKP 50 items $q=3$	570.20	271.69	2.740	0.326	0.015	0.006
MOKP 100 items $q=3$	6786.20	2954.61	2.416	0.145	0.008	0.003
MOKP 25 items $q=4$	425.40	155.06	3.228	0.909	0.028	0.007
MOST Problem 15 nodes $q=2$	81.80	10.01	2.100	0.127	0.008	0.006
MOST Problem 10 nodes $q=3$	625.40	104.76	2.406	0.152	0.015	0.004
MOSP Problem 200 nodes $q=2$	36.20	3.49	3.346	0.619	0.015	0.005
MOSP Problem 25 nodes $q=3$	86.40	55.89	2.812	0.227	0.026	0.004
MOSP Problem 50 nodes $q=3$	266.20	38.87	3.002	0.353	0.019	0.005
MOSP Problem 100 nodes $q=3$	469.60	98.73	3.400	0.435	0.014	0.004
MOSP Problem 150 nodes $q=3$	731.40	187.51	3.698	0.248	0.012	0.003
MOSP Problem 200 nodes $q=3$	778.00	180.92	4.030	0.315	0.009	0.002
MOSP Problem 25 nodes $q=4$	205.60	71.50	3.990	0.922	0.028	0.003
MOSP Problem 50 nodes $q=4$	376.80	46.47	4.008	0.700	0.022	0.005

* Averages or Standard Deviations for 5 problems per cell

** Average for all nondominated solutions

We also tested the performance of the heuristic approach finding representative points on the L_p surface by using the Tchebycheff distance measure. We employ the maximum Tchebycheff distance and average Tchebycheff distance as our performance measures. Since we measure the Tchebycheff distances in the scaled graph, our performance measure will take a value between 0 and 1.

According to the results in Table 4.2, the maximum Tchebycheff distance is in the interval $[0.006, 0.220]$ with the average value of 0.070. It gives Tchebycheff distances corresponding to the worst represented points of the selected test problems. On the other hand, average Tchebycheff distance takes into account Tchebycheff distances for all nondominated solutions. The average Tchebycheff distance is in the interval $[0.002, 0.078]$ with the average value of 0.022.

Since the L_p surface passes through the points $S = \{(0,1,...,1),(1,0,1,...,1),...,(1,1,...,1,0)\}$, we assume at the extreme nondominated solutions we have the structure that when $z_i' = 0$, $z_j' = 1$ for all $j \neq i$. In fact, this is not exactly the case for the problems with more than two objectives. Therefore, the nondominated solutions at the extreme points may not be well represented. That may explain the difference between the maximum Tchebycheff distances and average Tchebycheff distances.

These performance measures also show that our heuristic works well even for large-sized problems for which exact algorithms are not so practical.

Table 4.2 Performance of the Heuristic Method on Random Problems when representative points are found by using Tchebycheff distance measure *

Problem	Number of nondominated vectors (N)		p value		Maximum Tchebycheff Distance		Average Tchebycheff Distance	
	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
MOKP 100 items $q=2$	153.80	25.51	2.11	0.09	0.014	0.004	0.005	0.002
MOKP 200 items $q=2$	370.60	76.14	2.14	0.22	0.012	0.004	0.005	0.002
MOKP 25 items $q=3$	211.80	150.25	2.34	0.39	0.093	0.018	0.030	0.008
MOKP 50 items $q=3$	570.20	271.69	2.74	0.33	0.062	0.017	0.020	0.009
MOKP 100 items $q=3$	6786.20	2954.61	2.42	0.14	0.039	0.011	0.010	0.004
MOKP 25 items $q=4$	425.40	155.06	3.23	0.91	0.164	0.036	0.043	0.021
MOST Problem 15 nodes $q=2$	81.80	10.01	2.10	0.13	0.024	0.012	0.010	0.006
MOST Problem 10 nodes $q=3$	625.40	104.76	2.41	0.15	0.072	0.016	0.019	0.005
MOSP Problem 200 nodes $q=2$	36.20	3.49	3.35	0.62	0.040	0.008	0.019	0.007
MOSP Problem 25 nodes $q=3$	86.40	55.89	2.81	0.23	0.099	0.007	0.034	0.006
MOSP Problem 50 nodes $q=3$	266.20	38.87	3.00	0.35	0.079	0.013	0.025	0.007
MOSP Problem 100 nodes $q=3$	469.60	98.73	3.40	0.43	0.064	0.014	0.020	0.007
MOSP Problem 150 nodes $q=3$	731.40	187.51	3.70	0.25	0.065	0.016	0.015	0.004
MOSP Problem 200 nodes $q=3$	778.00	180.92	4.03	0.32	0.051	0.006	0.012	0.002
MOSP Problem 50 nodes $q=4$	205.60	71.50	3.99	0.92	0.126	0.014	0.045	0.009
MOSP Problem 25 items $q=4$	376.80	46.47	4.01	0.70	0.116	0.034	0.033	0.007

* Averages or Standard Deviations for 5 problems per cell

Although the heuristic method represents the efficient frontier well and it is practical for large-sized problems, there may be still some difficulties especially for the problems with more than two objectives.

Problem (P_i)

$$
\begin{aligned}
&(P_i)\\
&Max\ z_i(x)+\sum_{j\neq i}\varepsilon z_j(x)\\
&subject\ to\\
&x\in X
\end{aligned}
$$

will give the nondominated solution with the best z_i value, which we denote as z_i^{IP}, for sufficiently small ε as we discuss in previous section. That is, we can find the ideal criterion vector by solving problems (P_i) for each $i=1,2,...,q$.

On the other hand, there does not exist an exact way to find the nadir criterion as Korhonen et al. (1996) discuss. For the special case, $q=2$, problem (P_1) and (P_2) will give the nondominated vector (z_1^{IP}, z_2^{NP}) and (z_1^{NP}, z_2^{IP}) respectively where $(z_1^{IP}, z_2^{IP},..., z_q^{IP})$ and $(z_1^{NP}, z_2^{NP},..., z_q^{NP})$ denote the ideal point and nadir point respectively corresponding to problem (P) below where $z_i^{IP}=\max\limits_{j=1,...,N}\left(z_{i(j)}\right)$ and $z_i^{NP}=\min\limits_{j=1,...,N}\left(z_{i(j)}\right)$. However, this cannot be generalized for the problems $q>2$.

$$
\begin{aligned}
&(P_1)\\
&Max\ z_1(x)+\varepsilon z_2(x)\\
&subject\ to\\
&x\in X
\end{aligned}
$$

$$
\begin{aligned}
&(P_2)\\
&Max\ z_2(x)+\varepsilon z_1(x)\\
&subject\ to\\
&x\in X
\end{aligned}
$$

To estimate the nadir point, we solve problems (P_i) for each $i = 1, 2, ..., q$. We denote the nondominated solution vector for the optimal solution as $\left(z_{1(i)}^{IP}, z_{2(i)}^{IP}, ..., z_i^{IP}, ..., z_{q(i)}^{IP}\right)$. $z_{j(i)}^{IP}$ is the j^{th} objective function value corresponding to the nondominated solution with the maximum z_i value. Then we estimate the nadir point from the "payoff table" such that $\hat{z}_j^{NP} = \min_{i \neq j} z_{j(i)}^{IP}$, where $\left(\hat{z}_1^{NP}, \hat{z}_2^{NP}, ..., \hat{z}_q^{NP}\right)$ denotes the estimated nadir criterion vector.

Since we have all nondominated solutions available using Algorithm-2 for the test problems, we do not need to estimate the nadir. However, we also use $\left(\hat{z}_1^{NP}, \hat{z}_2^{NP}, ..., \hat{z}_q^{NP}\right)$ instead of the known nadir point to evaluate the performance of the heuristic method even when nadir point is estimated.

According to the results in Table 4.3, the increase in the average deviations are not so significant. That implies we can use the estimated nadir point for large-sized problems without sacrificing much from the heuristic approach's performance.

We also tested whether the performance of heuristic method is sensitive to small changes in the *p* value or not. Considering all nondominated solutions, we search for a better p value which minimizes total Euclidean distance between nondominated solutions and their representative points on this corresponding L_p surface by adjusting the p value in each iteration. Although the average deviations decrease with this p value, the difference is also not much as seen in Table 4.3.

Table 4.3 Effect of p value and nadir point estimation errors on the performance of the Heuristic

		Nadir Point is known *		Nadir point is Estimated **		p Search Algorithm is employed ***	
	Number of nondominated vectors (N)	p value	Average Deviation	p value	Average Deviation	p value	Average Deviation
MOKP $q=3$	76	2.90	0.024	2.71	0.026	2.56	0.019
	163	1.87	0.027	1.53	0.036	1.89	0.027
	168	2.26	0.025	1.67	0.039	2.37	0.024
	182	2.51	0.026	1.93	0.038	2.30	0.023
	470	2.16	0.014	1.62	0.017	2.24	0.011
	280	2.62	0.012	2.2	0.015	2.55	0.011
	356	3.03	0.018	2.59	0.021	2.82	0.016
	519	3.14	0.025	2.99	0.027	2.64	0.019
	784	2.41	0.009	2.13	0.011	2.37	0.009
	912	2.50	0.013	2.04	0.019	2.36	0.009
	2790	2.65	0.009	2.64	0.009	2.59	0.009
	5652	2.39	0.004	2.19	0.005	2.39	0.004
	6500	2.40	0.006	2.25	0.007	2.40	0.006
	8288	2.39	0.013	2.14	0.015	2.31	0.012
	10701	2.25	0.008	1.84	0.011	2.27	0.008
MOKP $q=4$	207	4.85	0.037	4.33	0.044	3.03	0.026
	394	2.79	0.035	2.19	0.044	3.03	0.030
	403	2.74	0.020	2.3	0.026	2.72	0.020
	491	2.92	0.025	2.46	0.029	2.80	0.025
	632	2.84	0.025	2.11	0.034	2.83	0.025
MOSP $q=3$	32	2.80	0.029	2.32	0.047	2.55	0.027
	664	3.71	0.007	3.00	0.011	3.62	0.007

* Nadir point is obtained from the set of all nondominated solutions

** Nadir point is estimated from the "payoff table"

*** p value which minimizes the Euclidean distance between nondominated solutions and represenatative points is employed

CHAPTER 5

CONCLUSIONS

We developed two exact algorithms to generate all nondominated solutions for MOCO problems. We compared the performance of our algorithm with the algorithm proposed by Sylva and Crema (2004). Although we showed that our algorithms work much better on selected test problems including MOKP, MOST and MOSP problems, computational times increase considerably as the problem size and the number of conflicting objectives increase. This is natural since the number of nondominated solutions increase substantially as we demonstrated. Therefore, it still may not be applicable to many real-life problems.

We proposed a fast heuristic method to approximate the efficient frontier of MOCO problems. Our heuristic method is based on fitting an L_p surface to approximate the efficient frontier. We showed the method approximates the efficient frontier well on our test problems whose nondominated solutions are generated by the help of our exact algorithm. Furthermore, it can be used for realistically large sized problems since we demonstrated that it performs well for those problems.

Interacting with the DM, we may search for the preferred regions of the L_p surface and generate the actual efficient solutions in those regions. Therefore, we may not need to generate all nondominated solutions and save substantial computational effort.

As a future work, it may be a good idea to focus on a selected region and find preferred solutions incorporating decision maker's preferences. Therefore, our exact algorithms may be modified to deal with some parts of the efficient frontier. Such a procedure may prove very useful when employed together with our heuristic procedure.

We may also modify the exact algorithms by using some smart start techniques for solving the integer programs since we solve a number of models at each iteration. For example, the solutions of previous iterations can be introduced as the starting solution of the current iteration. This however, still would not make the problems where binary variables are continuously introduced very practical, since they increase the complexity substantially.

For the heuristic approach, we may test the performance when the nadir point is estimated with the minimum nadir point as a future research. This may overestimate the true nadir point and hence the range of criterion values. This in turn may possibly negatively effect the representation of the whole space. We may compare its performance with the one where the nadir is estimated from the efficient payoff table and the one where the nadir is exactly known. Since the performance may be problem dependent, it may be useful to test on different problems with different size.

REFERENCES

H.W. Corley, Efficient spanning trees, Journal of Optimization Theory and Applications, 45 (1985) 481-485

K. Deb, Multiobjective optimization using evolutionary algorithms, John Wiley & Sons, Inc., New York, 2001

M. Ehrgott and X. Gandibleux, An annotated bibliography of multiobjective combinatorial optimization, OR Spektrum, 22 (2000) 425-460

M. Ehrgott and X. Gandibleux, Approximative solution methods for multiobjective combinatorial optimization, Sociedad de Estadística e Investigación Operativa, 12(1) (2004) 1-89

F. Guerriero and R. Musmanno, Label correcting methods to solve multicriteria shortest path problems, Journal of Optimization Theory and Applications, 111(3) (2001) 589-613

H. W. Hamacher and G. Ruhe, On spanning tree problems with multiple objectives, Annals of Operations Research, 52 (1994) 209-230

H.W. Hamacher, C. R. Pederson and S. Ruzika, Finding representative systems for discrete bicriterion optimization problems, Operations Research Letters, (2006)

M. P. Hansen, Tabu search for multiobjective optimization: MOTS, In proceedings of the 13th International Conference on Multiple Criteria Decision Making ({MCDM}'97), Cape Town, South Africa, (1997)

P. Korhonen, S. Salo and R. E. Steuer, A heuristic for estimating nadir criterion values in multiple objective linear programming, Operations Research, 45 (1997) 751-757

E. K. Karasakal and M. Köksalan, Generating a Representative Subset of the Efficient Frontier in Multiple Criteria Decision Making, Working Paper No: 01-20, Faculty of Administration, University of Ottawa, (2001)

M. M. Köksalan, A heuristic approach to bicriteria scheduling, Naval Research Logistics, 46 (1999) 777-789

E. Q. V. Martins (1984), On a multicriteria shortest path problem, European Journal of Operational Research, 16 (1884) 236-245

S. Phelps and M. Köksalan, An interactive evolutionary metaheurisitc for multiobjective combinatorial optimization, Management Science, 49 (2003) 1726-1738

R.C. Prim, Shortest connection networks and some generations, Bell System Technical Journal, 36 (1957) 1389-1401

R. M. Ramos, S. Alonso, J. Sicilia and C. Gonzáles, The problem of the optimal biobjective spanning tree, European Journal of Operational Research, 111 (1998) 617-628

P. K. Shukla and K. Deb, On finding multiple pareto-optimal solutions using classical and evolutionary generating methods, European Journal of Operational Research, (2006)

S. Steiner and T. Radzik, Solving the biobjective minimum spanning tree problem using a k-best algorithm, Technical Report TR-03-06, Department of Computer Science, King's College London Strand, London, WC2R 2LS, United Kingdom (2003)

R.E. Steuer, Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley & Sons, Inc., New York (1986)

J. Sylva and A. Crema, A method for finding the set of nondominated vectors for multiple objective integer linear programs, European Journal of Operational Research, 158 (2004) 46-55

J. Sylva and A. Crema, A method for finding well-dispersed subsets of nondominated vectors for multiple objective mixed integer linear programs, European Journal of Operational Research, 180 (2007) 1011-1027

C. T. Tung and K. L. Chew, A multicriteria pareto-optimal path algorithm, European Journal of Operational Research, 62 (1992) 203-209

E. L. Ulungu, J. Teghem, P. H. Fortemps and D. Tuyttens, MOSA method: A tool for solving multiobjective combinatorial optimization problems, Journal of Multi-criteria Decision Analysis, 8 (1999) 221-236

M. Visée, J. Teghem, M. Pirlot and E. L. Ulungu, Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem, Journal of Global Optimization, 12 (1998) 139-155

G. Zhou and M. Gen, Genetic algorithm approach on multi-criteria minimum spanning tree problem, European Journal of Operational Research, 114 (1999) 141-152

E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation, 3(4) (1999) 257-271

APPENDIX A

RESULTS FOR ALGORITHM 2

Table A.1 Performance of Algorithm-2 on Random Knapsack Problems

q	Number of items	Problem	Number of nondominated vectors (N)	Number of Models Solved (MS)	Solution Time (CPU Time in seconds) (ST)	Average Solution Time (ST/N)	$\frac{MS}{N}$
3	25	1	76	170	7.41	0.10	2.24
		2	163	362	19.45	0.12	2.22
		3	168	386	32.03	0.19	2.30
		4	182	411	28.63	0.16	2.26
		5	470	917	26.48	0.06	1.95
	50	1	280	629	184.23	0.66	2.25
		2	356	809	217.77	0.61	2.27
		3	519	1128	312.17	0.60	2.17
		4	784	1655	790.45	1.01	2.11
		5	912	1900	686.63	0.75	2.08
	100	1	2790	5493	9636.91	3.45	1.97
		2	5652	10553	43992.41	7.78	1.87
		3	6500	12476	43369.13	6.67	1.92
		4	8288	15079	93843.58	11.32	1.82
		5	10701	19672	164904.06	15.41	1.84
4	25	1	207	1700	228.51	1.10	8.21
		2	394	4164	470.03	1.19	10.57
		3	403	3581	574.31	1.43	8.89
		4	491	4754	770.80	1.57	9.68
		5	632	7269	1343.59	2.13	11.50

Table A.2 Performance of Algorithm-2 on Random Minimum Spanning Tree Problems

q	Number of nodes	Problem	Number of nondominated vectors (N)	Number of Models Solved (MS)	Solution Time (CPU Time in seconds) (ST)	Average Solution Time (ST / N)	$\frac{MS}{N}$
3	10	1	486	1083	800.80	1.65	2.23
		2	549	1179	717.01	1.31	2.15
		3	655	1419	841.05	1.28	2.17
		4	704	1376	887.19	1.26	1.95
		5	733	1574	1461.55	1.99	2.15

Table A.3 Performance of Algorithm-2 on Random Shortest Path Problems

q	Number of nodes	Problem	Number of nondominated vectors (N)	Number of Models Solved (MS)	Solution Time (CPU Time in seconds) (ST)	Average Solution Time (ST / N)	$\frac{MS}{N}$
3	25	1	32	71	1.19	0.04	2.22
		2	56	120	2.34	0.04	2.14
		3	81	180	4.61	0.06	2.22
		4	84	196	4.67	0.06	2.33
		5	179	409	13.19	0.07	2.28
	50	1	206	468	57.36	0.28	2.27
		2	249	563	71.41	0.29	2.26
		3	283	643	80.16	0.28	2.27
		4	295	665	109.78	0.37	2.25
		5	298	685	95.50	0.32	2.30
	100	1	375	829	521.63	1.39	2.21
		2	391	867	489.34	1.25	2.22
		3	434	931	597.61	1.38	2.15
		4	554	1170	981.03	1.77	2.11
		5	594	1300	926.63	1.56	2.19
	150	1	599	1255	2906.81	4.85	2.10
		2	617	1328	3276.38	5.31	2.15
		3	664	1403	3403.64	5.13	2.11
		4	721	1521	3757.91	5.21	2.11
		5	1056	2184	5461.56	5.17	2.07
	200	1	534	1164	6602.11	12.36	2.18
		2	693	1478	9743.00	14.06	2.13
		3	798	1642	10548.70	13.22	2.06
		4	843	1734	11011.99	13.06	2.06
		5	1022	2139	14553.06	14.24	2.09
4	25	109	877	20.42	0.19	8.05	109
		170	1499	33.09	0.19	8.82	170
		218	1431	41.55	0.19	6.56	218
		230	2271	62.48	0.27	9.87	230
		301	3173	120.06	0.40	10.54	301
	50	337	3166	727.78	2.16	9.39	337
		338	3216	624.92	1.85	9.51	338
		355	3250	695.36	1.96	9.15	355
		423	3175	654.28	1.55	7.51	423
		431	3280	585.97	1.36	7.61	431

APPENDIX B

RESULTS FOR THE HEURISTIC METHOD

Table B.1 Performance of the Heuristic on Random Knapsack Problems when representative points are found by using Euclidean distance measure

q	Number of items	Problem	Number of nondominated vectors (N)	p value	Average Deviation
2	100	1	126	2.19	0.003
		2	137	2.09	0.004
		3	147	2.03	0.002
		4	170	2.22	0.008
		5	189	2.03	0.005
	200	1	266	2.47	0.004
		2	339	2.24	0.004
		3	384	2.05	0.005
		4	390	2.04	0.006
		5	474	1.90	0.002
3	25	1	76	2.90	0.024
		2	163	1.87	0.027
		3	168	2.26	0.025
		4	182	2.51	0.026
		5	470	2.16	0.014
	50	1	280	2.62	0.012
		2	356	3.03	0.018
		3	519	3.14	0.025
		4	784	2.41	0.009
		5	912	2.50	0.013
	100	1	2790	2.65	0.009
		2	5652	2.39	0.004
		3	6500	2.40	0.006
		4	8288	2.39	0.013
		5	10701	2.25	0.008
4	25	1	207	4.85	0.037
		2	394	2.79	0.035
		3	403	2.74	0.020
		4	491	2.92	0.025
		5	632	2.84	0.025

Table B.2 Performance of the Heuristic on Random Minimum Spanning Tree Problems when representative points are found by using Euclidean distance measure

q	Number of nodes	Problem	Number of nondominated vectors (N)	p value	Average Deviation
2	15	1	68	1.97	0.006
		2	76	2.26	0.006
		3	85	2.08	0.004
		4	86	1.99	0.009
		5	94	2.20	0.018
3	10	1	486	2.48	0.010
		2	549	2.19	0.017
		3	655	2.44	0.019
		4	704	2.33	0.013
		5	733	2.59	0.016

Table B.3 Performance of the Heuristic on Random Shortest Path Problems when representative points are found by using Euclidean distance measure

q	Number of nodes	Problem	Number of nondominated vectors (N)	p value	Average Deviation
2	200	1	32	3.87	0.015
		2	34	3.30	0.018
		3	36	3.11	0.017
		4	38	2.46	0.007
		5	41	3.99	0.017
3	25	1	32	2.80	0.029
		2	56	2.99	0.021
		3	81	2.80	0.029
		4	84	3.02	0.027
		5	179	2.45	0.022
	50	1	206	3.49	0.022
		2	249	3.07	0.016
		3	283	3.11	0.024
		4	295	2.56	0.020
		5	298	2.78	0.013
	100	1	375	3.89	0.013
		2	391	3.00	0.014
		3	434	3.78	0.022
		4	554	2.94	0.011
		5	594	3.39	0.011
	150	1	599	3.57	0.015
		2	617	3.95	0.014
		3	664	3.71	0.007
		4	721	3.91	0.011
		5	1056	3.35	0.013
	200	1	534	3.77	0.008
		2	693	4.32	0.012
		3	798	3.95	0.009
		4	843	3.71	0.008
		5	1022	4.40	0.008
4	25	1	109	4.97	0.030
		2	170	4.90	0.027
		3	218	3.64	0.032
		4	230	2.83	0.024
		5	301	3.61	0.026
	50	1	337	3.25	0.022
		2	338	4.92	0.019
		3	355	3.53	0.021
		4	423	3.80	0.030
		5	431	4.54	0.019

Table B.4 Performance of the Heuristic on Random Knapsack Problems when representative points are found by using Tchebycheff distance measure

q	Number of items	Problem	Number of nondominated vectors (N)	p value	Maximum Tchebycheff Distance	Average Tchebycheff Distance
2	100	1	126	2.19	0.012	0.004
		2	137	2.09	0.014	0.005
		3	147	2.03	0.010	0.003
		4	170	2.22	0.019	0.009
		5	189	2.03	0.017	0.006
	200	1	266	2.47	0.014	0.005
		2	339	2.24	0.010	0.005
		3	384	2.05	0.013	0.005
		4	390	2.04	0.017	0.007
		5	474	1.90	0.006	0.002
3	25	1	76	2.90	0.082	0.033
		2	163	1.87	0.114	0.033
		3	168	2.26	0.098	0.032
		4	182	2.51	0.105	0.034
		5	470	2.16	0.068	0.016
	50	1	280	2.62	0.057	0.015
		2	356	3.03	0.082	0.023
		3	519	3.14	0.077	0.034
		4	784	2.41	0.048	0.011
		5	912	2.50	0.044	0.016
	100	1	2790	2.65	0.054	0.011
		2	5652	2.39	0.030	0.005
		3	6500	2.40	0.029	0.008
		4	8288	2.39	0.049	0.016
		5	10701	2.25	0.035	0.010
4	25	1	207	4.85	0.165	0.078
		2	394	2.79	0.220	0.045
		3	403	2.74	0.171	0.026
		4	491	2.92	0.134	0.034
		5	632	2.84	0.130	0.032

Table B.5 Performance of the Heuristic on Random Minimum Spanning Tree Problems when representative points are found by using Tchebycheff distance measure

q	Number of nodes	Problem	Number of nondominated vectors (N)	p value	Maximum Tchebycheff Distance	Average Tchebycheff Distance
2	15	1	68	1.97	0.017	0.007
		2	76	2.26	0.019	0.007
		2	85	2.08	0.017	0.005
		3	86	1.99	0.023	0.010
		4	94	2.20	0.046	0.021
3	10	1	486	2.48	0.055	0.012
		2	549	2.19	0.084	0.020
		3	655	2.44	0.090	0.024
		4	704	2.33	0.056	0.016
		5	733	2.59	0.077	0.022

Table B.6 Performance of the Heuristic on Random Shortest Path Problems when representative points are found by using Tchebycheff distance measure

q	Number of nodes	Problem	Number of nondominated vectors (N)	p value	Maximum Tchebycheff Distance	Average Tchebycheff Distance
2	200	1	32	3.87	0.042	0.020
		2	34	3.30	0.040	0.024
		3	36	3.11	0.047	0.023
		4	38	2.46	0.026	0.008
		5	41	3.99	0.045	0.022
3	25	1	32	2.80	0.108	0.042
		2	56	2.99	0.103	0.029
		3	81	2.80	0.096	0.037
		4	84	3.02	0.100	0.036
		5	179	2.45	0.090	0.027
	50	1	206	3.49	0.079	0.031
		2	249	3.07	0.076	0.021
		3	283	3.11	0.086	0.034
		4	295	2.56	0.095	0.024
		5	298	2.78	0.059	0.017
	100	1	375	3.89	0.073	0.020
		2	391	3.00	0.069	0.018
		3	434	3.78	0.072	0.033
		4	554	2.94	0.067	0.014
		5	594	3.39	0.039	0.016
	150	1	599	3.57	0.052	0.021
		2	617	3.95	0.065	0.011
		3	664	3.71	0.065	0.011
		4	721	3.91	0.052	0.016
		5	1056	3.35	0.090	0.018
	200	1	534	3.77	0.050	0.011
		2	693	4.32	0.048	0.016
		3	798	3.95	0.044	0.012
		4	843	3.71	0.055	0.010
		5	1022	4.40	0.058	0.011
4	25	1	109	4.97	0.138	0.054
		2	170	4.90	0.124	0.049
		3	218	3.64	0.140	0.052
		4	230	2.83	0.121	0.032
		5	301	3.61	0.105	0.040
	50	1	337	3.25	0.157	0.032
		2	338	4.92	0.075	0.029
		3	355	3.53	0.142	0.029
		4	423	3.80	0.116	0.045
		5	431	4.54	0.091	0.031