MULTI-OBJECTIVE INTEGER PROGRAMMING:
A GENERAL APPROACH FOR GENERATING ALL EFFICIENT SOLUTIONS

Melih Özlen[1]
Department of Industrial Engineering,
Hacettepe University, Ankara 06800, Turkey
Phone: +90 312 297 6885
Fax: +90 312 297 2078
E-Mail: mozlen@hacettepe.edu.tr

Meral Azizoğlu
Department of Industrial Engineering,
Middle East Technical University, Ankara 06531, Turkey
Phone: +90 312 210 22 81
Fax: +90 312 210 47 86
E-Mail: meral@ie.metu.edu.tr

Abstract

In this paper we develop a general approach to generate all efficient solutions of the Multi-Objective Integer Programming (MOIP) Problem. Our approach, which is based on identification of objective efficiency ranges, is an improvement over classical ε-constraint method. Objective efficiency ranges are identified by solving simpler MOIP problems with fewer objectives. We first provide the classical ε-constraint method on the Bi-Objective Integer Programming problem for the sake of completeness and comment on its efficiency. Then present our method on Tri-Objective Integer Programming problem and then extend it to the general MOIP problem with *k* objectives. A numerical example considering Tri-Objective Assignment problem is also provided.

Keywords: Multiple objective programming, Integer Programming

[1] *Corresponding Author*

1. Introduction

Multi-Objective Integer Programming (MOIP) is an important research area as many practical situations require discrete representations by integer variables and many decision makers have to deal with several objectives. Some note-worthy practical environments where the MOIP problems find their applications are supply chain design, logistics planning, scheduling and financial planning. The MOIP problems are theoretically challenging as well, as most of them, even their single objective versions, fall into the class of computationally intractable problems. Our aim in this study is to provide a general framework for the solution of the MOIP problems with the hope of facilitating this involved decision making process.

Multi-Objective Combinatorial Optimization (MOCO) problems are special cases of the MOIP problems that are distinguished due to their special structured constraint sets. Bi-objective, tri-objective or multi-objective versions of scheduling, shortest path, assignment, traveling salesman, minimum spanning tree are some noteworthy MOCO problems. Ehrgott and Gandibleux (2000) and Ehrgott and Gandibleux (2004) review the MOCO literature on the exact and approximate methods, respectively. They address some special problem types and discuss their solution methodologies. Ehrgott and Gandibleux (2002) survey some other Multi Criteria Optimization problems including, but not limited to, non-linear programming, scheduling, multi-level programming.

In recent years, there have been many developments for some particular MOCO problems. However the research on the MOIP problems, is still scarce. Klein and Hannan (1982), Sylva and Crema (2004) and Sylva and Crema (2007) study the Multi-Objective Mixed Integer Programming models. Klein and Hannan (1982) develop an approach based on the sequential solutions of the Single-Objective models which use some additional constraints to eliminate the known dominated solutions. Their algorithm generates a subset, but not necessarily the whole set, of all efficient solutions. Sylva and Crema (2004) improve Klein and Hannan (1982)'s approach by incorporating all objectives in a weighted function and guarantee to generate all efficient solutions. Sylva and Crema (2007) propose a similar approach for the problem of maximizing the infinity-norm distance from the set of known dominated solutions. They show that their approach provides a more uniform representation of the efficient set when compared to the approach by Sylva and Crema (2004). The major drawback of all three approaches is the difficulty in solving the constrained problems due to the increasing number of constraints and binary variables with an increase in the number of the efficient solutions.

Klamroth et al. (2004) and Ehrgott (2006) study the general MOIP problem. Klamroth et al. (2004) define composite functions to obtain upper bounds on the objective function values of the efficient solutions and discuss the use of the upper bounds in generating the efficient set. To form the composite functions, they propose some classical optimization methods like cutting plane method and branch and bound algorithm. Ehrgott (2006) develops some properties of the efficient solutions and proposes a scalarization technique to identify any efficient solution.

In this study we develop a method to identify individual objective efficiency ranges for problems with more than three objectives. To best of our knowledge our method is the first of its kind to identify these ranges without generating whole set. We use the solutions of simpler (having less objectives) multi-objective problems to obtain efficiency ranges for each objective. We use these ranges in constrained multi-objective problems and develop a method to generate all efficient solutions for the tri-objective integer programming (TOIP) problems and the general multi-objective integer programming (MOIP) problems. In contrast to the classical methods that search in feasibility ranges with incremental steps, our method searches within narrower efficiency ranges jumping between efficient solutions with large steps.

We compare the theoretical worst case performances of our method and the classical ε-constraint method. We illustrate the performances of the methods on a tri-objective assignment problem.

The rest of the paper is presented s follows. In Section 2, we state the well known classical ε-constraint method for Bi-Objective IP problems. In Section 3 we present our algorithm for the Tri-Objective IP problems. We give the generalization of our method to the MOIP problem with *k* objectives in Section 4. In Section 5, we illustrate our method on a tri-objective assignment problem instance and compare it with the classical ε-constraint method. We conclude in Section 6.

2. Bi-Objective Integer Programming (BOIP) Problem

The Bi-Objective Integer Programming (BOIP) problem is a special case of the Multi-Objective Integer Programming (MOIP) problem with two objectives. The BOIP problem is defined as:

(BOIP) Min $f_1(x)$
Min $f_2(x)$
s.t. $x \in X$

where X is the set of feasible solutions in which $x_j \geq 0$ and integer for all $j \in \{1, 2, ..., n\}$.

The individual objectives are defined as $f_1(x) = \sum_{j=1}^{n} c^1{}_j x_j$ and $f_2(x) = \sum_{j=1}^{n} c^2{}_j x_j$ where $c^i{}_j$ is integer for all $i \in \{1, 2\}$ and $j \in \{1, 2, ..., n\}$.

A solution $x' \in X$ is bi-objective efficient if and only if there is no $x \in X$ such that $f_i(x) \leq f_i(x')$ for $i \in \{1, 2\}$ and $f_i(x) < f_i(x')$ for at least one i.

We provide classical ε-constraint method on BOIP for the sake of completeness and then discuss its efficiency. Some theory on BOIP problems is provided to clarify the presentations in Sections 3 and 4.

The Constrained Weighted Single Objective Integer Programming (CWSOIP) problem is defined as:

(CWSOIP) Min $f_1(x) + w_2 f_2(x)$
s.t. $f_2(x) \leq l_2$
$x \in X$

The optimal solution of the CWSOIP problem is bi-objective efficient for the properly set values of w_2 and l_2. Classical ε-constraint method to generate all efficient solutions initializes by setting l_2 to a general upper bound, and then decreases the l_2 value systematically and stops when l_2 reaches the lower bound on $f_2(x)$.

Theorem 2.1 presents a property of an upper bound on the $f_i(x)$ value of all bi-objective efficient solutions provided that $f_i(x)$ value is no more than l_i.

Theorem 2.1. A bi-objective efficient solution providing an upper bound on $f_i(x)$, for a specified l_i value, gives a lower bound on the other objective, $f_j(x)$.

A range for w_2 values that leads to an efficient solution when used in the CWSOIP model is identified using f_2^{GUB} and f_2^{GLB} , the upper and lower bounds on the $f_2(x)$ value of any feasible solution.

Using these bounds, Theorem 2.2 states a range for w_2 that returns an efficient solution when used in the CWSOIP model for a specified l_2 value.

Theorem 2.2. The solution of the CWSOIP problem with $w_2 = \dfrac{1}{f_2^{GUB} - f_2^{GLB} + 1}$ provides an upper bound on $f_2(x)$ values of all bi-objective efficient solutions for a specified l_2 value.

Theorem 2.3 states that for a properly selected w_2 value, the following Weighted Single Objective Integer Programming (WSOIP) problem gives an upper bound on $f_2(x)$ values of all bi-objective efficient solutions

(WSOIP) Min $f_1(x) + w_2 f_2(x)$

s.t. $x \in X$

Theorem 2.3. The optimal solution of the WSOIP problem with $w_2 = \dfrac{1}{f_2^{GUB} - f_2^{GLB} + 1}$ provides an upper bound on the $f_2(x)$ value of all bi-objective efficient solutions.

We now provide the stepwise description of classical ε-constraint method for the generation of all efficient solutions for the BOIP problem.

Procedure 2.1. Generating all Efficient Solutions of a Bi-Objective Problem

(classical ε-constraint method)

Step 0. Find f_2^{GUB} and f_2^{GLB}.

Let $w_2 = \dfrac{1}{f_2^{GUB} - f_2^{GLB} + 1}$

Let $l_2 = f_2^{GUB}$

Step 1. Solve the CWSOIP problem with l_2.

If the solution is infeasible, then STOP.

Step 2. Let the optimal solution be x^*.

$E = E \cup (f_1(x^*), f_2(x^*))$

$l_2 = f_2(x^*) - 1$

Go to Step 1

The first solution returned by the procedure is (f_1^{GLB}, f_2^{BUB}), and the last solution is (f_1^{BUB}, f_2^{GLB}), where f_1^{BUB}, f_2^{BUB}, f_1^{GLB}, f_2^{GLB} are upper and lower bounds on the $f_1(x)$ and $f_2(x)$ values of all bi-objective efficient solutions, respectively. In other words, $f_1(x)$ values of all efficient solutions are between f_1^{GLB} and f_1^{BUB}, and their $f_2(x)$ values are between f_2^{GLB} and f_2^{BUB}.

The procedure iterates as the number of efficient solutions, which is upper bounded by the pseudo-polynomial value, $Min\{f_1^{BUB} - f_1^{GLB} + 1, f_2^{BUB} - f_2^{GLB} + 1\}$. Each iteration returns a new efficient solution by solving the CWSOIP problem. Hence the complexity of Procedure 2.1 depends on the complexity of the specific CWSOIP problem solved in Step 1.

The classical ε-constraint method only iterates within objective efficiency ranges, and each solved IP identifies a new efficient solution. The objective efficiency ranges are easy to find for the BOIP problem using hierarchical optimization. However no similar efficient method is available for the TOIP or general MOIP problems. For these problems, we present a way to identify objective efficiency ranges and use the ranges in developing a new method to generate all efficient solutions, in Sections 3 and 4.

3. Tri-Objective Integer Programming (TOIP) Problem

The Tri-Objective Integer Programming (TOIP) Problem is a special case of the MOIP problem with three objectives. The TOIP problem is defined as:

(TOIP) Min $f_1(x)$

Min $f_2(x)$

Min $f_3(x)$

s.t. $x \in X$

where X is the set of feasible solutions in which $x_j \geq 0$ and integer for all $j \in \{1,2,...,n\}$.

The individual objectives are defined as $f_1(x) = \sum_{j=1}^{n} c^1{}_j x_j$, $f_2(x) = \sum_{j=1}^{n} c^2{}_j x_j$ and $f_3(x) = \sum_{j=1}^{n} c^3{}_j x_j$ where $c^i{}_j$ is integer for all $i \in \{1,2,3\}$ and $j \in \{1,2,...,n\}$.

A solution $x' \in X$ is tri-objective efficient if and only if there is no $x \in X$ such that $f_i(x) \leq f_i(x')$ for all $i \in \{1,2,3\}$ and $f_i(x) < f_i(x')$ for at least one i.

We first present our method to solve TOIP problem, and then provide the classical ε-constraint method extension, and then compare the two algorithms in terms of their efficiency.

The Constrained Weighted Bi-Objective Integer Programming (CWBOIP) Problem is defined as:

$$
\begin{array}{lll}
\text{(CWBOIP)} & \text{Min} & f_1(x)+w_3 f_3(x) \\
 & \text{Min} & f_2(x)+w_3 f_3(x) \\
 & \text{s.t.} & f_3(x)\le l_3 \\
 & x \in X &
\end{array}
$$

Note that, we consider $f_3(x)$, together with $f_1(x)$ and $f_2(x)$ to ensure tri-objective efficiency. Moreover we bound $f_3(x)$ by adding the constraint, $f_3(x)\le l_3$. The solution of the CWBOIP problem provides a set of bi-objective efficient solutions with respect to $f_1(x)$ and $f_2(x)$, that are also tri-objective efficient, for the properly set w_3 and l_3 values. All tri-objective efficient solutions can be found by solving the CWBOIP problem for all possible w_3 and l_3 values. However, enumerating all possible values explicitly, may be very impractical, even infeasible, in many cases. Recognizing this fact, we find the right way to set w_3 and l_3 values.

To generate the tri-objective efficient set, efficiently, we initially set l_3 to the general upper bound on $f_3(x)$ value of all feasible solutions and form the bi-objective efficient set with respect to $f_1(x)$ and $f_2(x)$. We then generate the other bi-objective efficient solution sets by decreasing l_3 value systemically, towards the set of bi-objective efficient solutions that provide a lower bound on $f_3(x)$.

Lemma 3.1 presents a property of an upper bound on the $f_i(x)$ value of all tri-objective efficient solutions provided that $f_i(x)$ value is no more than l_i.

Lemma 3.1. A tri-objective efficient solution that provides an upper bound on objective $f_i(x)$, for a specified l_i value, is bi-objective efficient with respect to other two objectives.

Proof. A tri-objective efficient solution $x'\in X$ providing an upper bound on $f_i(x)$, satisfies $f_i(x')\ge f_i(x)$, for all $x\in X$, hence is not non-dominated in objective i. The definition of efficiency follows that, an efficient solution has at least one non-dominated objective by any other efficient solution. So, a solution dominated by all solutions in one objective should be bi-objective efficient in other two objectives in order have at least one non-dominated objective. ■

We next find a range for w_3 values that leads to an efficient solution when used in the CWBOIP problem. In doing so, we define f_3^{GUB} and f_3^{GLB} as the respective general upper and lower bounds on $f_3(x)$ value of any feasible solution. As in bi-objective case, the general bounds can be obtained by solving the individual single objective problems.

Using these bounds, Lemma 3.2 specifies a range for w_3 that guarantees the generation of tri-objective efficient solutions for a specified l_3 value.

Lemma 3.2. The solution of the CWBOIP problem (by Procedure 2.1) with $w_3 = \dfrac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)}$ provides an upper bound on the $f_3(x)$ value of all tri-objective efficient solutions for a specified l_3 value.

Proof. We solve the following CWBOIP problem in Step 1 of Procedure 2.1:

$$\begin{aligned}\text{Min }\ & f_1(x) + \frac{1}{f_2^{GUB} - f_2^{GLB} + 1} f_2(x) + w_3 f_3(x) \\ \text{s.t. }\ & f_2(x) \le l_2 \\ & f_3(x) \le l_3 \\ & x \in X\end{aligned}$$

From Lemma 3.1 we know that the tri-objective efficient solutions having the highest $f_3(x)$ value should be bi-objective efficient with respect to $f_1(x)$ and $f_2(x)$. To find the tri-objective upper bound on $f_3(x)$, one could generate the bi-objective efficient solutions with respect to $f_1(x)$ and $f_2(x)$, thus guarantee bi-objective efficiency and minimize $f_3(x)$ over the bi-objective efficient set. In place of this two-step hierarchical procedure, we use a single objective in the CWSOIP problem and generate the bi-objective efficient solutions that are also tri-objective efficient.

The CWSOIP problem provides a bi-objective efficient set having the highest $f_3(x)$ value, if w_3 is defined such that the largest decrease in $f_3(x)$, will not contribute to the objective function as much as the smallest increase in either $f_1(x)$ or $f_2(x)$. Since Procedure 2.1 hierarchically optimizes $f_1(x)$ and then $f_2(x)$, the contribution of $f_2(x)$ is always smaller than that of $f_1(x)$, so one can only consider the contribution by $f_2(x)$ rather than considering both objectives. The largest decrease in $f_3(x)$ is no more than $(f_3^{GUB} – f_3^{GLB})$, and its contribution to the objective function is no more than $w_3(f_3^{GUB} – f_3^{GLB})$. The smallest increase in $f_2(x)$ is no less than $\dfrac{1}{f_2^{GUB} - f_2^{GLB} + 1}$, due to the integrality of the parameters and decision variables. This follows, when

$\frac{1}{f_2^{GUB} - f_2^{GLB} + 1} > w_3(f_3^{GUB} - f_3^{GLB})$, equivalently $w_3 < \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB})}$, the largest decrease in $f_3(x)$ is no more than the smallest increase in $f_2(x)$. As, $\frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)} < \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB})}$, the CWBOIP problem with $w_3 = \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)}$ provides an upper bound on the $f_3(x)$ values of the tri-objective efficient solutions, for a specified l_3 value. ■

Through Lemma 3.3 we state that for a properly selected w_3 value, the following Weighted Bi-Objective Integer Programming (WBOIP) problem gives an upper bound on $f_3(x)$ values of all tri-objective efficient solutions.

(WBOIP) Min $f_1(x) + w_3 f_3(x)$

Min $f_2(x) + w_3 f_3(x)$

s.t. $x \in X$

Lemma 3.3. The optimal solution of the WBOIP problem (by Procedure 2.1) with $w_3 = \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)}$ provides an upper bound on the $f_3(x)$ value of all tri-objective efficient solutions.

Proof. Lemma 3.2 states that the optimal solution of the CWBOIP problem with $w_3 = \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)}$ provides an upper bound on the $f_3(x)$ value of all tri-objective efficient solutions for a specified l_3 value. Once we set l_3 to a very big number no less than f_3^{GUB}, the CWBOIP problem does not skip any tri-objective efficient solution and the optimal solution provides an upper bound on the $f_3(x)$ value of all tri-objective efficient solutions. Moreover the CWBOIP problem reduces to the WBOIP problem, as the constraint $f_3(x) \le l_3$ becomes redundant. ■

We now state the stepwise description of our procedure to generate all tri-objective efficient solutions.

Procedure 3.1 Generating all Tri-Objective Efficient Solutions

Step 0. Find f_2^{GUB}, f_2^{GLB}, f_3^{GUB} and f_3^{GLB}.

Let $w_3 = \dfrac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)}$

Let $l_3 = f_3^{GUB}$

Step 1. Solve the CWBOIP problem with l_3 using Procedure 2.1.

If the solution is infeasible, then STOP.

Step 2. Let the bi-objective solutions set be BE^*.

$E = E \cup BE^*$

Set $l_3 = \max\{f_3(x), x \in BE^*\}$ - 1.

Go to Step 1

Set E returned by Procedure 3.1 resides all tri-objective efficient solutions. Lemma 3.4 states this result formally.

Lemma 3.4. Procedure 3.1 generates all tri-objective efficient solutions.

Proof. In Step 0, no feasible, hence efficient, solution is eliminated, as l_3 is set to the general upper bound value on $f_3(x)$. From Lemma 3.2 we know that, Step 1 provides an efficient solution whose $f_3(x)$ value, $\max\{f_3(x), x \in BE^*\}$, is an upper bound on the $f_3(x)$ value of all efficient solutions for a specified l_3 value, by identifying the bi-objective efficient set. As $\max\{f_3(x), x \in BE^*\}$ is an upper bound, there cannot exist any tri-objective efficient solution whose $f_3(x)$ value is between $\max\{f_3(x), x \in BE^*\} + 1$ and l_3. Due to the integrality of the parameters and decision variables, the next $f_3(x)$ value is upper bounded by $\max\{f_3(x), x \in BE^*\}$-1. So setting l_3 to $\max\{f_3(x), x \in BE^*\}$- 1 does not eliminate any tri-objective efficient solution, and identify the next efficient solution if there exists any. Each iteration identifies a bi-objective efficient solution set by solving the CWBOIP problem for a specified l_3 value. The procedure terminates when the CWBOIP problem returns an infeasible solution. An infeasible solution shows that there is no tri-objective efficient solution having $f_3(x)$ value smaller than or equal to the last specified l_3 value, thus all tri-objective efficient solutions are identified. ■

Procedure 3.1 starts with $l_3 = f_3^{GUB}$ and finds f_3^{TUB}, i.e., an upper bound on the $f_3(x)$ values of all tri-objective efficient solutions. The last $f_3(x)$ value returned by the algorithm, f_3^{GLB} , is a lower bound on the $f_3(x)$ value of all tri-objective efficient solutions. Hence the $f_3(x)$ values of all tri-objective efficient solutions are between f_3^{GLB} and f_3^{TUB}.

The procedure iterates at most as the product of any two objective ranges which is upper bounded by pseudo-polynomial value Max $\{(f_1^{TUB} - f_1^{GLB} +1)\ (f_2^{TUB} - f_2^{GLB} +1),\ (f_1^{TUB} - f_1^{GLB} +1)(\ f_3^{TUB} - f_3^{GLB} +1),\ (f_2^{TUB} - f_2^{GLB} +1)(\ f_3^{TUB} - f_3^{GLB} +1)\}$. Hence the complexity of the procedure depends on the complexity of the specific CWBOIP problem.

An upper bound on the number of iterations can also be expressed as a function of the number of tri-objective efficient solutions, $|E|$. In Procedure 3.1, each update of l_3 eliminates one tri-objective efficient solution from the search space. The maximum number of bi-objective efficient solutions that can be identified at iterations 1,2,…, $|E\ |$ are $|E|$. $|E|$-1, …, 2, 1, respectively. Hence, the total number of such solutions is $\sum_{i=1}^{|E|} i = \frac{|E|(|E|+1)}{2}$.

We now present the extension of classical ε-constraint method, consider the following CWBOIP problem.

$$\text{Min } f_1(x) + \frac{1}{f_2^{GUB} - f_2^{GLB} + 1} f_2(x) + \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)} f_3(x)$$

$$\text{s.t.} \quad f_2(x) \le l_2$$

$$f_3(x) \le l_3$$

$$x \in X$$

Procedure 3.2. Generating all Efficient Solutions of a Tri-Objective Problem

(classical ε-constraint method)

Step 0. Find f_2^{GUB}, f_2^{GLB}, f_3^{GUB} and f_3^{GLB}.

Let $l_2 = f_2^{GUB}$

Step 1. Let $l_3 = f_3^{GUB}$

Step 2. Solve the CWBOIP problem with l_2, and l_3.

Step 3. If the solution is infeasible, go to Step 5.

Step 4. Let the optimal solution be x^*.

$E = E \cup (f_1(x^*), f_2(x^*), f_3(x^*))$

$l_3 = l_3 - 1$

Go to Step 2

Step 5. If $l_2 \geq f_2^{GLB}$, $l_2 = l_2 - 1$, go to Step 1.

Otherwise STOP.

The procedure iterates at most as the product of any two objective general ranges which is upper bounded by pseudo-polynomial value Max $\{(f_1^{GUB} - f_1^{GLB} + 1)(f_2^{GUB} - f_2^{GLB} + 1), (f_1^{GUB} - f_1^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1), (f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)\}$. Hence the complexity of the procedure depends on the complexity of the specific CWBOIP problem.

The number of times Procedure 3.1 iterates is bounded by the tighter objective efficiency ranges, whereas the same number for Procedure 3.2 is bounded by the general objective ranges.

$$\frac{\text{Max \# of iterations Procedure 3.1}}{\text{Max \# of iterations Procedure 3.2}} \sim \prod_{i=1}^{2}\left(\frac{f_i^{GUB} - f_i^{GLB}}{f_i^{TUB} - f_i^{GLB}}\right)$$

The difference between these two ranges enlarges with the increase in the size of the problem and the variability of the problem parameters. Moreover, l_2 and l_3 are decreased by unit decrements in Procedure 3.2, however their values are decreased to the related value of the next efficient solution in Procedure 3.1.

In Section 5, we provide a numerical example considering the Tri-Objective Assignment Problem that also compares the efficiency of these two methods.

4. *K*-Objective (Multi Objective) Integer Programming (MOIP) Problem

The Multi-Objective Integer Programming problem with *k*-objectives (MOIP) is defined as:

(MOIP) Min $f_1(x)$

Min $f_2(x)$

.

.

.

Min $f_k(x)$

s.t. $x \in X$

where X is the set of feasible solutions, where $x_j \geq 0$ and integer for all $j \in \{1, 2, ..., n\}$.

A solution $x' \in X$ is *k*-objective efficient if and only if there is no $x \in X$ such that $f_i(x) \leq f_i(x')$ for all $i \in \{1, ..., k\}$ and $f_i(x) < f_i(x')$ for at least one *i*.

The individual objectives are defined as $f_1(x) = \sum_{j=1}^{n} c^1{}_j x_j$, $f_2(x) = \sum_{j=1}^{n} c^2{}_j x_j$,..., $f_k(x) = \sum_{j=1}^{n} c^k{}_j x_j$ where $c^i{}_j$ is integer for all *i* and *j*.

We first present our method to solve MOIP problem, and then provide the classical ε-constraint method extension, and then compare the two algorithms in terms of their efficiency.

The Constrained Weighted (*K*-1)-Objective Integer Programming (CW(K-1)OIP) problem is defined as:

(CW(K-1)OIP) Min $f_1(x) + w_k f_k(x)$

Min $f_2(x) + w_k f_k(x)$

.

.

.

Min $f_{k-1}(x) + w_k f_k(x)$

s.t. $f_k(x) \leq l_k$

$x \in X$

Note that, we consider the k^{th} objective $f_k(x)$ together with $f_1(x)$, $f_2(x)$,…, $f_{k-1}(x)$ to ensure k-objective efficiency. Moreover we bound the $f_k(x)$ value via the constraint, $f_k(x) \le l_k$. The solution of the CW(K-1)OIP problem provides a set of (k-1)-objective efficient solutions, that are also k-objective efficient, for the properly set w_k and l_k values. All k-objective efficient solutions can be found by solving the CW(K-1)OIP problem for all possible w_k and l_k values. However, enumerating all possible values explicitly, may be very impractical, even infeasible, in many cases. Recognizing this fact, we find the right way to set w_k and l_k values.

To generate the k-objective efficient set, efficiently, we initially set l_k to the general upper bound on $f_k(x)$ value of all feasible solutions and find the corresponding (k-1)-objective efficient set. We then generate the other (k-1)-objective efficient solution sets by decreasing l_k systemically, towards the set of (k-1)-objective efficient solutions that provide a lower bound on $f_k(x)$ value.

Lemma 4.1 presents a property of an upper bound on the $f_i(x)$ value of all k-objective efficient solutions provided that $f_i(x)$ value is no more than l_i.

Lemma 4.1. A k-objective efficient solution that provides an upper bound on one objective $f_i(x)$, for a specified l_i , is (k-1)-objective efficient with respect to other k-1 objectives.

Proof. A k-objective efficient solution providing an upper bound on $f_i(x)$, satisfies $f_i(x') \ge f_i(x)$, for all $x' \in X$, hence is not non-dominated in objective i. The definition of efficiency follows that, an efficient solution should have at least one non-dominated objective by any other efficient solution. In order to have at least one non-dominated objective, an efficient solution not non-dominated in objective i, should be (k-1)-objective efficient in other k-1 objectives. ■

We next find a range for w_k value that leads to an efficient solution when used in the CW(K-1)OIP problem. In doing so, we define f_k^{GUB} and f_k^{GLB} as the respective general upper and lower bounds on $f_k(x)$ value of any feasible solution. The general bounds can be obtained by solving the individual single objective problems.

Using these bounds, Lemma 4.2 specifies a range for w_k that guarantees the generation of k-objective efficient solutions for a specified l_k value.

Lemma 4.2. The solution of CW(K-1)OIP problem (by Procedure 4.1) with $w_k = \dfrac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)\dots(f_k^{GUB} - f_k^{GLB} + 1)}$ provides an upper bound on the $f_k(x)$ value of all k-efficient solutions for a specified l_k value.

Proof. The CW(K-1)OIP problem solved within Procedure 4.1 can be stated as follows:

$$\text{Min } f_1(x) + \frac{1}{f_2^{GUB} - f_2^{GLB} + 1} f_2(x) + \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)} f_3(x) + \dots$$

$$+ \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)\dots(f_{k-1}^{GUB} - f_{k-1}^{GLB} + 1)} f_{k-1}(x) + w_k f_k(x)$$

$$\begin{aligned} \text{s.t.} \quad & f_2(x) \le l_2 \\ & f_3(x) \le l_3 \\ & . \\ & . \\ & f_k(x) \le l_k \\ & x \in X \end{aligned}$$

From Lemma 4.1 we know that the k-objective efficient solutions having the highest $f_k(x)$ value should be (k-1)-objective efficient for other k-1 objectives. To find an upper bound on $f_k(x)$ values of k-objective efficient solutions, one can generate the (k-1)-objective efficient solutions with respect to objectives $f_1(x)$, $f_2(x)$, . . ., $f_{k-1}(x)$, thus guarantee (k-1)-objective efficiency and minimize $f_k(x)$ over the (k-1)-objective efficient set. In place of this two-step hierarchical procedure, we solve a single CW(K-1)OIP problem and generate the (k-1)-objective efficient solutions which are also k-objective efficient.

The CW(K-1)OIP problem provides a (k-1)-objective efficient set having the highest $f_k(x)$ value, if w_k is defined such that the largest decrease in $f_k(x)$, will not contribute to the objective function as much as the smallest increase in either $f_1(x)$ or $f_2(x)$or ... or $f_{k-1}(x)$. Since $f_1(x)$, $f_2(x)$, . . ., $f_{k-1}(x)$ are hierarchically optimized in that order, the contribution of $f_{k-1}(x)$ is always smaller than those of $f_1(x)$, $f_2(x)$, . . ., $f_{k-2}(x)$, so one can only consider the contribution by $f_{k-1}(x)$ and ignore the other contributions. The largest decrease in $f_k(x)$ is no more than ($f_k^{GUB} - f_k^{GLB}$), and its contribution to the objective function is no more than $w_k(f_k^{GUB} – f_k^{GLB})$. The smallest increase in $f_{k-1}(x)$ is no less than $\frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)...(f_{k-1}^{GUB} - f_{k-1}^{GLB} + 1)}$, due to the integrality of the parameters and decision variables. This follows, when $\frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)...(f_{k-1}^{GUB} - f_{k-1}^{GLB} + 1)} > w_k(f_k^{GUB} - f_k^{GLB})$, equivalently $w_k < \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)...(f_{k-1}^{GUB} - f_{k-1}^{GLB} + 1)(f_k^{GUB} - f_k^{GLB})}$, the ($k$-1)-objective efficient solutions are also k-objective efficient for a specified l_k, are found. As $\frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)...(f_k^{GUB} - f_k^{GLB} + 1)} <$ $\frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)...(f_{k-1}^{GUB} - f_{k-1}^{GLB} + 1)(f_k^{GUB} - f_k^{GLB})}$, the CW(K-1)OIP problem with w_k=$\frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)...(f_k^{GUB} - f_k^{GLB} + 1)}$ provides an upper bound on the $f_k(x)$ values of the k -objective efficient solutions for a specified l_k value. ■

Through Lemma 4.3 we state that for a properly selected w_k value, the following Weighted (K-1)-Objective Integer Programming (W(K-1)OIP) problem gives an upper bound on $f_k(x)$ values of all k-objective efficient solutions.

(W(K-1)OIP) Min $f_1(x) + w_k f_k(x)$

Min $f_2(x) + w_k f_k(x)$

.

.

.

Min $f_{k-1}(x) + w_k f_k(x)$

s.t. $x \in X$

Lemma 4.3. The optimal solution of the W(K-1)OIP problem (by Procedure 4.1) with w_k= $\frac{1}{(f_2^{GUB}-f_2^{GLB}+1)(f_3^{GUB}-f_3^{GLB}+1)...(f_k^{GUB}-f_k^{GLB}+1)}$ provides an upper bound on the $f_k(x)$ value of all *k*-efficient solutions.

Proof. Lemma 4.2 states that the optimal solution of the CW(K-1)OIP problem (by Procedure 4.1) with w_k= $\frac{1}{(f_2^{GUB}-f_2^{GLB}+1)(f_3^{GUB}-f_3^{GLB}+1)...(f_k^{GUB}-f_k^{GLB}+1)}$ provides an upper bound on the $f_k(x)$ value of all *k*-efficient solutions for a specified l_k value. Once we set l_k to a very big number no less than f_k^{GUB}, the CW(K-1)OIP problem does not skip any *k*-objective efficient solution and the optimal solution provides an upper bound on the $f_k(x)$ value of all *k*-objective efficient solutions. Moreover the CW(K-1)OIP problem reduces to the W(K-1)OIP problem, as the constraint $f_k(x) \le l_k$ becomes redundant. ■

We now state the stepwise description of our algorithm to generate all *k*-objective efficient solutions. The procedure uses the solution of CW(K-1)OIP problem using the same procedure with the first (*k-1*)-objective in each iteration.

Procedure 4.1 Generating all *K*-Objective Efficient Solutions

Step 0. Find the general bounds f_2^{GUB}, f_2^{GLB}, f_3^{GUB}, f_3^{GLB},…, f_k^{GUB} , f_k^{GLB}

Let w_k= $\frac{1}{(f_2^{GUB}-f_2^{GLB}+1)(f_3^{GUB}-f_3^{GLB}+1)...(f_k^{GUB}-f_k^{GLB}+1)}$

Let l_k=f_k^{GUB}

Step 1. Solve the CW(K-1)OIP problem with l_k using this procedure with the first *k*-1 objectives.

If the solution is infeasible, STOP.

Step 2. Let the (*k*-1)-objective solutions set be ME^*.

$E = E \cup ME^*$

l_k= $\max\{f_k(x), x \in ME^*\}$ - 1

Go to Step 1

Set *E* returned by the above procedure resides all *k*-objective efficient solutions. Lemma 4.4 states this result formally.

Lemma 4.4. Procedure 4.1 generates all *k*-objective efficient solutions.

Proof. In Step 0, no feasible, hence efficient, solution is eliminated as l_k is set to the general upper bound value of $f_k(x)$. From Lemma 4.2 we know that, Step 1 provides an efficient solution whose f_k value, $\max\{f_k(x), x \in ME^*\}$, is an upper bound on the $f_k(x)$ values of all efficient solutions for a specified l_k value, by identifying the (*k*-1)-objective efficient set. As $\max\{f_k(x), x \in ME^*\}$ is an upper bound, there cannot exist any *k*-objective efficient solution whose $f_k(x)$ value is between $\max\{f_k(x), x \in ME^*\} + 1$ and l_k. Due to the integrality of the parameters and decision variables, the next $f_k(x)$ value cannot be higher than $\max\{f_k(x), x \in ME^*\}$-1. So setting l_k to $\max\{f_k(x), x \in ME^*\}$-1 does not eliminate any *k*-objective efficient solution, and identifies the next one if there exists any. Each iteration identifies a (*k*-1)-objective efficient solution set by solving the CW(K-1)OIP problem with an updated l_k value. The procedure terminates when the associated CW(K-1)OIP problem returns an infeasible solution. An infeasible solution implies that there is no *k*-objective efficient solution having $f_k(x)$ value smaller than or equal to the specified l_k value, thus all *k*-objective efficient solutions are generated. ■

Procedure 4.1 starts with $l_k = f_k^{GUB}$ and provides an upper bound on the $f_k(x)$ values of the *k*-objective efficient solutions, i.e., f_k^{kUB}. The last feasible solution returned by the algorithm gives a lower bound on the $f_k(x)$ value, i.e., f_k^{GLB}. Hence the $f_k(x)$ values of all *k*-objective efficient solutions are between f_k^{GLB} and f_k^{kUB}.

The procedure iterates at most as the product of any *k*-1 objective ranges which is upper bounded by pseudo-polynomial value Max $\{g_1 . g_2 \ldots., g_k\}$ where $g_i = [(f_1^{kUB} - f_1^{GLB} +1)\ (f_2^{kUB} - f_2^{GLB} +1) \ldots (f_k^{kUB} - f_k^{GLB} +1)\ ]/\ (f_i^{kUB} - f_i^{GLB} +1)$. Hence the complexity of the procedure depends on the complexity of the specific CW(K-1)OIP problem.

An upper bound on number of iterations can also be expressed by the number of *k*-objective efficient solutions, |*E*|. In procedure 4.1, each update of l_i eliminates one *k*-objective efficient solution from the search space. The sum of maximum number of (*k*-1)-objective efficient solutions that can identified by solving C(K-1)BOIP problem is, $\sum_{i_{k-2}=1}^{|E|} \cdots \sum_{i_2=1}^{i_3} \sum_{i_1=1}^{i_2} i_1 = \frac{|E|(|E|+1)\ldots(|E|+k-2)}{2.3\ldots(k-1)}$.

We now present the extension of classical ε-constraint method, consider the following CW(K-1)OIP problem.

$$\text{Min } f_1(x) + \frac{1}{f_2^{GUB} - f_2^{GLB} + 1} f_2(x) + \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)} f_3(x) + ...$$

$$... + \frac{1}{(f_2^{GUB} - f_2^{GLB} + 1)(f_3^{GUB} - f_3^{GLB} + 1)...(f_k^{GUB} - f_k^{GLB} + 1)} f_k(x)$$

$$\begin{aligned} \text{s.t.} \quad & f_2(x) \le l_2 \\ & f_3(x) \le l_3 \\ & . \\ & . \\ & f_k(x) \le l_k \\ & x \in X \end{aligned}$$

Procedure 4.2. Generating all Efficient Solutions of a Multi-Objective Problem

(classical ε-constraint method)

Step 0. Find f_2^{GUB}, f_2^{GLB}, ..., f_k^{GUB} and f_k^{GLB}.

Let l_2=f_2^{GUB}

Step 1.1. Let l_3=f_3^{GUB}

Step 1.2. Let l_4=f_4^{GUB}

…

Step 1.*k*-2. Let l_k=f_k^{GUB}

Step 2. Solve the CW(K-1)BOIP problem with l_2, l_3, ..., and l_k.

Step 3. If the solution is infeasible, go to Step 5.

Step 4. Let the optimal solution be x^*.

$E = E \cup (f_1(x^*), f_2(x^*), ..., f_k(x^*))$

$l_k = l_k - 1$

Go to Step 2

Step 5.1. If $l_{k-1} \geq f_{k-1}^{GLB}$, $l_{k-1} = l_{k-1}$ -1, go to Step 1.*k*-2.

Step 5.2. If $l_{k-2} \geq f_{k-2}^{GLB}$, $l_{k-2} = l_{k-2}$ -1, go to Step 1.*k*-1.

…

Step 5.*k*-1. If $l_3 \geq f_3^{GLB}$, $l_3 = l_3$ -1, go to Step 1.2.

Step 5.*k*-2. If $l_2 \geq f_2^{GLB}$, $l_2 = l_2$ -1, go to Step 1.1.

Otherwise STOP.

The procedure iterates at most as the product of any *k*-1 objective general ranges which is upper bounded by pseudo-polynomial value Max $\{g_1 . g_2 ...., g_k\}$ where $g_i = [(f_1^{GUB} - f_1^{GLB} +1) (f_2^{GUB} - f_2^{GLB} +1) \ldots (f_k^{GUB} - f_k^{GLB} +1)] / (f_i^{GUB} - f_i^{GLB} +1)$. Hence the complexity of the procedure depends on the complexity of the specific CW(K-1)OIP problem.

The number of times Procedure 4.1 iterates is bounded by the tighter objective efficiency ranges, whereas the same number for Procedure 4.2 is bounded by the general objective ranges.

$$\frac{\text{Max \# of iterations Procedure 4.1}}{\text{Max \# of iterations Procedure 4.2}} \sim \prod_{i=1}^{k-1} \left(\frac{f_i^{GUB} - f_i^{GLB}}{f_i^{kUB} - f_i^{GLB}} \right)$$

The difference between two ranges increases with an increase in the size of the problem and the variability of the problem parameters. In Procedure 4.2, l_2, l_3, ... l_k decrease by one unit, however, in Procedure 4.1, these values decrease by the value of the next efficient solution.

In the next section, we provide a numerical example to demonstrate the efficiency of our procedure in comparison with the classical ε-constraint method.

5. An Example Problem

In this section we compare the performance of our procedure with the extension of classical ε-constraint method, on a Tri-Objective Assignment Problem (TAP) instance. We consider a 5 by 5 problem instance and generate the objective function coefficients randomly from a discrete uniform distribution between 1 and 100. Table 5.1 has the three objective coefficients for assigning each row to each column. We represent each solution by a sequence of column index values assigned to rows 1 through 5. Accordingly in sequence 5-4-3-2-1, row 1 is assigned to column 5 and row 2 is assigned to column 5.

Table 5.1 Three objective coefficients for the example problem instance

c^1	1	2	3	4	5
1	99	19	74	55	41
2	23	81	93	39	49
3	66	21	63	24	38
4	65	41	7	39	66
5	93	30	5	4	13

c^2	1	2	3	4	5
1	28	39	19	42	7
2	66	98	49	83	42
3	73	26	42	13	54
4	46	42	28	27	99
5	80	17	99	59	68

c^3	1	2	3	4	5
1	29	67	2	90	7
2	84	37	64	64	87
3	54	11	100	83	61
4	75	63	69	96	3
5	66	99	34	33	21

Using the single objective assignment solutions one can identify general upper and lower bounds on individual objectives as;

f_1^{GLB}= 86, 2-1-4-3-5

f_2^{GLB}= 128, 1-5-4-3-2

f_3^{GLB}= 129, 3-2-1-5-4

f_1^{GUB}=358, 4-2-3-5-1

f_2^{GUB}=411, 4-2-1-5-3

f_3^{GUB}=451, 4-5-3-1-2

We now give the iteration details of Procedure 3.1. We report the number of IPs solved, the l_2 and l_3 bound values. The objective function values of the bi-objective solutions are staed in groups each representing a single execution of Step 1.

	$l_3 \le$	451		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
1	86	214	324	411
2	96	186	204	213
3	125	131	342	185
4	209	128	367	130
5	Infeasible			127
	Max($f_3(x)$)		367	

	$l_3 \le$	366		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
6	86	214	324	411
7	96	186	204	213
8	125	131	342	185
9	Infeasible			130
	Max($f_3(x)$)		342	

	$l_3 \le$	341		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
10	86	214	324	411
11	96	186	204	213
12	180	183	229	185
13	253	132	328	182
14	Infeasible			131
	Max($f_3(x)$)		328	

	$l_3 \le$	327		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
15	86	214	324	411
16	96	186	204	213
17	180	183	229	185
18	269	173	320	182
19	Infeasible			172
	Max($f_3(x)$)		324	

	$l_3 \le$	323		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
20	91	246	314	411
21	96	186	204	245
22	180	183	229	185
23	269	173	320	182
24	Infeasible			172
	Max($f_3(x)$)		320	

	$l_3 \le$	319		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
25	91	246	314	411
26	96	186	204	245
27	180	183	229	185
28	Infeasible			182
	Max($f_3(x)$)		314	

	$l_3 \le$	313		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
29	96	186	204	411
30	180	183	229	185
31	Infeasible			182
	Max($f_3(x)$)		229	

	$l_3 \le$	228		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
32	96	186	204	411
33	Infeasible			185
	Max($f_3(x)$)		204	

	$l_3 \le$	204		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
34	171	261	191	411
35	179	233	194	260
36	224	187	190	232
37	Infeasible			186
	Max($f_3(x)$)		194	

	$l_3 \le$	193		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
38	171	261	191	411
39	212	242	173	260
40	224	187	190	241
41	Infeasible			186
	Max($f_3(x)$)		191	

	$l_3 \le$	190		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
42	188	269	133	411
43	212	242	173	268
44	224	187	190	241
45	Infeasible			186
	Max($f_3(x)$)		190	

	$l_3 \le$	189		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
46	188	269	133	411
47	212	242	173	268
48	Infeasible			241
	Max($f_3(x)$)		173	

	$l_3 \le$	172		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
49	188	269	133	411
50	283	261	140	268
51	Infeasible			260
	Max($f_3(x)$)		140	

	$l_3 \le$	139		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
52	188	269	133	411
53	Infeasible			268
	Max($f_3(x)$)		133	

	$l_3 \le$	132		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
54	291	348	129	411
55	Infeasible			347
	Max($f_3(x)$)		129	

	$l_3 \le$	128		$l_2 \le$
#	$f_1(x)$	$f_2(x)$	$f_3(x)$	
56	Infeasible			411

Table 5.2, below, lists the efficient solutions.

Table 5.2 The efficient solutions for the example problem

$f_1(x)$	$f_2(x)$	$f_3(x)$	r_1	r_2	r_3	r_4	r_5
86	214	324	2	1	4	3	5
91	246	314	2	1	5	3	4
96	186	204	5	1	2	3	4
125	131	342	5	1	4	3	2
171	261	191	5	4	2	1	3
179	233	194	1	4	2	3	5
180	183	229	1	5	2	3	4
188	269	133	3	1	2	5	4
209	128	367	1	5	4	3	2
212	242	173	3	4	2	1	5
224	187	190	5	3	2	1	4
253	132	328	5	3	4	1	2
269	173	320	5	3	1	4	2
283	261	140	1	3	2	5	4
291	348	129	3	2	1	5	4

A total of 56 IPs are solved to identify 15 tri-objective efficient solutions. If Procedure 3.2, an extension of classical ε-constraint method, was used then Min $\{(f_1^{GUB} - f_1^{GLB} +1)\ (f_2^{GUB} - f_2^{GLB} +1),\ (f_1^{GUB} - f_1^{GLB} +1)(\ f_3^{GUB} - f_3^{GLB} +1),\ (f_2^{GUB} - f_2^{GLB} +1)(\ f_3^{GUB} - f_3^{GLB} +1)\}$= 77532 IP problems would be solved. Note that for this example problem, our proposed method provides 1400 folds improvement over the classical method, in the number of IPs solved. For larger problem instances with wider feasibility ranges, the improvements can be much more significant.

6. Conclusions

In this study we develop a method to identify individual objective efficiency ranges for multi objective problems. Our method uses simpler (having less objectives) multi-objective problems to obtain efficiency ranges of the individual objectives. By modifying and generalizing this range identification method to the constrained problems we develop a method to generate all efficient solutions for tri-objective and general multi-objective integer programming problems.

We also present the classical ε-constraint method for the bi-objective integer programming (BOIP) problem, and provide its extensions for the tri-objective (TOIP) and multi-objective (MOIP) integer programming problems.

We compare the theoretical worst case performances of two algorithms based on the number of IPs solved for generating the efficient set. We demonstrate our method on a small size tri-objective assignment problem, and observe significant improvement over the classical ε-constraint method in the number of IPs solved. The amount of improvement mainly depends on the difference between feasibility and efficiency ranges, and the number of the efficient solutions.

We hope our study helps to stimulate future work on the Multi-Objective Integer Programming area. Future work may include the applications to some practical problems like supply chain design, scheduling, logistics and location. Our general results when applied to these special problems may trigger the development of some specialized procedures. Another promising research area may include the development of the procedures for the optimal solution of the pre-specified function of the *k*-objectives.

References

Ehrgott. M., X. Gandibleux. 2000. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spec.* **22** 425-460.

Ehrgott, M. X. Gandibleux (Eds.). 2002. *Multiple Criteria Optimization State of the Art Annotated Bibliographic Surveys: International Series in Operations Research & Management Science*, Vol. 52. Springer.

Ehrgott. M., X. Gandibleux. 2004. Approximative solution methods for multiobjective combinatorial optimization. *TOP* **12** 1-63

Ehrgott, M. 2006. A discussion of scalarization techniques for multiple objective integer programming. *Ann. Oper. Res.* **147** 343-360.

Klamroth, K., J. Tind, S. Zust. 2004. Integer Programming Duality in Multiple Objective Programming. *J. Glob. Opt.* **29** 1-18.

Klein, D., E. Hannan. 1982. An algorithm for the multiple objective integer linear programming problem. *Eur. J. Oper. Res*. **9** 378-385.

Sylva J., A. Crema. 2004. A method for finding the set of nondominated vectors for multiple objective integer linear programs. *Eur. J. Oper. Res*. **158** 46-55.

Sylva J., A. Crema. 2007. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *Eur. J. Oper. Res*. **180** 1011-1027.