Finding all non-dominated points for a bi-objective generalized assignment problem

Ali Al-Hasani1,3, Masar Al-Rabeeah1,3, Santosh Kumar1,2, Andrew Eberhard1

1 School of Mathematical and Geospatial Sciences, RMIT University, Melbourne, Australia.
2 Department of Mathematics and Statistics, University of Melbourne, Melbourne, Australia.
3 Department of Mathematics, College of Sciences, University of Basrah, Al- Basrah, Iraq.

E-mail: santosh.kumarau@gmail.com
E-mail: ali.alhasani@rmit.edu.au

Abstract. In this paper, we consider a bi-objective generalized assignment problem (BOGAP) and find all non-dominated points using two methods; Two Phases Method (TPM) and the Balanced Box Method (BBM). The computational results are shown by testing a number of instances, which indicates that the TPM performs better compared to the BBM.

Keywords: Balanced Box Method; Bi-objective generalized assignment problem; Non-dominated points; Two Phases method.

1. Introduction

A single objective generalized assignment problem (GAP) is an integer program, which is a NP-hard problem [1]. These problems require an efficient solution method for real-life applications. A single-objective GAP has been extensively discussed, for example, Terry and Richard [2] developed a branch and bound algorithm, Martello and Toth [3] have provided an enumerative approach, Cattrysse and Wassenhove [4] have presented a survey of algorithms that were developed for solving the GAP. This survey presents a summary of all mathematical programming techniques that addressed the solution to the GAP before 1990. Shmoys and Tardos [5] have presented an approximation algorithm for the GAP, which has been classified as a polynomial-time approximation that minimizes a weighted sum of cost and make-span time. A survey covering latest three decades of developments on GAP have been presented by ncan [6]. Munapo et al. [7] have introduced a new technique to solve GAP by using a transportation branch and bound algorithm. Sethanan and Pitakaso [8] have improved differential evolution algorithms to solve the GAP.

In this paper, we consider a bi-objective generalized assignment problem (BOGAP) and find all non-dominated points. There are many approaches to find non-dominated point-set for multiobjective integer programming (MOIP) model, which have been reviewed in [9, 10]. Here, our attempt is to find all non-dominated points by an exact method, which does not miss out any non-dominated point in the criterion space.

A literature review of BOGAP indicated that two studies were carried out, the first one was by Zhang and Ong [11], they used a LP-based heuristic and generated approximate non-
dominated points. The second study was carried out by Prakash et al. [12], where one objective was linear and the other one was a non-linear function.

This paper finds all non-dominated points by two exact approaches, i.e., Two Phases method (TPM) [13], and the Balanced box method (BBM) [14]. Both approaches have been implemented on instances of varying size problems. We compare the two methods in terms of CPU time, number of non-dominated points and the number of IP’s. These random test instances for BOGAP are discussed in Section 4.

The mathematical formulation for a Bi-objective combinatorial optimization problem can be as follows:

\[
\text{Max } Z(x) = \left( Z_1(x), Z_2(x) \right)
\]

Subject to

\[ Ax \leq b, \]

where \( Z_1(x) \) and \( Z_2(x) \) are two objective functions and \( x \) are binary variables from the decision space \( X \), and \( A \) is a matrix coefficients of constraints, and \( b \) is non-negative integer.

**Definition 1** A feasible solution \( x' \in X \) is called efficient or (Pareto) solution, if there is no other \( x \in X \) such that \( Z(x') \geq Z(x) \). The set of all efficient solutions \( x' \in X \) denoted \( X_E \) and is called the efficient solution set.

**Definition 2** if \( x' \) is efficient solution, \( Z(x') \) is called non-dominated (Pareto) point (NDP), and the set of all non-dominated points \( y' \in Y \) is denoted \( Y_{ND} \) and is called the non-dominated points set.

**Definition 3** The efficient solution \( x \) is called a supported efficient solution for problem (1), if there is some \( \lambda \in \mathbb{R}^2 \) satisfies an optimal solution of the following single objective weighted-sum problem:

\[
\text{Max } x \in X \left( \lambda_1 Z_1(x) + \lambda_2 Z_2(x) \right)
\]

and the \( Z(x) \) is called supported nondominated point. The set of all supported efficient solution and supported nondominated sets are denoted \( X_{SE} \) and \( Y_{SN} \), respectively. Otherwise any efficient solution \( x \) does not satisfy (2) is called unsupported efficient solution and \( Z(x) \) is called unsupported nondominated point. The set of all unsupported efficient solution and unsupported nondominated sets are denoted \( X_{USE} \) and \( Y_{USN} \), respectively.

**Definition 4** The Point \( Z^{Id} \) is called Ideal Point if one can achieve the following condition:

\[
Z^{Id}_k := \max_{x \in X} Z_k(x) \quad \forall k = 1, 2.
\]

**Definition 5** The Point \( Z^{Na} \) is called Nadir Point if one can achieve the following condition:

\[
Z^{Na}_k := \min_{x \in X} Z_k(x) \quad \forall k = 1, 2.
\]

For further details see [15, 16]. The Figure 1 illustrates supported non-dominated points, unsupported non-dominated points, ideal point and nadir point.

This paper has been organized into 5 sections. In Section 2 a brief review of the Two Phases method and the Balanced Box Method is provided. Determination all non-dominated points for BOGAP by both methods using an illustrative example is given in Section 3. The computational results by Two Phases method and Balanced Box Method have been compared with respect to CPU time, which is presented in the Section 4. Finally, a few concluding remarks are given in Section 5.
2. A brief review of ‘Two Phases Method’ and ‘Balanced Box Method’

Both TPM and BBM are effective approaches to find all non-dominated points in criterion space. There is a similarity between the two methods in terms of style, since both TPM and BBM divide the solution region in the criterion space into triangles and rectangles, respectively. We have chosen these two methods because they can be applied to a Bi-objective integer programming problem, both methods search for the non-domination points in the criterion space, as well as both methods do not depend on imposing a small value in the procedures steps to find other non-domination points.

2.1. Two Phases Method

2.1.1. Phase one

In phase one, we find all supported non-dominated points. There are many methods to compute \( Y_{SN} \), however, in this paper, the method by Aneja and Nair \[17\] has been used. The following are the steps in phase one:

(i) Find the upper bond point \( Z^1 = (z^1_1, z^1_2) \) and the lower bound point \( Z^2 = (z^2_1, z^2_2) \) by applying the lexicographic method.

(ii) Apply Weighted sum method \[17\] to compute the supported non-dominated points as follows:

(a) \( \lambda_1 = |z^1_1 - z^2_1|, \lambda_2 = |z^1_2 - z^2_2| \)

(b) Solve integer problem \( Z_{\lambda} = \lambda_1 Z_1 + \lambda_2 Z_2 \), substitute \( x_{\lambda} \) on \( Z_1 \) and \( Z_2 \) we get \( (z^3_1, z^3_2) \).

(c) Test \( (z^3_1, z^3_2) \) whether a non-dominated point or not, if yes add on to the supported non-dominated points set then go to Step 2, if not ignored that point and stop the search.

Figure 2 illustrates finding six supported non-dominated points \( \{Z_1, Z_3, Z_4, Z_5, Z_6, Z_2\} \), which were calculated in the first phase for general BOGAP.

2.1.2. Phases two

This Phase depends on the supported non-dominated points that were calculated in the first phase. Triangles are drawn between each adjacent supported non-dominated points and then each triangle is examined for unsupported non-dominated points. The search in each triangle is carried out by using the approach by Ulungu and Teghem \[13\] and Przybylski, Gandibleux and Ehrgott \[18\]. The steps for the second phase are:

(i) Identify each adjacent supported NDP points \( \{(z_1, z_3), (z_3, z_4), (z_4, z_5), (z_5, z_6), (z_6, z_2)\} \)

and draw a triangle between each adjacent supported NDP points such that the hypotenuse is between those two consecutive points, see Figure 4. Let the first triangle be \( (z_1, z^*, z_3) \).
Here, $z^*$ is local nadir point of two points $(z_1, z_3)$. We repeat this process for all SNDPs, then obtain the set of triangles TS. The bi-objective maximization problem, ideal point and nadir point defined as follows: ideal

$$z_{\text{ideal}} = \left( \max_{i=(1,n)} (z_i)(x), \max_{i=(1,n)} (z_i)(y) \right)$$

$$z_{\text{nadir}} = \left( \min_{i=(1,n)} (z_i)(x), \min_{i=(1,n)} (z_i)(y) \right)$$

(ii) Consider the first triangle $T_1 (z_1, z^*, z_3)$. Inside the triangle $T_1$, we solve the weighted sum objective minimization problem.

(a) If the solution exist, let us denote the resulting point by $z_r$, see Figure 3. Note that $z_r$ is value the weighted sum objective function.

Draw a line parallel to $z_1$ and $z_3$ and passing through $z_r$ is intersected with each horizontal and vertical line, i.e. $(z_{t1}, z_{t2})$. Then, by using mathematical formula of line equation, the coordinate of these points are calculated as following:

$$z_{t1}(x) = z_r(x) - \frac{z_{21} - z_{11}}{z_{12} - z_{22}} \times (z_{12} - z_r(y))$$

$$z_{t1}(y) = z_{12}$$

$$z_{t2}(y) = z_{21}$$

$$z_{t2}(x) = z_r(y) - \frac{z_{21} - z_{11}}{z_{12} - z_{22}} \times (z_{21} - z_r(x))$$
(b) Construct two new triangles with \( (z_{t1}, z_r, [z^*_t]) \) and \( (z_{t2}, z_r, z^*_t) \) and insert into the set \( TS \) and old triangle \( T \) is removed from \( TS \).

c) if there is no solution, chose the next triangle.

(iii) Repeat the above process for each triangle until the set of triangles \( TS \) is empty.

The Figure 4 illustrates determination of unsupported non-dominated points in phase two by drawing triangles between each adjacent supported non-dominated points. TPM solves one integer program (IP) to find one non-dominated points, as well as it solves one IP to check whether the solution exist above the line (obtain in first phase) or inside tingles (second phase). As result, TPM solves \( 2n \) number of IP’s to find \( n \) non-dominated. Therefore, TPM is one of the best algorithm to solve bi-objective integer programming problems.

![Figure 4: The unsupported non-dominated points from phase two calculations](image)

The steps of phase 1 and 2 are explained in algorithms [1] and [2].

### 2.2. Balanced Box Method

Balanced Box Method was developed by Boland, Charkhgard and Savelsbergh [14], which is a recent criterion search method that deals with rectangles in search of the non-dominated points. First, the method explores the main rectangle (the large one) that includes all non-dominated points. The search starts by solving two lexicographic problems,

\[
\begin{align*}
Z^1 &= \text{lex max}_{x \in Z} (z_1(x), z_2(x)) \\
Z^2 &= \text{lex max}_{x \in Z} (z_2(x), z_1(x))
\end{align*}
\]

which resulted in two points \( Z^1 \) and \( Z^2 \) giving the first rectangle \( R(Z^1, Z^2) \), as shown in Figure 5.

The algorithm maintains a queue of rectangles, so in the first iteration, it picks up the first rectangle and decomposes it horizontally into two smaller rectangles along \( Z_2(x) \) axis. These two rectangles will be added to the queue, the algorithm searches in the bottom rectangle first and then in the above one for unknown non-dominated points which will determine the new smaller rectangles. This process is continued until all non-dominated points have been determined then the search will terminate. The number of IPs that need to be solve by BBM is three times the number of non-dominated points. For example, if a problem has 30 non-dominated solutions,
Algorithm 1: Main step of TPM

\[
\text{max } z(x) = \{ z_1(x), z_2(x) \}
\]

Step 0. Initialize

\[
\text{sndp} = \emptyset \quad \text{usndp} = \emptyset \quad \text{tri} = \emptyset
\]

Step 1. First Phase();

Step 1.1. \( z_{LR}, z_{UL} \leftarrow \text{getBoundPoints}() \);

Step 1.1.1. \( z_{LR} \leftarrow \text{solveLexiOpt}(z_2, z_1) \);

Step 1.1.2. \( z_{UL} \leftarrow \text{solveLexiOpt}(z_1, z_2) \);

Step 1.2. \( \text{sndp} \rightarrow \text{add}(z_{LR}) \);

\( \text{sndp} \rightarrow \text{add}(z_{UL}) \);

Step 1.3. \( z_1 = z_{LR}; \quad z_2 = z_{UL}; \)

Step 1.4. While \( z_1 \neq z_2 \)

Step 1.4.1. \( z_{tmp} = \text{sndp} \rightarrow \text{next}(z_1) \);

Step 1.4.2. \( z_{new} \leftarrow \text{findSNDP}(z_1, z_{tmp}) \);

Step 1.4.3. \( \text{sndp} \rightarrow \text{add}(z_{new}) \);

Step 1.4.4. \( z_1 \leftarrow z_{new}; \)

Step 2. Second Phase();

Step 2.1. \( z_1 = z_{LR}; \quad z_2 = z_{UL}; \)

Step 2.2. While \( z_1 \neq z_2 \)

Step 2.2.1. \( z_{tmp} = \text{sndp} \rightarrow \text{next}(z_1) \);

Step 2.2.2. \( \text{tri} \rightarrow \text{push}((z_1, z_{tmp})) \);

Step 2.2.3. While \( \text{tri} \neq \emptyset \)

\( t = \text{tri} \rightarrow \text{pop}() \);

\( \text{usndp} = \text{insideTriangle}(t) \);

\( \text{usndp} \rightarrow \text{add}(z_{usndp}) \);

Step 2.2.4. \( z_1 \leftarrow z_{new}; \)

return \( \text{sndp}, \text{usndp} \);

Algorithm 2: Finding supported non-dominated points

input: \( z_1, z_2 \)

Step 1. \( \lambda_1 = z_1(y) - z_2(y); \quad \lambda_2 = z_2(x) - z_1(x); \)

Step 2. \( \text{val} = \lambda_1 * z_1(x) + \lambda_2 * z_1(y); \)

Step 3. \( \text{obj}_\text{fun} = \lambda_1 * z_1 + \lambda_2 * z_2; \)

Step 4. \( \text{result} \leftarrow \text{solveOpt(obj}_\text{fun}); \)

Step 5. if \( (\lambda_1 * \text{result}(x) + \lambda_2 * \text{result}(y) > \text{val}) \)

\( \text{sndp} \rightarrow \text{add}(\text{result}); \)

return;

BBM will solve 90 IPs to find these non-dominated points and that can be time consuming. Another aspect is that BBM need to solve two problems for any empty rectangle instead of one and that may be another disadvantage.
**Algorithm 3:** Finding unsupported non-dominated points inside triangle

Step 0. Initialize
\[ z_{11} = \text{tri} \rightarrow z_1(x); \]
\[ z_{12} = \text{tri} \rightarrow z_1(y); \]
\[ z_{21} = \text{tri} \rightarrow z_2(x); \]
\[ z_{22} = \text{tri} \rightarrow z_2(y); \]

Step 1. Add constraint(obj1,z_{21});
add constraint(obj2,z_{12});

Step 2. \( \text{result} \leftarrow \text{solveOpt(obj.fun)}; \)

Step 3. \( z_{\text{tmp}}(x) = \text{floor}(\text{result}(x) - \frac{z_{21} - z_{11}}{z_{12} - z_{22}} \times (z_{12} - \text{result}(y))); \)
\[ z_{\text{tmp}}(y) = z_{12}; \]
\[ \text{tri} \rightarrow \text{push}((z_{\text{tmp}}, \text{result})); \]

Step 4. \( z_{\text{tmp}}(x) = z_{21}; \)
\[ z_{\text{tmp}}(y) = \text{floor}(\text{result}(y) - \frac{z_{21} - z_{11}}{z_{12} - z_{22}} \times (z_{21} - \text{result}(x))); \]
\[ \text{tri} \rightarrow \text{push}((z_{\text{tmp}}, \text{result})); \]

Step 5. \( \text{usndp} \rightarrow \text{add(result)}; \)
return;

![Figure 5: The first rectangle for BBM](image)

3. Finding All Non-dominated points for BOGAP by TPM and BBM

The bi-objective generalized assignment problem is a case between bi-objective Knapsack (BOK) and bi-objective assignment (BOA). A few papers have discussed BOK and BOA problems, see [18], [19]. The BOGAP can be described as follows:

\[
\text{Max } Z^r(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} P_{ij}^r x_{ij} \quad r = 1, 2
\]
Subject to

\[ \sum_{j=1}^{n} W_{i,j} x_{i,j} \leq b_i \quad i = 1, \ldots, m. \]

\[ \sum_{i=1}^{m} x_{i,j} = 1 \quad j = 1, \ldots, n. \]

\[ x_{i,j} = 0 \text{ or } 1; \forall i = 1, \ldots, m \text{ and } j = 1, \ldots, n. \]

Where \( P_{i,j}^r \) are coefficients of the variables \( x_{i,j} \) in objective functions, \( r = 1, 2 \). Similarly \( W_{i,j} \) represent amount of resource consumed corresponding to the variables \( x_{i,j} \). The value \( b_i \) represents the total resource available for the processor \( i \).

3.1. Illustrative Example

Consider an instance BOGAP of size \( 7 \times 8 \).

Max \( Z^r(x) = \sum_{i=1}^{7} \sum_{j=1}^{8} P_{i,j}^r x_{ij} \quad r = 1, 2 \)

Subject to

\[ \sum_{j=1}^{8} W_{i,j} x_{i,j} \leq b_i \quad i = 1, \ldots, 7. \]

\[ \sum_{i=1}^{7} x_{i,j} = 1 \quad j = 1, \ldots, 8. \]

\[ x_{i,j} = 0 \text{ or } 1; \forall i = 1, \ldots, 7 \text{ and } j = 1, \ldots, 8 \]

Where,

\[
\begin{bmatrix}
85 & 95 & 83 & 92 & 6 & 21 & 31 & 32 \\
1 & 36 & 76 & 66 & 68 & 81 & 39 & 57 \\
13 & 9 & 14 & 30 & 95 & 98 & 26 & 66 \\
40 & 1 & 17 & 8 & 63 & 98 & 67 & 62 \\
52 & 56 & 77 & 17 & 40 & 38 & 7 & 75 \\
9 & 68 & 18 & 96 & 80 & 24 & 8 & 72 \\
9 & 18 & 60 & 7 & 39 & 79 & 34 & 42 \\
\end{bmatrix}
, P_{i,j}^2 =
\begin{bmatrix}
58 & 77 & 898 & 92 & 86 & 91 & 33 & 32 \\
45 & 89 & 2 & 99 & 45 & 6 & 43 & 54 \\
31 & 86 & 9 & 63 & 34 & 89 & 63 & 47 \\
51 & 96 & 53 & 56 & 26 & 58 & 62 & 12 \\
28 & 91 & 90 & 94 & 20 & 65 & 29 & 94 \\
1 & 9 & 41 & 14 & 23 & 48 & 63 & 45 \\
63 & 34 & 45 & 2 & 48 & 45 & 32 & 33 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
66 & 73 & 70 & 94 & 15 & 4 & 49 & 46 \\
68 & 28 & 29 & 79 & 31 & 15 & 48 & 48 \\
60 & 24 & 78 & 32 & 12 & 20 & 79 & 3 \\
53 & 32 & 89 & 26 & 85 & 60 & 41 & 42 \\
44 & 11 & 88 & 92 & 12 & 89 & 28 & 5 \\
86 & 94 & 66 & 41 & 84 & 93 & 59 & 98 \\
78 & 69 & 49 & 48 & 2 & 45 & 76 & 35 \\
\end{bmatrix}
, b_i =
\begin{bmatrix}
83 \\
42 \\
25 \\
72 \\
12 \\
64 \\
57 \\
\end{bmatrix}
\]

We solve the above instance by the Two Phases Method.

The Phase one

Step 1:
(i) Find the optimal solution $Z^1 (620)$, then substitute that solution of $Z^1$ to find $Z^2 (353)$.

(ii) Find the optimal solution $Z^2 (587)$, then substitute that solution of $Z^2$ to find $Z^1 (341)$.

**Step 2:** Apply the weighted sum method by Aneja and Nair \[17\] to find all supported non-dominated points. $(Z^1, Z^2) = (620, 353)$ and $(Z^1, Z^2) = (587, 341)$.

The supported non-dominated points from the first phase are shown in Table 1 and described in the Figure 6.

| No. | Supported nond. points |
|-----|------------------------|
| 1   | (620, 353)             |
| 2   | (596, 451)             |
| 3   | (556, 499)             |
| 4   | (523, 537)             |
| 5   | (435, 579)             |
| 6   | (341, 587)             |

![Figure 6: Supported nondominated points](image)

**The Phase two**

Using the above sixth supported non-dominated points obtain in first phase, we find the unsupported non-dominated points by creating five triangles. Each triangle has its hypotenuse between two adjacent supported non-dominated points, then apply the Phase two to find the unsupported non-dominated points in each triangle. The unsupported nondominated points from phase two are show in Table 2 and described in Figure 7. $(z_1, z_2) = (606, 357)$ $(z_1, z_2) = (567, 454)$ $(z_1, z_2) = (468, 541)$ $(z_1, z_2) = (376, 580)$

| No. | Unsupported nond. points |
|-----|--------------------------|
| 1   | (606, 357)               |
| 2   | (567, 454)               |
| 3   | (468, 541)               |
| 4   | (376, 580)               |

![Figure 7: Unsupported nondominated points](image)
Finally, for this illustrative example, we get ten non-dominated points given in Table 3 and Figure 8.

Table 3: All nondominated points

| No. | Nondominated point | point type    |
|-----|-------------------|---------------|
| 1   | (620, 353)        | Supported nond. |
| 2   | (606, 357)        | Unsupported nond. |
| 3   | (596, 451)        | Supported nond. |
| 4   | (567, 454)        | Unsupported nond. |
| 5   | (556, 499)        | Supported nond. |
| 6   | (523, 537)        | Supported nond. |
| 7   | (468, 541)        | Unsupported nond. |
| 8   | (435, 579)        | Supported nond. |
| 9   | (376, 580)        | Unsupported nond. |
| 10  | (341, 587)        | Supported nond. |

Figure 8: All nondominated points

4. Computational study

The Table 4 shows a comparison of Two Phases Method and Balanced Box Method by using randomly generated test instances for BOGAP of varied sizes (20 \(4 \times 5\), 25 \(5 \times 5\), 56 \(7 \times 8\), 100 \((10 \times 10)\), 400 \((20 \times 20)\), 625 \((25 \times 25)\), 900 \((30 \times 30)\) and 1600 \((40 \times 40)\)).

These random test instances for BOGAP were generated on C program, and random coefficients were generated in the interval \([20, 2500]\). Both Two Phasas Method and Balanced Box Method algorithms were implemented in C and run on a 2.93 GHz workstation with 2 GB of RAM memory. CPLEX 12.1 \([20]\) was the solver used.

The Figure 9 shows the CPU time statistics of Two Phases Method and Balanced Box Method for the given BOGAP instances.

5. Conclusion

In this paper, a bi-objective generalized assignment problem was considered to find all non-dominated points by using two methods: Two Phases Method and Balanced Box Method. A comparison between the two methods indicate that TPM performance is better compare to the BBM. As a prospective study of this model, mixed integer program and other comparison \([21]\) will be investigated in subsequent publications.
Table 4: A comparison Two Phasas Method and Balanced Box Method for BOGAP

| Class | Two Phasas Method | Balanced Box Method |  |
|-------|-------------------|---------------------|---|
|       | CPU (Sec.) | IP's | No.Nond.* | CPU (Sec.) | IP's | No.Nond.* |
| C20   |           |     |           |           |     |           |
| 0.0151 | 8   | 4   | 0.0154 | 12 | 4 |
| 0.0125 | 8   | 4   | 0.0304 | 12 | 4 |
| 0.0115 | 10  | 5   | 0.0344 | 15 | 5 |
| 0.0219 | 12  | 6   | 0.0445 | 18 | 6 |
| 0.0316 | 12  | 6   | 0.0495 | 18 | 6 |
| C25   |           |     |           |           |     |           |
| 0.0218 | 6   | 3   | 0.0249 | 9 | 3 |
| 0.0481 | 8   | 4   | 0.0532 | 12 | 4 |
| 0.0820 | 8   | 4   | 0.1235 | 12 | 4 |
| 0.0246 | 10  | 5   | 0.0354 | 15 | 5 |
| 0.8622 | 10  | 5   | 0.1256 | 15 | 5 |
| C56   |           |     |           |           |     |           |
| 0.3092 | 30  | 15  | 0.4939 | 45 | 15 |
| 2.2077 | 56  | 28  | 3.6488 | 84 | 28 |
| 1.015  | 26  | 13  | 1.5367 | 39 | 13 |
| 1.7901 | 22  | 11  | 2.2497 | 33 | 11 |
| 1.2191 | 38  | 19  | 2.9366 | 57 | 19 |
| C100  |           |     |           |           |     |           |
| 1.7096 | 86  | 43  | 2.1952 | 129 | 43 |
| 0.8696 | 70  | 35  | 1.3433 | 105 | 35 |
| 1.2423 | 84  | 42  | 1.8667 | 126 | 42 |
| 1.8567 | 74  | 37  | 2.2245 | 111 | 37 |
| 0.7298 | 58  | 29  | 1.1184 | 87 | 29 |
| C400  |           |     |           |           |     |           |
| 17.2497 | 298 | 149 | 56.4184 | 447 | 149 |
| 26.8678 | 274 | 137 | 50.7103 | 411 | 137 |
| 27.4580 | 270 | 135 | 49.9887 | 405 | 135 |
| 34.1718 | 316 | 158 | 58.0312 | 474 | 158 |
| 28.4286 | 284 | 142 | 52.5916 | 426 | 142 |
| C625  |           |     |           |           |     |           |
| 56.8223 | 478 | 239 | 136.3314 | 717 | 239 |
| 36.6104 | 368 | 184 | 84.3307 | 552 | 184 |
| 22.3297 | 312 | 156 | 101.2659 | 468 | 156 |
| 36.7271 | 396 | 198 | 143.5519 | 594 | 198 |
| 32.4449 | 356 | 178 | 66.2620 | 534 | 178 |
| C900  |           |     |           |           |     |           |
| 157.7154 | 708 | 354 | 323.0213 | 1062 | 354 |
| 196.2855 | 792 | 396 | 424.23 | 1188 | 396 |
| 110.7546 | 598 | 299 | 226.7212 | 897 | 299 |
| 235.2311 | 862 | 431 | 273.0010 | 690 | 230 |
| 91.6092 | 518  | 259 | 199.5028 | 777 | 259 |
| C1600 |           |     |           |           |     |           |
| 727.5382 | 1062 | 531 | 1673.338 | 1593 | 531 |
| 1084.4557 | 1360 | 680 | 2711.139 | 2040 | 680 |
| 368.8044 | 970 | 395 | 626.9675 | 1185 | 395 |
| 1080.4999 | 1372 | 686 | 2485.15 | 2085 | 686 |
| 514.3021 | 948  | 474 | 2711.139 | 1422 | 474 |

* Number of non-dominated points
Figure 9: CPU time statistics of Two Phases Method and Balanced Box Method for BOGAP
References

[1] Martello, Silvano. Knapsack problems: algorithms and computer implementations. John Wiley & Sons Ltd., 1990.

[2] Ross, G Terry and Soland, Richard M. “A branch and bound algorithm for the generalized assignment problem”. In: Mathematical programming 8.1 (1975), pp. 91–103.

[3] Silvano, Martello and Paolo, Toth. “An algorithm for the generalized assignment problem”. In: Operations Research Proceedings 1981 18 (1981), pp. 589–603.

[4] Cattrysse, Dirk G and Van Wassenhove, Luk N. “A survey of algorithms for the generalized assignment problem”. In: European journal of operational research 60.3 (1992), pp. 260–272.

[5] Shmoys, David B and Tardos, Éva. “An approximation algorithm for the generalized assignment problem”. In: Mathematical programming 62.1-3 (1993), pp. 461–474.

[6] ¨Oncan, Temel. “A survey of the generalized assignment problem and its applications”. In: INFOR: Information Systems and Operational Research 45.3 (2007), pp. 123–141.

[7] Munapo, Elias, Lesoaona, Maseka, Nyamugure, Philimon, and Kumar, Santosh. “A transportation branch and bound algorithm for solving the generalized assignment problem”. In: International Journal of System Assurance Engineering and Management 6.3 (2015), pp. 217–223.

[8] Sethanan, Kanchana and Pitakaso, Rapeepan. “Improved differential evolution algorithms for solving generalized assignment problem”. In: Expert Systems with Applications 45 (2016), pp. 450–459.

[9] Antunes, Carlos Henggeler, Alves, Maria João, and Clímaco, João. Multiobjective Linear and Integer Programming. Springer, 2016.

[10] Ehrgott, Matthias and Gandibleux, Xavier. “A survey and annotated bibliography of multiobjective combinatorial optimization”. In: OR-Spektrum 22.4 (2000), pp. 425–460.

[11] Zhang, Cai Wen and Ong, Hoon Liong. “An efficient solution to biobjective generalized assignment problem”. In: Advances in Engineering Software 38.1 (2007), pp. 50–58.

[12] Prakash, S, Sharma, MK, and Singh, A. “Pareto optimal solutions for multi-objective generalized assignment problem: general article”. In: South African Journal of Industrial Engineering 21.2 (2010), pp. 91–100.

[13] EL, ulungu and J., Teghem. “The two phases method: an efficient procedure to solve bi-objective combinatorial optimization problems”. In: Foundations of Computing and Decision Sciences 20.2 (1995), pp. 149–165.

[14] Boland, Natasha, Charkhgard, Hadi, and Savelsbergh, Martin. “A criterion space search algorithm for biobjective integer programming: The balanced box method”. In: INFORMS Journal on Computing 27.4 (2015), pp. 735–754.

[15] Ehrgott, Matthias. Multicriteria optimization. Vol. 491. Springer Science & Business Media, 2005.

[16] Steuer, Ralph E. Multiple criteria optimization: theory, computation, and applications. Wiley, 1986.

[17] Aneja, Yash P and Nair, Kunhiraman PK. “Bicriteria transportation problem”. In: Management Science 25.1 (1979), pp. 73–78.

[18] Przybylski, Anthony, Gandibleux, Xavier, and Ehrgott, Matthias. “Two phase algorithms for the bi-objective assignment problem”. In: European Journal of Operational Research 185.2 (2008), pp. 509–533.

[19] Visée, Marc, Teghem, Jacques, Pirlot, Marc, and Ulungu, Ekunda L. “Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem”. In: Journal of Global Optimization 12.2 (1998), pp. 139–155.

[20] Cplex, IBM. ILOG CPLEX 12.6 Optimization Studio. 2014.

[21] Al-Hasani, Ali, Al-Rabeeah, Masar, Kumar, Santosh, and Eberhard, Andrew. “An Improved Triangle Splitting Method for Solving Biobjective Mixed Integer Programming Problems”. In: International Journal of Mathematical, Engineering and Management Sciences (2018).