Benders Decomposition Algorithms To Solve Bi-Objective Linear Programmes

Ali Sohrabi[1], Richard Lusby[2], Andrea Raith[1]
Engineering Science, University of Auckland, New Zealand[1]
Department of Technology, Management and Economics, Technical University of Denmark, Denmark[2]
asoh438@aucklanuni.ac.nz

Abstract

We present two variants of a Benders decomposition (BD) algorithm to solve Bi-objective Linear Programmes (BLPs). BD is incorporated within the bi-objective simplex algorithm by decomposing the problem into a Bi-objective Master Problem (BM) and a Bi-objective Sub Problem (BS). Like the bi-objective simplex algorithm, the Bi-objective Benders Simplex Algorithm (BBSA) aims to find a set of extreme efficient solutions by proceeding iteratively from the optimiser of one objective to that of the other. In this algorithm, iteratively, an efficient solution of BM is chosen to be solved with BS, this is called exploring an efficient solution in BS. If BS is infeasible, a feasibility cut is added to BM; otherwise, an optimality cut is generated for each efficient solution of BS and added to BM. The BBSA stops when no new non-dominated point in BM needs to be explored. Since solving BS is a time-consuming step in BBSA, we propose a variant of BBSA that generates fewer weighted optimality cuts from BS. We also describe a bidirectional variant of BBSA, where the idea is to solve the problem by simultaneously starting from the minimisers of the first and second objectives. This variant of the algorithm proceeds in two directions until the same non-dominated point is found from each direction. We compare the performance of the proposed variants of the BBSA on different problem instances.

1 Introduction

The nature of a multi-objective optimisation problem involves several objectives with conflicting objectives, and it is usually impossible to find a unique optimal solution to optimise all objectives simultaneously. Many real-world problems are modelled with complex mathematical programs that are large. Different methods have been presented, but interest in utilising decomposition techniques, particularly BD [Benders, 1962], has grown due to their applicability in single-objective large scale optimisation problems [Rahmaniani et al., 2017]. Some decomposition approaches such as column generation and BD can be integrated in the procedure of the bi-objective simplex algorithm. Raith et al. [2012] describe how the entering variable selection of the bi-objective simplex algorithm can be reformulated as a column generation subproblem for BLP. Moradi et al. [2015] adapt this approach for the bi-objective multi-commodity network flow problem by developing a bi-objective version of Dantzig-Wolfe decomposition. Sohrabi et al. [2022] propose a new methodology for the Benders reformulation of BLPs, present the BBSA, where BD is integrated with the bi-objective simplex algorithm.

In this paper, firstly, some fundamental concepts of BLPs are presented in the remainder of Section 1. Section 2 then gives an overview of the BBSA that is proposed in Sohrabi et al. [2022] and describes two new variants of the BBSA. We summarise our computational study in Section 3 and present our conclusions in Section 4.

A BLP in standard form is given by $\left\{\min \begin{pmatrix} (\hat{c}^1)^\top x \\ (\hat{c}^2)^\top x \end{pmatrix} = \begin{pmatrix} z^2(x) \\ z^1(x) \end{pmatrix} | x \in \mathcal{X} \right\}$, where $\mathcal{X} = \{x | \hat{A}x = \hat{b}, x \geqq 0\}$ denotes the **feasible set** and $\hat{A}$, $\hat{b}$, $\hat{c}^1$, $\hat{c}^2$ have appropriate dimensions. A feasible solution $x \in \mathcal{X}$ is an **efficient** solution if there is no $\overline{x} \in \mathcal{X}$ such that $z(\overline{x}) \leq z(x)$ (i.e. $z_i \leqq \bar{z}_i$ for $i = 1, 2$ and $z \neq \bar{z}$). The set of efficient solutions is called the **efficient set** denoted by $\mathcal{X}_e$. The image $z(x)$ of $x \in \mathcal{X}_e$ is called a **non-dominated** point. The set of all non-dominated points is denoted with $\mathcal{Z}_N$. A set $\bar{\mathcal{X}}_e$ is called a **complete set** of efficient solutions if for each efficient solution $x' \in \mathcal{X}_e \setminus \bar{\mathcal{X}}_e$ there exists at least one $x^* \in \bar{\mathcal{X}}_e$ such that $z(x') = z(x^*)$.

To solve a multi-objective, particularly bi-objective, linear programme, different scalarisation techniques can be employed that solve a sequence of single-objective optimisation problems, where the original problem with multiple objectives is converted into a related single-objective one. A widely used scalarisation is the weighted-sum technique that converts multiple objectives into one by weighting and summing objectives. For given λ, a BLP in the standard from can be converted into a single-objective optimisation problem of the form $\{\min{(\lambda\hat{c}^1 + (1-\lambda)\hat{c}^2)^\top x} \mid x \in \mathcal{X}\}$, which we term λ-BLP. It is well-known that each efficient solution is an optimal solution of a λ-BLP for a weight $\lambda \in (0, 1)$. There are some non-dominated points that are the optimal solutions of λ-BLP for a range of values of λ. These non-dominated points are termed extreme non-dominated points, and efficient solutions corresponding to these extreme non-dominated points are termed extreme efficient solutions. To find a complete set of extreme non-dominated points, a sequence of problems λ-BLP can be solved for a set of $\lambda = a_1, a_2, \ldots, a_{l+1} \in (0, 1)$. Each extreme efficient solution is an optimal solution of λ-BLP for all $\lambda \in [a_i, a_{i+1}]\, \forall i = 1, 2, \ldots, l$. Alongside the weighted-sum scalarisation technique, other methods, like the bi-objective simplex algorithm, can be employed to identify the extreme efficient solutions and λ-values.

2 BD Algorithms for BLPs

In this section, we consider BD in a bi-objective setting. We introduce the bi-objective Benders reformulation of a BLP and describe the BBSA. We then present the two new variants of the BBSA. Consider a BLP of the following form:

$$\begin{array}{llclcl} \min & (c^1)^\top x & + & (f^1)^\top y & & \\ \min & (c^2)^\top x & + & (f^2)^\top y & & \\ \text{s.t.} & Ax & + & By & \geqq & b \\ & & & Dy & \geqq & d \\ & x, & & y & \geqq & 0. \end{array} \tag{1}$$

Here $x \in \mathbb{R}^n$, $y \in \mathbb{R}^q$ and vectors c^1, c^2, f^1, f^2, b, d and matrices A, B, and D have appropriate dimensions. Because of the specific structure of (1), BD can be employed to decompose problem (1) into a BM (2) that includes the y variables and a BS (3) that includes the x variables as follows:

$$\begin{array}{llclcl} \min & \theta_1(y) & + & (f^1)^\top y & & \\ \min & \theta_2(y) & + & (f^2)^\top y & & \\ \text{s.t.} & & & Dy & \geqq & d \\ & & & x, y & \geqq & 0 \end{array} \tag{2}$$

$$\begin{array}{llclcl} \min & (c^1)^\top x & & & & \\ \min & (c^2)^\top x & & & & \\ \text{s.t.} & Ax & \geqq & b & - & B\bar{y} \\ & x & \geqq & 0 & & \end{array} \tag{3}$$

Here $\theta_1(\bar{y})$ and $\theta_2(\bar{y})$ define the objective values of (3), and $\bar{y}$ is a fixed solution obtained from (2). In Sohrabi et al. [2022], it is shown that only generating optimality cuts based on the individual objective functions of BS is insufficient to fully describe all extreme efficient solutions and corresponding non-dominated points of (1). Instead, we need to generate what we call weighted optimality cuts as shown in the third set of constraints of (4). Suppose $\mathcal{D}_p^\lambda$ and $\mathcal{D}_r$ are the set of extreme points and extreme rays, respectively, of the dual of problem BS (3) with weighted-sum objective with weight λ. Then, we can reformulate the BM (2) as:

$$\begin{array}{lrcrcrcll} \min & \theta_1 & + & & & (f^1)^\top y & & & \\ \min & & & \theta_2 & + & (f^2)^\top y & & & \\ \text{s.t.} & & & & & Dy & \geqq & d, & \\ & & & & & (b-By)^\top \pi_r & \leqq & 0 & \forall \pi_r \in \mathcal{D}_r, \\ & -\lambda\theta_1 & - & (1-\lambda)\theta_2 & + & (b-By)^\top \pi_p & \leqq & 0 & \forall \pi_p \in \mathcal{D}_p^\lambda, \lambda \in [0,1], \\ & & & & & y & \geqq & 0. & \end{array} \tag{4}$$

While the reformulation contains an infinite number of weighted optimality cuts, Sohrabi et al. [2022] prove that a finite set of weighted optimality cuts is sufficient to identify a complete set of extreme efficient solutions and corresponding non-dominated points of (1).

2.1 BBSA for BLPs

A full description of the BBSA can be found in Sohrabi et al. [2022], and a brief outline is provided here. The BBSA iteratively identifies non-dominated points of the Benders master problem BM by applying bi-objective parametric simplex iterations. The initial BM is a problem of the form (2), which is obtained from (4) by removing all optimality and feasibility cuts. This Benders reformulation with a subset of feasibility and optimality cuts that have been added up to the current iteration of the BBSA, is the problem we refer to as BM in the context of the BBSA. BM is iteratively updated by adding all identified optimality and feasibility cuts.

We term a non-dominated point $\bar{z} = (\bar{z}_1, \bar{z}_2)$ of BM an *explored non-dominated point* if a BS (3) has been solved for an efficient solution of BM $(\bar{\theta}_1, \bar{\theta}_2, \bar{y})$ associated with this non-dominated point (i.e. $\bar{z}_i = \bar{\theta}_i + (f^i)^\top \bar{y}, i = 1, 2$). Otherwise, this non-dominated point is termed an *unexplored non-dominated point* of BM. In each iteration, BBSA identifies at most one unexplored non-dominated point $\bar{z}$ of BM and explores it by solving BS with $\bar{y}$ for the corresponding efficient solution $(\bar{\theta}_1, \bar{\theta}_2, \bar{y})$ of BM. If BS is infeasible, $\bar{y}$ is removed from BM by adding a feasibility cut. Otherwise, BS is fully solved, i.e. all non-dominated extreme points of BS are found with the bi-objective simplex algorithm. Corresponding to each non-dominated point of the BS,

Algorithm 1 BBSA to solve bi-objective linear optimisation problems

Input: A, B, D, $C = (c^1, c^2)$, $F = (f^1, f^2)$, b, d.

1 Initialise: $\mathcal{Z}_n \leftarrow \{\}$, $\mathcal{X}_e \leftarrow \{\}$, $\overline{y} \leftarrow \overline{y}^1$, $\lambda \leftarrow 1$
2 $[\overline{\theta}_1, \overline{\theta}_2, \overline{y}^1] \leftarrow$ Solve BM for the first objective function with BD
3 $\overline{z} \leftarrow$ Calculate the non-dominated point of step 2
4 **do**
5 Explore $\bar{z}$ by solving BS for given $\overline{y}$ with the bi-objective simplex algorithm
6 **if** BS infeasible: Add feasibility cut to BM; **else** add weighted optimality cut(s)
7 $\mathcal{Z} \leftarrow \{z^1, \dots, z^l\}$: Start bi-objective simplex algorithm from the last feasible explored non-dom. point z^1; Find first unexplored non-dominated point z^l.
8 **if** *first iteration* **then**
9 $\overline{z} \leftarrow \mathcal{Z}(1)$, $(\overline{\theta}_1, \overline{\theta}_2, \overline{y}) \leftarrow (\theta_1^1, \theta_2^1, y^1)$
10 **end**
11 **if** $\overline{z}$ *remains feasible* **then**
12 $\bar{\mathcal{Z}}_n \leftarrow \bar{\mathcal{Z}}_n \cup \{\overline{z}\}$, $\bar{\mathcal{X}}_e \leftarrow \bar{\mathcal{X}}_e \cup \{(\overline{\theta}_1, \overline{\theta}_2, \overline{y})\}$
13 **end**
14 $\overline{z} \leftarrow z^l$, store $[(\overline{\theta}_1, \overline{\theta}_2, \overline{y}), \lambda = a_l]$
15 **while** *There exists an unexplored non-dominated point in BM*;

Output: $\mathcal{X}_e$ and $\mathcal{Z}_n$ from which extreme solutions can be filtered

a weighted optimality cut is generated and added to BM. By adding these optimality cuts to BM, the values of θ_1 and θ_2 in BM are adjusted as necessary. The BBSA terminates when there are no remaining unexplored non-dominated points of BM.

Algorithm 1 provides an overview of the BBSA algorithm. The BBSA starts by finding the first extreme non-dominated point, which is optimal for the first objective function, using the standard BD technique (line 2). Then BS is solved with the bi-objective simplex algorithm to find the first extreme non-dominated point of BM. Since the minimiser is feasible for the first objective, a set of weighted optimality cuts is generated. All of the cuts that are found when solving the initial single-objective problem and BS are added to BM, and a weighted BM is solved for $\lambda = 1$ to obtain non-dominated point z^1 with corresponding efficient solution $(\theta_1^1, \theta_2^1, y^1)$. From here, the bi-objective simplex algorithm is utilised to move to the first unexplored non-dominated point z^l (line 7) and to thereby determine a range of values of λ, for which $(\theta_1^1, \theta_2^1, y^1)$ remains an optimal solution of the weighted BM problem. Point z^l is then the next non-dominated point to be explored. We denote this non-dominated point and its efficient solution as $\overline{z}$ (line 14) and $(\overline{\theta}_1, \overline{\theta}_2, \overline{y})$, respectively, and set $\lambda = a_l$ (line

14). BS is then solved with $\overline{y}$. A feasibility cut is generated (line 6) if BS is infeasible, and this is used to remove $\overline{y}$ and $\overline{z}$ from BM. One or more weighted optimality cuts are generated if BS is feasible (line 6) to update θ_1 and θ_2 if necessary. All new cuts are added to BM, and the next iteration can then be performed. At each iteration the set of efficient solutions and non-dominated points is updated (line 11-13) if $\bar{z}$ is still a non-dominated point ($\bar{z} \in \mathcal{Z}$), and an unexplored non-dominated point is chosen to be explored if it exists (line 14). The BBSA stops when there are no remaining unexplored non-dominated points.

2.2 Variant 1 of BBSA

The BBSA explores an unexplored non-dominated point of BM in each iteration by solving BS, which requires considerable effort. In the original BBSA, this is done by fully solving the BS, which means identifying a complete set of extreme efficient solutions, with the bi-objective simplex algorithm, and then generating a weighted optimality cut corresponding to each solution. However, it can be challenging to solve BS, especially for problems that are degenerate, which makes fully solving BS time-consuming. Secondly, generating all weighted optimality cuts may not be required to identify a new unexplored non-dominated point. In this section, we show that, to explore an efficient solution of BM, at most two weighted optimality cuts need to be generated by identifying at most two extreme efficient solutions of BS. We term this variant of the BBSAthe BBSA 2 (BBSA2).

BBSA2 proceeds in the same iterative manner as as described in Algorithm 1. The only difference is in line 6, where a new non-dominated point is explored by solving BS. Assume z^l is the selected unexplored non-dominated point of BM with a corresponding efficient solution $(\theta_1^l, \theta_2^l, y^l)$ that is optimal for a weighted sum version of BM for $\lambda \in [a_l, a_{l+1}]$ in the current iteration. BBSA2 exploits the observation that, to *explore* z^l, it is sufficient for the bi-objective simplex algorithm to start from weight $\lambda = a_l$. Suppose z_{BS}^1 is a non-dominated point that is optimal for the weighted version of BS for this weight λ. From this non-dominated point, the bi-objective simplex moves forward to find a new non-dominated point z_{BS}^2 with $\lambda \in [a_{l+1}^{'}, a_{l+2}^{'}]$ (if it exists). A weighted optimality cut is then generated and added to BM for non-dominated points z_{BS}^1 and z_{BS}^2 (with $\lambda = a_l$ and $a_{l+1}^{'}$, respectively). If there is no new non-dominated point z_{BS}^2, a weighted optimality cut for the second objective is generated instead (i.e. $\lambda = 0$) in addition to a weighted optimality cut for the first non-dominated point z_{BS}^1 (where $\lambda = a_l$).There are two reasons why generating two

Algorithm 2 Bi-Directional Benders Simplex Algorithm (BDBBSA) to solve BLPs

Input: A, B, D, c^1, c^2, f^1, f^2, b, d in (1).

1 Initialise: $[\mathcal{Z}_n^l, \mathcal{X}_e^l] \leftarrow \{\}$, $[\mathcal{Z}_n^r, \mathcal{X}_e^r] \leftarrow \{\}$

2 **do**

3 $\quad [\overline{z}^l, (\overline{\theta}_1^l, \overline{\theta}_2^l, \overline{y}^l)] \leftarrow$ Iterate BBSA2 steps (lines 5-14 of Algorithm 1) **from the left** direction until finding a new feasible explored non-dominated point.

4 $\quad \mathcal{Z}_n^l \leftarrow \mathcal{Z}_n^l \cup \{\overline{z}^l\}$, $\mathcal{X}_e^l \leftarrow \mathcal{X}_e^l \cup \{(\overline{\theta}_1^l, \overline{\theta}_2^l, \overline{y}^l)\}$

5 $\quad [\overline{z}^r, (\overline{\theta}_1^r, \overline{\theta}_2^r, \overline{y}^r)] \leftarrow$ repeat line 3 **from the right**

6 $\quad \mathcal{Z}_n^r \leftarrow \mathcal{Z}_n^r \cup \{\overline{z}^r\}$, $\mathcal{X}_e^r \leftarrow \mathcal{X}_e^r \cup \{(\overline{\theta}_1^r, \overline{\theta}_2^r, \overline{y}^r)\}$

7 **while** $\overline{z}^l \neq \overline{z}^r$;

8 $\mathcal{Z}_n \leftarrow \mathcal{Z}_n^l \cup \mathcal{Z}_n^r$, $\mathcal{X}_e \leftarrow \mathcal{X}_e^l \cup \mathcal{X}_e^r$

Output: $\mathcal{X}_e$ and $\mathcal{Z}_n$ from which extreme solutions can be filtered

weighted optimality cuts is sufficient. Firstly, when an unexplored non-dominated point with associated weight λ is selected, it is not necessary to generate cuts with weights larger than λ from BS as the area associated with larger values of λ in objective space has already been explored. Secondly, in each iteration we do not need to move forward beyond the next non-dominated extreme point in BS to generate weighted optimality cuts as generating them will not affect the exploration of the current non-dominated point of BM.

2.3 Variant 2 of BBSA

In this section, the BDBBSA, a bidirectional version of BBSA2, is developed. To find a complete set of extreme non-dominated points, the BDBBSA explores non-dominated points simultaneously from left to right and also from right to left. The term left to right (right to left) refers to starting from a minimiser of the first (second) objective and exploring non-dominated points in the direction that improves the second (first) objective, respectively. In each iteration of BDBBSA, an unexplored non-dominated point from each direction is chosen to be explored. Similar to the BBSA2 in Section 2.2, it is not necessary to fully solve BM with the bi-objective simplex algorithm in each iteration. Suppose z^{l_i} and z^{r_i}, with corresponding efficient solutions $(\theta_1^{l_i}, \theta_2^{l_i}, y^{l_i})$ and $(\theta_1^{r_i}, \theta_2^{r_i}, y^{r_i})$, are the most recently explored non-dominated points (that are feasible in BM) from the two directions and are optimal solutions of the corresponding weighted sum problem for values of $\lambda \in [a_{l_i}, a_{l_{i+1}}]$, $\lambda \in [a_{r_i}, a_{r_{i+1}}]$, respectively. To find a new unexplored non-dominated point from the left (right) direction, the bi-objective simplex algorithm starts from z^{l_i} (z^{r_i}) and moves forward in a direction to improve the second (first) objective value, and thereby identifies a new unexplored non-dominated point from left (right), if it exists. The idea of BDBBSA adapts BBSA2 to explore from both directions as outlined in Algorithm 2. The algorithm stops when the same explored non-dominated point is identified from two directions.

3 Numerical results

We test and compare BBSA, BBSA2 and the BDBBSA on two types of problem instances with a decomposable structure. The numerical experiments were carried out in MATLAB R2020b and performed on a 64-bit Windows 10 PC system with Intel Xeon W-2145 CPU 3.7 GHz processor and 32 GB RAM.

For Bi-objective Fixed Charge Transportation Problem (BFCTP) instances, we created our own instances by varying the number of supply and demand points ($n \in \{10, 12, 14, 16, 18, 20\}$ and $m \in \{20, 25, 30, 40\}$). Supply capacities and demand requests were uniformly generated in the intervals $[100, 1000]$ and $[0, 50]$, respectively. Fixed and unit costs were uniformly generated in intervals $[30, 100]$ and $[0, 5]$. For each combination of the possible n and m values 10 instances were generated giving a total of 240 instances. Groups of instances are referred to as n_m in the following. We also tested the performance of the proposed algorithms on Bi-objective Multiple Knapsack Problem (BMKP) that are based on Angulo et al. [2016] who formulate a single-objective stochastic version of the problem, although we consider the deterministic version of the problem here. There are 30 instances of identical problem dimensions available for which we generate a second objective function.

3.1 BFCTP results

We observe that each BFCTP instance group has a varying number of extreme non-dominated points, between 21 and 82 in total. There tend to be more extreme non-dominated points as instances become more challenging, i.e. with higher number of suppliers n and demands m. Figure 1 shows the variation in runtime of the different instances for the BBSA and its variants. As can be observed, increasing

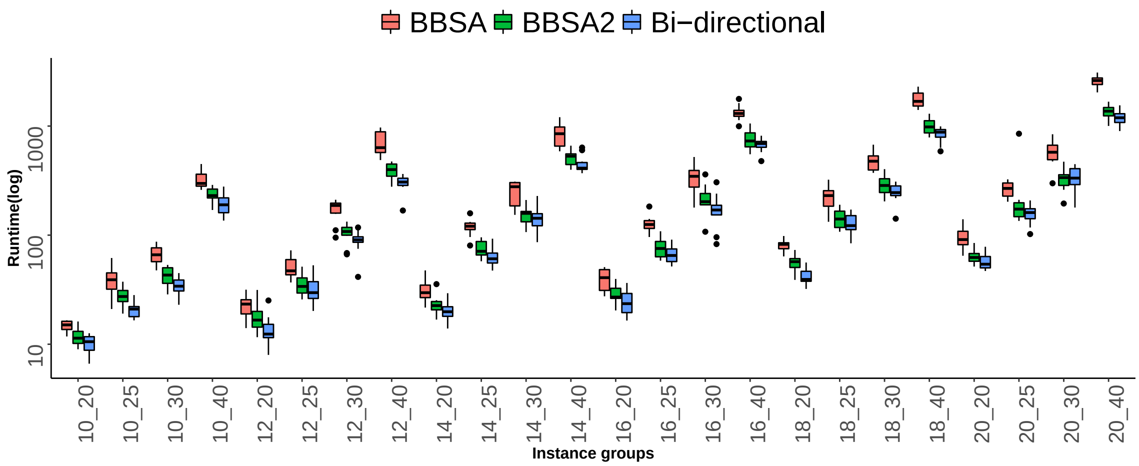

Figure 1: Runtime of the BBSA and its two variants for BFCTP instances

problem size, i.e. increasing the number of suppliers n and demands m, leads to more challenging and time-consuming problem instances. It is clear that for all groups of instances, the two described variants of the BBSA perform better and can identify the complete set of extreme points faster than the BBSA, particularly when n and m have larger values. Our results indicate that BBSA2 performs better than BBSA, which is expected as it reduces the computational effort in BS. Furthermore, on average, BDBBSA even works better than BBSA2 for BFCTP instances. In terms of time taken to identify an extreme non-dominated point, on average, our observation shows that, for BBSA, it takes between 0.5 to 34.6 seconds, and it is between 0.41 and 18.52 and between 0.35 and 16.26 seconds for BBSA2, and BDBBSA, respectively.

It is interesting to understand the key factors affecting algorithm performance. Figure 2.A shows the number of explored non-dominated points to identify an extreme non-dominated point of BM. BBSA explores between 2.58 and 6.31 non-dominated points, whereas BBSA2 and BDBBSA explore between 2.57 and 5.91 and between 1.92 and 5.48, respectively. One explanation for why BDBBSA has fewer explored non-dominated points is its ability to use information (in the form of cuts generated) from two directions in solving BM thereby avoiding to explore some non-dominated points that should be cut off.

The number of degenerate pivots in BM and BS per iteration is another key factor that impacts the algorithm performance. BBSA2 has the lowest number of degenerate pivots of BS per iteration. For BDBBSA, most of the degeneracy happens in

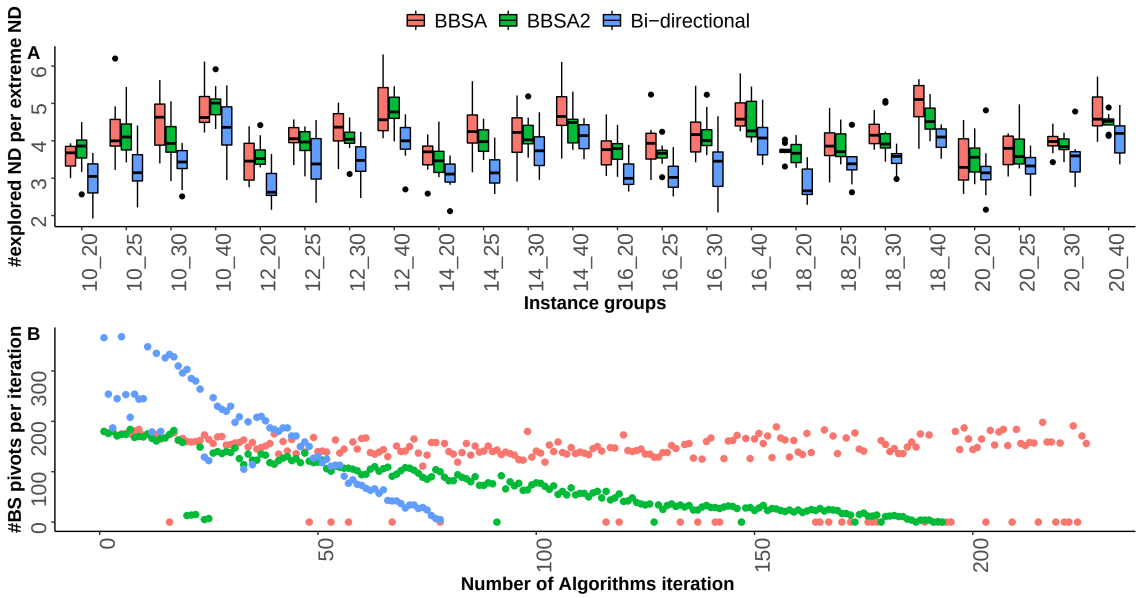

Figure 2: Comparison of number of explored non-dominated points of BBSA variants for BFCTP instances (A) and number of pivots in BS for one 12_30 instance (B).

the initial steps when we solve problems with bigger (smaller) λ from left (right) in BS. The reason is that for BFCTP instances, there is often only one extreme non-dominated point in BS. Despite this, when λ is bigger (smaller), many degenerate pivots are required to confirm that there is only one extreme non-dominated point. As such, the number of degenerate pivots in the BDBBSA is higher than in the BBSA2, and it is highest for the BBSA where BS is always fully solved. To clarify this matter, we report the number of pivots for solving one BS (i.e. per iteration) in all three algorithms (Figure 2.B) on an example of BFCTP in group 12_30. It is shown here that solving BS in the BDBBSA has many more pivots than the other two algorithms in the initial iterations. This is because in each iteration of the BDBBSA, two BS are solved, one from each direction, and the total number of BS pivots is reported here. In contrast with BBSA2 and BDBBSA, the BBSA has almost the same number of pivots in BS in each iteration as it always fully solves BS with the bi-objective simplex algorithm. When the number of BS pivots is shown as zero it means that a feasibility cut is identified. Another interesting point in BBSA2 and BDBBSA is that in the last iterations of the algorithm, the number of pivots of BS approaches zero as λ gets smaller. In BM on the other hand, BDBBSA has the lowest number of degenerate pivots since it benefits from more information, namely

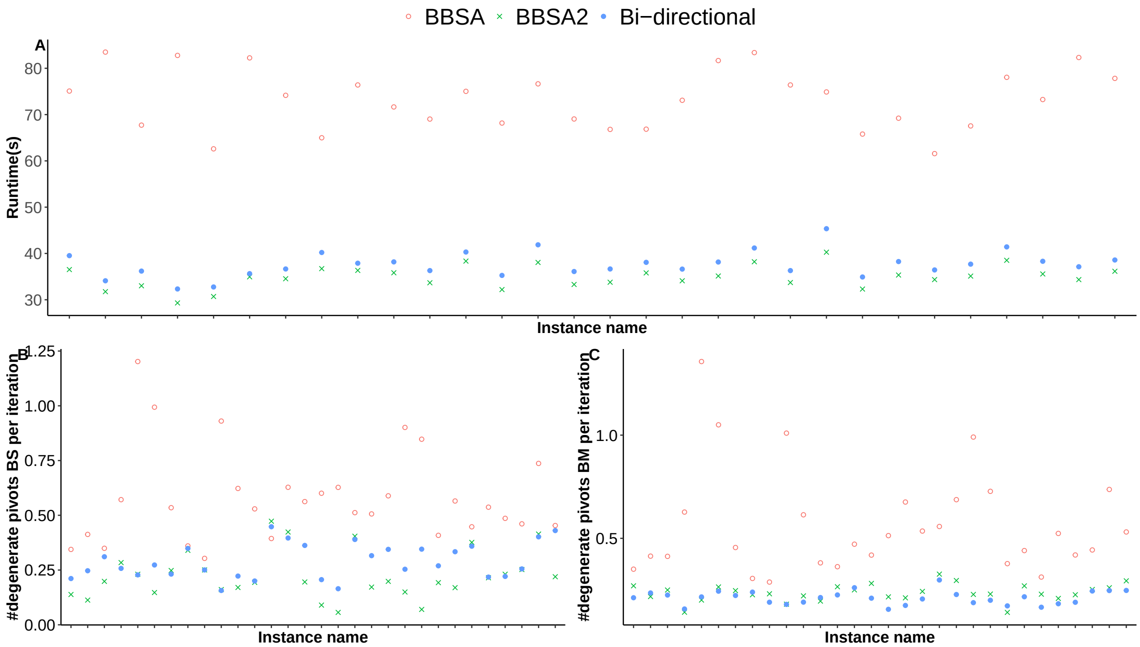

Figure 3: Comparison of key factors of different algorithms for BMKP instances

the cuts generated from both directions and fewer pivots are required to find a new unexplored non-dominated point from each direction.

3.2 BMKP results

Figure 3.A shows the runtime of the different algorithms for the BMKP instances. Again, the two variants of the BBSA perform much better than the BBSA in identifying a complete set of extreme efficient solutions, where BBSA2 is slightly faster than BDBBSA. Solving BM is slower in BDBBSA compared to the BBSA2 since BDBBSA uses the cuts generated from two directions, which makes BM larger and therefore slower to solve. In terms of the number of explored non-dominated points per extreme non-dominated point, all algorithms tend to have a similar ratio, between 1.3 and 1.7 for most instances. Our results also indicate that BBSA has more degeneracy in BS and BM compared with the other two algorithms. This is because we fully solve BS in each BBSA iteration, and for BM, there are more constraints in comparison with the other two algorithms (see Figures 3.B, 3.C).

4 Conclusion and Future work

In this paper, we presented two variants of the BBSA Benders decomposition algorithm to solve BLPs. We compare the performance of the proposed algorithm with two types of problem instances. It is observed that both variants show better performance than the original BBSA, and factors that affect problem difficulty are discussed. There are some topics that can be investigated in the future. Firstly, proposing an algorithm where a sequence of weighted-sum problems are solved with BD when the next weight λ is identified is of interest. Secondly, extending the presented algorithms to solve stochastic bi-objective optimisation problems with multiple scenarios is another interesting topic that can be investigated.

Acknowledgement: This research has been partially supported by the Marsden Fund, grant number 16-UOA-086.

References

Gustavo Angulo, Shabbir Ahmed, and Santanu S Dey. Improving the integer L-shaped method. *INFORMS Journal on Computing*, 28(3):483–499, 2016.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962. ISSN 09453245, 0029599x.

S. Moradi, A. Raith, and M. Ehrgott. A bi-objective column generation algorithm for the multi-commodity minimum cost flow problem. *European Journal of Operational Research*, 244(2):369–378, 2015.

R. Rahmaniani, T.G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3): 801–817, 2017. ISSN 03772217, 18726860.

A. Raith, S. Moradi, M. Ehrgott, and M. Stiglmayr. Exploring bi-objective column generation. In *ORSNZ 46th Annual Conference Proceedings*, pages 288–296, 2012.

A.A Sohrabi, R. Lusby, and A. Raith. Benders decomposition for bi-objective linear programs. *European Journal of Operational Research, Submitted*, 2022.