



**HAL**  
open science

## Bi-level multi-objective combinatorial optimization using reference approximation of the lower level reaction

Malek Abbassi, Abir Chaabani, Lamjed Ben Said, Nabil Absi

### ► To cite this version:

Malek Abbassi, Abir Chaabani, Lamjed Ben Said, Nabil Absi. Bi-level multi-objective combinatorial optimization using reference approximation of the lower level reaction. *Procedia Computer Science*, 2020, 176, pp.2098 - 2107. 10.1016/j.procs.2020.09.246 . hal-03493948

**HAL Id: hal-03493948**

**<https://hal.science/hal-03493948>**

Submitted on 17 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

## Bi-level multi-objective combinatorial optimization using reference approximation of the lower level reaction

Malek Abbassi<sup>a,b,\*</sup>, Abir Chaabani<sup>a</sup>, Lamjed Ben Said<sup>a</sup>, Nabil Absi<sup>b</sup>

<sup>a</sup>SMARTLab, Institut Supérieur de Gestion de Tunis, University of Tunis (Université de Tunis), Tunisia

<sup>b</sup>LIMOS, Ecole des Mines de Saint-Etienne, Georges Charpak Gardanne, France

### Abstract

Bi-level optimization has gained a lot of interest during the last decade. This framework is suitable to model several real-life situations. Bi-level optimization problems refer to two related optimization tasks, each one is assigned to a decision level (i.e., upper and lower levels). In this way, the evaluation of an upper level solution requires the evaluation of the lower level. This hierarchical decision making necessitates the execution of a significant number of Function Evaluations (FEs). When dealing with a multi-objective optimization context, new complexities are added and imposed by the conflicting objectives and their evaluation techniques. In this paper, we aim to reduce the induced complexity using approximation techniques in order to obtain the lower level optimality. To this end, ideas from multi-objective optimization have been extracted, improved, and hybridized with evolutionary methods to build an efficient approach for Multi-objective Bi-Level Optimization Problems (MBLOPs). In this work, three techniques are suggested: (1) a complete lower level approximation Pareto front procedure, (2) a reference-based approximation selection procedure, and (3) a sub-set reference-based approximation selection one. The proposed variants are applied to a new multi-objective formulation of a well-known combinatorial problem integrating two systems in the supply chain management, namely, the Bi-level Multi Depot Vehicle Routing Problem (Bi-MDVRP). The statistical analysis demonstrates the efficiency of each algorithm according to a set of metrics. Indeed, a large number of savings are detected which confirms the efficiency of our proposals for solving combinatorial optimization problems.

© 2020 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of KES International.

**Keywords:** Bi-level optimization; Multi-objective optimization; Evolutionary algorithms; Approximation techniques.

### 1. Introduction

Bi-level optimization problems involve two interactive decision making tasks to be optimized. These processes are integrated into a nested structure which makes the principle problem (i.e., the problem at the upper level) sequentially

*E-mail address:* malekkabbassi@gmail.com

dependent on the secondary optimization problem (i.e., the problem at the lower level). Generally, two Decision Makers DMs (the leader and the follower) are collaborating to ensure the whole optimization process. This hierarchical design of optimization is suitable to model a large number of real-life applications [3, 17, 4]. However, bi-level optimization is characterized by an inherent NP-hard complexity even in the simplest case of linear bi-level programming [10]. Furthermore, the presence of a related lower level optimization problem in the Bi-Level Optimization Problems (BLOPs) brings other struggles of disconnectedness and non-convexity. In fact, the evaluation depends on the follower's reaction. In this regard, each upper level solution is passed to the lower level task to perform the lower optimization using a search algorithm to find a near-optimal solution. The interaction between both optimization levels is generally complex, and thus increases the computational cost of the evaluation process. This complexity leads frequently to NP-hard problems [16]. It is important to note that the evaluation step consumes the majority of the needed effort to achieve the convergence of such algorithm.

In the context of MBLOPs, the lower level optimization problem has a set of trade-off optimal solutions. These trade-off solutions should be considered all as optimized reactions taken into consideration to evaluate one upper solution. This fact complicates the task for the leader, which performs its search for solutions that are fixed as parameters for optimal trade-off solutions of the follower.

A variety of classical and evolutionary methods have been conducted on bi-level optimization research area. In the field of classical optimization, we observe the presence of intensive studies using approximate solution techniques solving bi-level problems. Although the efficiency of these methods, they still restricted to small instances of BLOPs. Besides, most of these works are limited to a set of regularity properties such as linearity, convexity, differentiability or smoothness assumptions [13]. This fact makes them inapplicable for complex non-convex, non-linear or large dimensional bi-level problems. Regarding Evolutionary Algorithms (EAs), there exists a panoply of solution methods that have shown their effectiveness to solve such problems [1, 7]. However, these population based-algorithms are computationally expensive since they necessitate a huge number of FEs. For instance, if we have 100 individuals at the upper level, and the population is evolved for 50 generations, 50 individuals at the lower level with 50 generations, the total number of FEs of a nested evolutionary algorithm will be 12.5 million ( $50 \times 100 \times 50 \times 50$ ) FEs. To cope with this high computational cost, several research studies have been investigated in the literature. In this context, we cite the work of [Sinha et al. \(2013\)](#) where the authors proposed a quadratic approximation to find a near-optimal lower solution consuming a small number of FEs. Another attempt to reduce the number of FEs is the study of [Chaabani et al. \(2018\)](#) who proposed a co-evolutionary decomposition-based scheme. The majority of these works deal with single objective bi-level optimization problems. In the context of MBLOPs, we find only the work [Deb and Sinha \(2009\)](#), [Ankhili \(2018\)](#), etc. These approaches have proven their efficiency on several test problems. However, they are restricted to the continuous research area. In this paper, we are working in the context of discrete decision variables since the majority of real situations can be modelled as combinatorial optimization problems. The idea of this paper is inspired by multi-objective optimization ideas and evolutionary algorithms to solve efficiently such kind of problems. We are motivated by evolutionary algorithms within the nested scheme since they present: (1) the high-quality solutions in the upper level [15]. (2) the upkeep of the structure of BLOPs to provide the most appropriate solution. (3) the powerful features of EAs which are unfortunately limited to small size problems. Thus, we plan to extend their strengths to solve efficiently MBLOPs with large size decision variables. Particularly, we aim in this paper to optimize the bi-level multi-objective optimization evaluation step, allowing then the scalability of such solution approach to high dimensional problems.

The key contributions of this work are:

- Proposing a new approach for solving bi-level multi-objective optimization with three variants able to (1) neutralize the lower multi-objective effect in the upper evaluation step which is the most computational complexity consuming part in EAs. (2) Reduce the complexity of bi-level multi-objective optimization solution method by reducing the required number of FEs.
- Proposing a new multi-objective formulation of the Bi-MDVRP, which is a well-studied problem in operation research [3].
- Reporting a statistical analysis of our approaches for solving the multi-objective Bi-MDVRP.

## 2. Basic definition and related work

Bi-level optimization problems contain two tasks of optimization with two decision levels. The upper optimization problem is known as the leader (upper level) and the lower optimization problem is referred to as the follower (lower level). The bi-level framework includes two types of decision vectors, namely, the upper level decision vector  $x_u$  and the lower level one  $x_l$ . The leader acts as a fixed parameter for the follower. Besides, each level has its own objectives and constraints. Considering the number of objectives, the multi-objective bi-level optimization is a natural extension of single objective bi-level optimization where at least one problem is presented as a multi-objective optimization problem. MBLOPs require the optimization of more than one objective at a single decision level. This type of optimization problem includes three variants: (1) multiple conflicting objectives are only handled by the upper level, (2) multiple conflicting objectives are only handled by the lower level, and (3) multiple conflicting objectives are handled by both levels. In this work, we deal with bi-level multi-objective optimization problems with a single upper level objective. Figure 1 illustrates a general sketch of this variant's structure necessitating interlinked optimization at both levels. The Figure shows that for each upper solution (marked with blue point) there corresponds a set of possible obtained Pareto fronts (curves in red color). The solving algorithm aims to approximate the lower level Pareto front to be returned to the upper level. To this end, a general mathematical formulation of the problem can be presented as follows:

$$\begin{aligned} & \text{Min } F(x_u, x_l) \\ & \text{s.t } G(x_u, x_l) \leq 0 \begin{cases} x_l \in \underset{x_l}{\text{argmin}} f(x_u, x_l) = f_1(x_u, x_l), \dots, f_n(x_u, x_l) \\ \text{s.t } g(x_u, x_l) \leq 0 \end{cases} \end{aligned} \quad (1)$$

Where the optimization of the upper objective function  $F(x_u, x_l)$  depends on the lower optimal Pareto front for  $f_1(x_u, x_l), \dots, f_n(x_u, x_l)$  (which are the lower objective functions constrained by  $g(x_u, x_l)$ ).

Existing bi-level solution methods are classified into two main categories of groups: (1) classical methods and (2) evolutionary ones. The first family includes mainly the trust region function methods [6], descent methods [16], penalty functions methods [2] etc. Despite the significant progress made in classical optimization, the main shortcoming of these techniques is that they depend on the mathematical properties of the BLOP. This makes them unadaptable to handle real-world problems difficulties such as non-linearity, non-convexity, and large dimensionality. The second family which includes EAs has been successfully applied to solve BLOPs. In this way, attempts have been made to solve BLOPs using these methods, we cite [7] [8] [1]. However, the advantages come with a trade-off that most of the evolutionary bi-level algorithms are computationally very intensive due to the nested structure imposed by the problem. To address this problem researches have been investigated to reduce the computational expense of evolutionary bi-level optimization algorithms by utilizing meta-modelling-based principles such as the quadratic approximations based algorithm of [14] and the kriging approximations based approach of [13]. In fact, these proposed techniques are suggested to the continuous single BLOPs. Similarly, regarding the multi-objective bi-level optimization there exists a number of works solving the multi BLOPs in the continuous research area. However little has been done on MBLOPs in the combinatorial case. This topic remains relatively less explored and we acknowledge the need for methodological improvements.

## 3. Proposed method

As we mentioned before, the main shortcoming of the nested solution methods is their computationally cost which makes them not viable for large scale bi-level combinatorial optimization problems [15]. Besides, solving BLOPs associated with multiple objective conflicts remains a challenging issue. Based on the above observations, we design a new bi-level framework that applies lower level approximations to reduce the total required number of FEs for MBLOPs. Figure 2 illustrates the general scheme of our proposed approach incorporating our three suggested types of interaction between the two levels: When the variation step at the upper level is terminated, an upper offspring population  $Q_u$  is formed. Now, for each upper child solution, the upper decision variable  $x_u$  is passed as a fixed

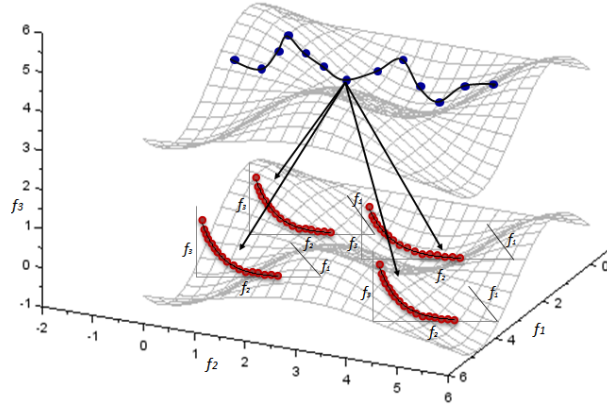


Fig. 1. Bi-level multi-objective optimization nested scheme.

parameter to the lower level to execute the lower optimization task for  $T_l$  generations. After that, the lower level response becomes ready to be transferred for the leader. To this end, the upper algorithm is able now to perform the evaluation step on the complete solutions. At this stage, we model the lower reaction using three alternatives: (1) return the whole lower Pareto front for the upper evaluation as performed in the classical scheme which is described in part "c" from Figure 2, (2) send the best solution presenting the maximum compromise between lower level Pareto solutions (part "b" in Fig. 2), (3) consider the best trade-off solutions (part "a" in Fig. 2), as a surrogate sub-set of the whole Pareto lower level front. We note that all the mentioned classical and improved variants share the same classical upper level environmental selection procedure. This can be done while filling the next upper population with all the lower obtained non-dominated solutions saved in an archive  $A_r$ . It is important to note that we use the Genetic Algorithm (GA) to evolve the upper level population for  $T_u$  generations. The upper level step by step procedure used for all the above mentioned variants can be summarized as follows:

- **Step 1 (Initialization):** Randomly generate the upper level initial population of size  $N_u$ . Then, the lower level algorithm is executed to optimize the lower level objectives for each upper individual.
- **Step 2 (Upper level parent selection):** Choose  $(N_u/2)$  best members from the parent population using the binary tournament selection operator.
- **Step 3 (Upper level Variation):** Perform the upper evolution through the crossover and mutation operators.
- **Step 4 (Offspring upgrading and evaluation):** Execute the corresponding lower level approximation technique for each  $x_u$ . Then, the lower level vectors are combined with their corresponding upper level ones to form the entire offspring solutions. The algorithm is able now to perform the evaluation step on the complete solutions.
- **Step 5 (Environmental selection):** Fill the new upper population with the best  $N_u$  solutions. If the leader uses an alternative described in section 3.2 or 3.3, then, all the lower level stored solutions in  $A_r$  need to be combined with their upper level decision vectors. The process is repeated from Step 2 until the stopping criteria is met.

### 3.1. Classical complete lower level evaluation technique

The traditional bi-level framework appeals the complete Pareto front obtained at the lower level for the upper level to continue the upper evaluation procedure. The lower level may use the GA on the lower population of size  $N_l$  and the overall process can be described as follows:

- **Step 1 (Initialization):** Randomly generate the initial population based on the fixed upper level solution.
- **Step 2 (Lower level parent selection):** Select  $(N_l/2)$  promising solutions that can participate in the lower mating pool using the binary tournament selection operator.
- **Step 3 (Lower level variation):** Perform crossover and mutation operators to create the offspring population.

- **Step 4 (Offspring evaluation):** Evaluate the lower obtained offspring solutions using the lower level objective functions and constraints.
- **Step 5 (Environmental selection):** Append the new population with the best  $N_l$  individuals based on the Non-Dominated Sorting Algorithm (NDSA) and the crowding Distance (CD) measure of [7]. The overall process is repeated from Step 2 until reaching the stopping criteria.

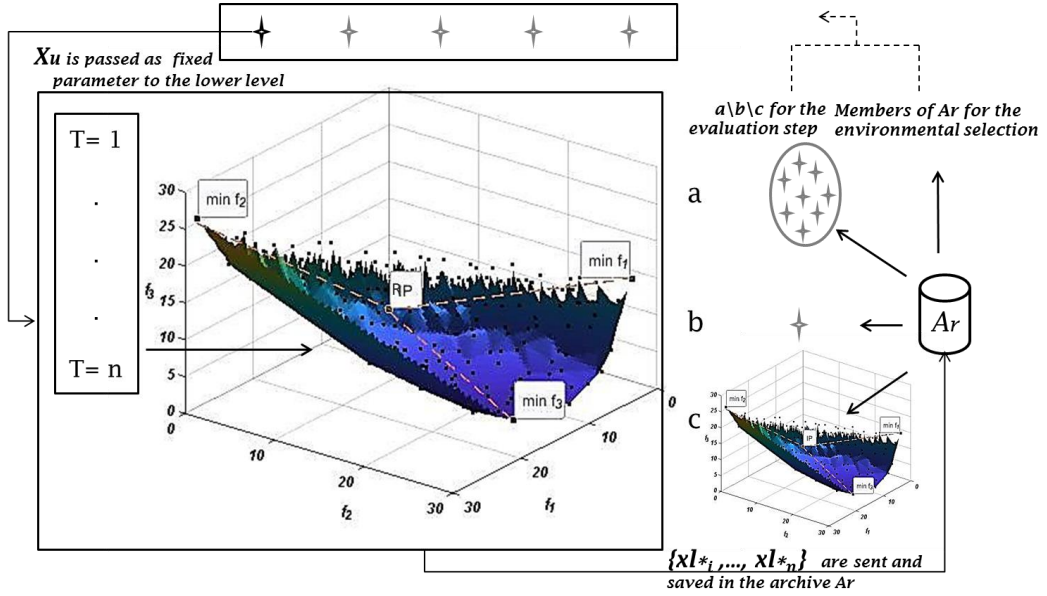


Fig. 2. General scheme of the multi-objective bi-level optimization interaction using: sub-set lower approximation technique (a), lower level single solution approximation technique (b), and classical complete lower level evaluation technique (c).

### 3.2. Lower level single solution approximation technique

In this section, we aim to provide a new variant of the nested multi-objective resolving method. The main motivation behind this proposal is to cope with the high computational cost spent by the traditional nested approach. Thus, we propose to proceed with a reduction-based evaluation technique aiming to identify a single solution presenting the best compromise lower Pareto front. In this context, a crucial issue that should be considered is whether this solution is cooperative or non-cooperative. A cooperative solution is obtained under an optimistic cooperative follower's behaviour to send the best lower solution. However, in pessimistic bi-level optimization a non-cooperative solution is obtained when the follower selects the worst reaction for the leader [15]. In this work we address the optimistic case. Therefore, we can use an explicit formulation to evaluate a multi-objective bi-level solution using a single lower level surrogate one that we note  $x_l^s$  (equation 2).

$$(x_{ui}, (x_{l_{j,1}}, x_{l_{j,2}}, x_{l_{j,3}}) \dots (x_{l_{n,1}}, x_{l_{n,2}}, x_{l_{n,3}})) \quad i \in I, j \in n \quad \text{becomes} \quad (x_{ui}, x_l^s) \quad (2)$$

Where  $I$  is the upper level dimension and  $n$  is the lower level one. The step by step procedure is as follows:

- **Step 1 (Initialization):** Randomly generate the initial population based on the sent upper level solution.
- **Step 2 (Lower level parent selection):** Select the best solutions to participate in the lower mating pool.
- **Step 3 (Lower level variation):** Perform the lower evolution through the crossover and mutation operators.
- **Step 4 (Offspring evaluation):** Evaluate the lower obtained offspring solutions using the lower level objective functions and the constraints.
- **Step 5 (Environmental selection):** Choose the best  $N_l$  solutions, using NDSA and CD measure, to obtain a new lower level population.
- **Step 6 (Archive updating):** Fill the archive with the complete obtained approximate Pareto front to be combined with the new best upper  $N_u$  solutions during the replacement step.

- **Step 7 (Best lower level representative compromise technique):** Identify a Real Surrogate Point (RSP) which is passed to the leader. We note here that the identification of the RSP solution is detailed in subsection 3.4.

### 3.3. Sub-set lower approximation technique

The challenge when trying to reduce the computational effort needed in multi-objective optimization is to maintain the initial basic structure of the problem. Therefore, we opt to keep the fundamental dominance structure combining the previously stated complete lower approximate Pareto front based-evaluation and single objective lower level reduction. Moreover, the target of this strategy is to find a compromise between reducing the number of generated FEs and the number of returned lower decision variables, while preserving the original structure of the multi-objective optimization problem. In this regard, the lower level optimization procedure generates a subset of Pareto optimal solutions that will be used in the upper level evaluation. However, the entire lower level Pareto front is always saved in the archive  $A_r$  to be injected in the new parent population via new upper population replacement step. The step by step procedure is as follows:

- **Step 1 (Initialization):** Randomly generate the initial population.
- **Step 2 (Lower level parent selection):** Select the candidate solutions to participate in the lower mating pool.
- **Step 3 (Lower level variation):** Perform the lower level evolution through the crossover and mutation operators.
- **Step 4 (Offspring evaluation):** Evaluate the obtained lower offspring solutions using the lower level objective functions and constraints.
- **Step 5 (Environmental selection):** Fill the new population based on NDSA and CD replacement strategies.
- **Step 6 (Archive updating):** Fill the archive with the complete obtained approximate Pareto front.
- **Step 7 (Surrogate sub-best compromise set):** The upper level decision maker selects the compromise solution from the obtained lower Pareto front using Algorithm 1. This solution is deleted from the Pareto Front to repeat the extraction procedure and obtain a new one.

### 3.4. New reference point extraction technique for the best compromise situations

The idea of extracting the best compromise solution is inspired by the work of [9, 12]. Besides, we are inspired by the ideal point generation, which employs a single objective optimization to construct the best objective values over the search space. However, we propose to reach our Reference Point (RP) considering the best objective values over the objective space. Thus, the RP allows the upkeep of: (1) the non-dominance structure of the multi-objective optimization and (2) the Pareto optimality feature. Indeed, a strictly based similarity ranking procedure is used to find the closest feasible existing solution to the RP. Figure 3 illustrates the construction of this reference solution: first we identify the best reached fitness value for each objective function over the objective space. Each identified solution is marked with its reached minimum fitness as  $\min_{x \in PF} f_m$  the  $m^{th}$  objective function. Then, we generate the reference point combining coordination using these fitness values. Equation 3 describes the formulation of the RP by combining the coordination of these extreme solutions from the generated approximate Pareto front. Similarly to the concept of the ideal point, the RP is an unfeasible constructed point used only to guide the process of the algorithm to the best compromise solution. For this reason, the purpose here is to achieve the highest level of similarity between the RP and the non-dominated obtained solution to find the real surrogate delegate solution that represents the Pareto front. This process is detailed in Algorithm 1.

$$RP = \min_{x \in PF} f_m(x) \forall m = 1 \dots M, x \text{ is Pareto optimal} \quad (3)$$

## 4. Application to the Multi-objective Bi-level Multi-Depot Vehicle Routing Problem

The Multi-objective Bi-level Multi-Depot Vehicle Routing Problem (M-Bi-MDVRP) is a variant of the Bi-MDVRP. In this variant, we refer to the proposed model in [11], where the leader minimizes the distance between retailers and depots at the upper level. In contrast, the follower minimizes two costs: (1) the direct transport between plants and depots and (2) the production cost aiming to find an approximation of the Pareto front at the lower level. Thus the problem can be formulated as follows:

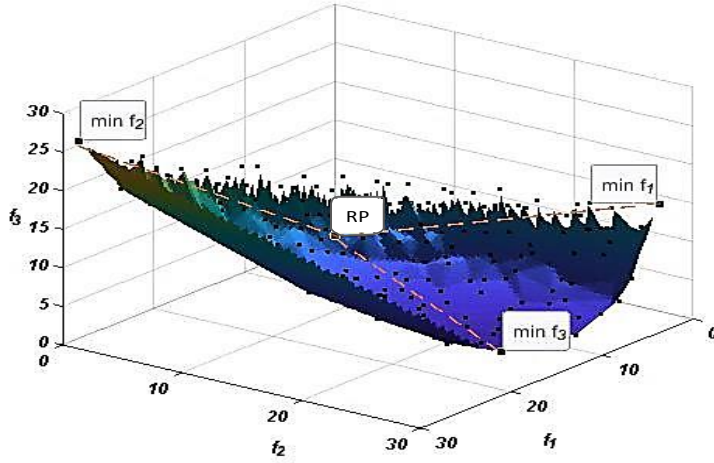


Fig. 3. Reference point for an approximate Pareto front.

$$\text{Min } F(x_{ij}^v, y_{p,d}) = \sum_{v=1}^V \sum_{i,j=1}^E c_{i,j}^a x_{i,j}^v \quad (4)$$

$$\sum_{i=1}^R \sum_{v=1}^V x_{ij}^v = 1, \quad \forall 1 \leq j \leq R \quad (5)$$

$$\sum_{i=1 \setminus \{j\}}^E x_{ij}^v = \sum_{i=1 \setminus \{j\}}^E x_{ji}^v, \quad \forall j \leq E, \forall v \in V \quad (6)$$

$$\sum_{i=1}^E \sum_{j=1}^E d_r x_{ij}^v \leq C_v, \quad \forall v \in V \quad (7)$$

$$u_j^s \geq u_i^s + 1(|R| - 1)(1 - y_{ij}^s), \quad \forall i, j \in R, \forall s \in S \quad (8)$$

$$\text{given } \{x_{ij}^v\}, \{y_{p,d}\} \text{ solves : } \min f(x_{ij}^v, y_{p,d}) = (f_1, f_2), \quad f_1 = \sum_{p=1}^P c_{p,d}^b y_{p,d}, \quad f_2 = \sum_{d=1}^D c_{p,d}^c y_{p,d} \quad (9)$$

$$\text{s.t. } \sum_{d=1}^D \sum_{p=1}^P y_{p,d} = 1 \quad (10)$$

$$\sum_{d=1}^D q_p y_{p,d} \leq PC_p, \quad \forall 1 \leq p \leq P \quad (11)$$

$$\sum_{p=1}^D q_p y_{p,d} \geq \sum_{v=1}^V \sum_{r=1}^R d_r, \quad 1 \leq d \leq D \quad (12)$$

Where  $P, D, R,$  and  $V$  design the set of plants, depots, retailers, and vehicles, respectively.  $x_{ij}^v$  and  $y_{p,d}$  denote the upper and the lower decision variables, respectively.  $E$  is the edge set between retailers and depots,  $d_r$  represents the demand of retailer  $r$  and  $PC_p$  is the production capacity of plant  $p$ .  $q_p$  is the quantity of goods manufactured at  $p$ . We note that  $c_{i,j}^a, c_{p,d}^b,$  and  $c_{p,d}^c$  design the cost of transporting goods from node  $i$  to  $j$  for the leader, the cost of manufacturing a product at plant  $p$  for depot  $d$ , and the cost of transporting goods from  $p$  to  $d$ , respectively.  $u$  is the upper bound of a node reachability indexation. The objective function of the upper problem is defined by (4). According to (5) each retailer is visited exactly once by one assigned vehicle. Constraint (6) implies flow conservation. (7) asserts that the total demand of all retailers on one particular route must not exceed the capacity of the vehicle assigned to this route. Constraint (8) ensures sub-tours elimination. The objective functions of the lower level are defined by (9). (10) imposes variable binary requirements. Constraint (11) guarantees that the total demands of assigned depots should not exceed the production availability of plant  $p$ . Constraint (12) ensures that the demands of retailers should be satisfied by associating a part of plant production to depots.



---

**Algorithm 1** Reference point generation algorithm

---

**Input:** Pareto non-dominated solutions  $Pf$ , Objective functions set  $f$ .

**Output:** Reference Point  $RP$ , Real Surrogate Point  $RS P$ .

```
1: Begin
2:  $min \leftarrow \infty$ 
3: For each  $f_m$  in  $Pf$  do
4:   While ( $f_m < min$ ) do
5:      $min \leftarrow f_m$ 
6:   End While
7: End For
8:  $RP \leftarrow \mathbf{Buildcoord}(\min_{(f_1)} \dots \min_{(f_m)})$  /* Build the reference coordinates with minimum obtained fitness values in  $Pf$ .
9:  $RS P \leftarrow \mathbf{MaxSim}(RP, Pf)$  /* Select a solution from  $Pf$  with high rate of similarity with  $RP$ .
10: End
```

---

## 5. Experimental study

In this section, we conduct an empirical study to evaluate the efficiency and the limits of each of the proposed variants. We perform 31 runs for each instance from the pr suite inspired by Legillon et al. (2013). The nature of stochastic processes imposes sometimes the convergence to the same set of solutions. To cope with this dilemma, the well known Wilcoxon's rank-sum test is used to determine if there is a significant difference between the algorithms' behaviours. These results at level  $\alpha = 0.05$  are presented in the form of (-: no significance) and (+: significance) in the order on which the algorithms appear.

To investigate the performance of the proposed variants, we use the new multi- bi-level multi-depot vehicle routing problem formulated in section 4. Then, we compare the results against the classical nested approach to show the merit of the injected techniques regarding the traditional one.

### 5.1. Experimental settings

Since the lower level optimization procedure is executed interacting with the upper level one, we choose a fewer number of generation and smaller population size at the lower level. We set the parameter values through preliminary experiments as follows:  $N_u = 40$ ,  $N_l = 30$ ,  $T_u = 50$ ,  $T_l = 40$ . The variation operators are inspired from the baseline model (Bi-MDVRP) of [11] for both levels and their probabilities are fixed to: sequence-based-crossover probability = 0.9, uniform mutation probability = 0.1.

### 5.2. Results and Discussion

In this section, we aim to assess the performance of the proposed algorithms conducting a set of computational experiments. We used in this statistical experiments the following criteria: (1) upper level solution quality, (2) number of FEs, and (3) the consumed CPU time.

1) *Solution Quality*: The solution qualities are summarized in Table 1 in terms of best and averaged upper fitness value obtained by the Lower level Single Solution Approximation based variant (LSSA), Sub-set Lower Approximation based variant (SLA), and the Classical complete Lower level Evaluation based variant (CLE), across 31 independent runs, on all the used benchmarks. For each from the used instance, we highlight the best performance in bold. We observe from this Table that the used LSSA algorithm achieves superior performance compared to the classical and the lower subset reduction variants in 6 over 10 used instances in terms of best fitness value. This fact can be explained by the focus on the upper level decision variables in the evaluation step, and the efficiency of considering most similar solution among the candidate ones to the ideal lower level point. For the best fitness evaluation, this variant achieves also a reasonable performance with best results, especially for high dimensional used benchmarks (e.g. M-bipr07, M-bipr08, M-bipr10). The subset approximation variant succeeds as the second best performing algorithm to obtain 4 best results in terms of best fitness.

2) *Number of FEs*: As shown in Figure 4 the best performance is obtained by the proposed variant that incorporates the surrogate approximation extraction technique. This can be attributed to the reference point that allows saving the

number of FEs by summarizing all the lower vector solution i.e., the lower obtained pareto front, in one representative solution. Indeed, the green curve which denotes the LSSA algorithm obtains usually the less FEs on all progress of the

Table 1. Upper level average and best distance obtained by SLA, LSSA, and CLE based algorithms on the M-bipr instances.

Instance	Average Fitness Distance			Best Fitness Distance		
	LSSA	SLA	CLE	LSSA	SLA	CLE
M-bipr01	<b>3310.17</b> ++	3250.16 ++	3415.36 ++	<b>2992.47</b> ++	3076.84 ++	3057.25 ++
M-bipr02	<b>5910.33</b> ++	5940.64 ++	5980.17 ++	5890.16 ++	<b>5884.25</b> ++	5884.58 ++
M-bipr03	11647.12 -+	11778.42 -+	<b>11644.35</b> ++	<b>10968.25</b> ++	11558.36 ++	11578.25 ++
M-bipr04	<b>12940.51</b> ++	12976.15 ++	12996.47 ++	12899.32 ++	<b>12446.26</b> ++	12548.36 ++
M-bipr05	11918.63 ++	11870.16 ++	<b>11864.23</b> ++	11621.24 ++	<b>11351.80</b> ++	11411.14 ++
M-bipr06	18024.71 ++	17998.35 ++	<b>17991.19</b> ++	17105.43 ++	17091.31 ++	<b>17029.65</b> ++
M-bipr07	<b>5160.86</b> ++	5317.74 ++	5203.14 ++	<b>4970.18</b> ++	4980.93 ++	4992.15 ++
M-bipr08	<b>9917.65</b> ++	10233.42 ++	10193.15 ++	<b>9320.11</b> ++	9340.27 ++	9354.21 ++
M-bipr09	13632.21 ++	<b>13524.04</b> ++	13740.59 ++	13051.16 ++	<b>13000.76</b> ++	13291.56 ++
M-bipr10	<b>20714.32</b> ++	20940.98 ++	20862.45 ++	<b>20576.19</b> ++	20580.13 ++	20681.38 ++

number of generations. For instance, it takes only  $10^6$  FEs for 50 upper generations, while the SLA variant consumes more than  $2.4 * 10^7$  FEs, and the classical approach requires  $4 * 10^7$  FEs for the same number of generations (cf. Fig. 4) for bi-pr04. The same observation is confirmed on the other number of generations. We note that we choose the Bi-pr04 instance as a comparison based benchmark because it represents a medium dimension test problem. These results may also characterize the same behaviour of the used algorithm from a computational time viewpoint.

3) *CPU time comparison*: Box plots in Figure 5 illustrate the CPU time consumed for small, medium and large size instances. These box plots reveal a detailed view of the algorithms' performances with the first quartile, the third one, the median, and the outliers. These results remain valid according to our selected combinatorial optimization problem and obviously, the suitable configuration is the reference point based approximation technique variant.

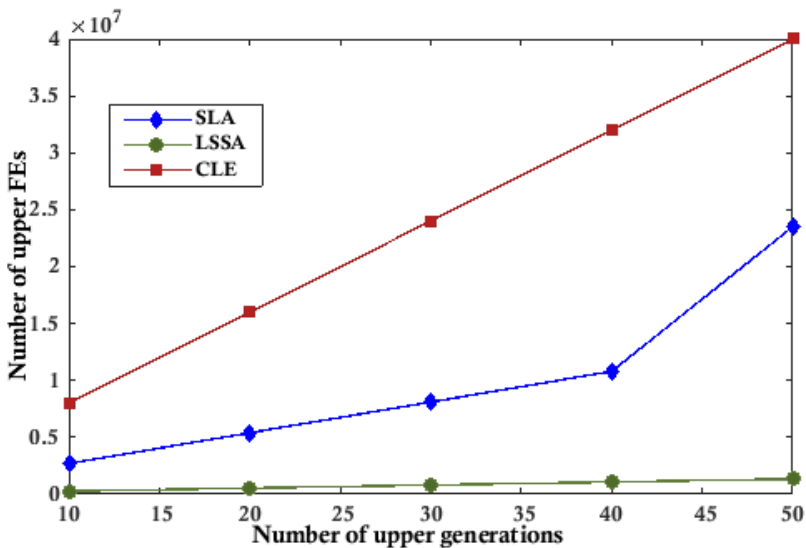


Fig. 4. Numbers of FEs obtained by SLA, LSSA, and CLE at the upper levels on the Bi-pr04 problem.

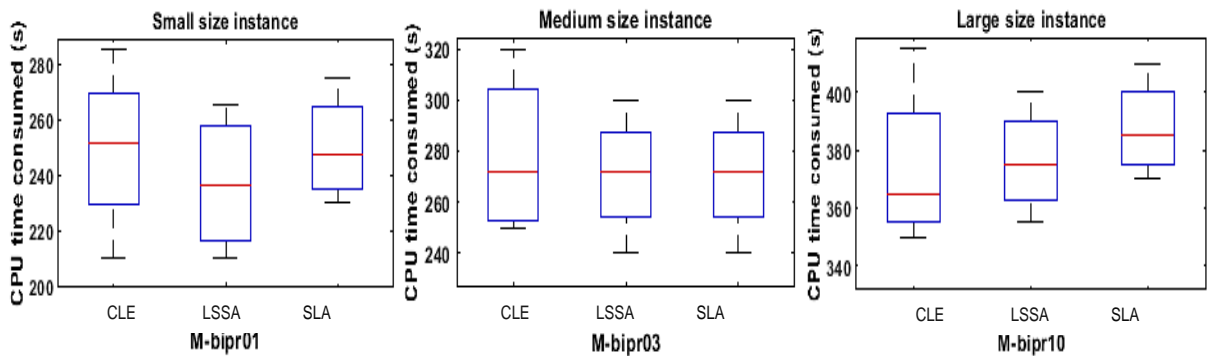


Fig. 5. CPU time consumed for small dimension instance M-bipr01, medium dimension instance M-bipr03, and large dimension instance M-bipr10.

## 6. Conclusion

In this paper, we tackle combinatorial multi-objective bi-level optimization problems by proposing new approximation techniques with the nested approach, that we named LSSA and SLA. The main interest of this work is to incorporate ideas from multi-objective optimization (e.g. ideal point) with the evolutionary process to make the proposed variants efficient in solving combinatorial MBLOP. The experimental results revealed the efficiency of LSSA and SLA when compared to traditional bi-level multi-objective EAs. As future works, we wish firstly to extract a set of recommendations that can be useful for the decision maker while choosing an approximation variant. Secondly, the obtained results are promising, thus it would be a challenging perspective to apply our variants to other real-world bi-level problems.

## References

- [1] Abbassi, M., Chaabani, A., Said, L.B., 2019. An investigation of a bi-level non-dominated sorting algorithm for production-distribution planning system, in: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer. pp. 819–826.
- [2] Ankhili, Z., 2018. Multiobjective bilevel optimization problem: Penalty method, in: *Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications*, pp. 1–4.
- [3] Calvete, H.I., Galé, C., 2010. A multiobjective bilevel program for production-distribution planning in a supply chain, in: *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*. Springer, pp. 155–165.
- [4] Chaabani, A., Said, L.B., 2020. A co-evolutionary decomposition-based algorithm for the bi-level knapsack optimisation problem. *International Journal of Computational Intelligence Studies* 9, 52–67.
- [5] Chaabani, A., et al., 2018. Hybrid codba-ii algorithm coupling a co-evolutionary decomposition-based algorithm with local search method to solve bi-level combinatorial optimization, in: *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE. pp. 506–513.
- [6] Colson, B., Marcotte, P., Savard, G., 2005. A trust-region method for nonlinear bilevel programming: algorithm and computational experience. *Computational Optimization and Applications* 30, 211–227.
- [7] Deb, K., Sinha, A., 2009. Solving bilevel multi-objective optimization problems using evolutionary algorithms, in: *International conference on evolutionary multi-criterion optimization*, Springer. pp. 110–124.
- [8] Fliege, J., Vicente, L.N., 2006. Multicriteria approach to bilevel optimization. *Journal of optimization theory and applications* 131, 209–225.
- [9] Imai, K., Lo, J., Olmsted, J., 2016. Fast estimation of ideal points with massive data. *American Political Science Review* 110, 631–656.
- [10] Jeroslow, R.G., 1985. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical programming* 32, 146–164.
- [11] Legillon, F., Liefoghe, A., Talbi, E.G., 2013. Cobra: A coevolutionary metaheuristic for bi-level optimization, in: *Metaheuristics for Bi-level Optimization*. Springer, pp. 95–114.
- [12] Opricovic, S., Tzeng, G.H., 2004. Compromise solution by mcdm methods: A comparative analysis of vikor and topsis. *European journal of operational research* 156, 445–455.
- [13] Sinha, A., Bedi, S., Deb, K., 2018. Bilevel optimization based on kriging approximations of lower level optimal value function, in: *2018 IEEE Congress on Evolutionary Computation (CEC)*, IEEE. pp. 1–8.
- [14] Sinha, A., Malo, P., Deb, K., 2013. Efficient evolutionary algorithm for single-objective bilevel optimization. *arXiv preprint arXiv:1303.3901*.
- [15] Talbi, E.G., 2013. A taxonomy of metaheuristics for bi-level optimization, in: *Metaheuristics for bi-level optimization*. Springer, pp. 1–39.
- [16] Vicente, L., Savard, G., Júdice, J., 1994. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications* 81, 379–399.
- [17] Wang, X., Bie, Z., Liu, F., Kou, Y., Jiang, L., 2020. Bi-level planning for integrated electricity and natural gas systems with wind power and natural gas storage. *International Journal of Electrical Power & Energy Systems* 118, 105738.