IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 29, 2020, accepted July 17, 2020, date of publication August 3, 2020, date of current version August 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013568

Solving Combinatorial Multi-Objective Bi-Level Optimization Problems Using Multiple Populations and Migration Schemes

RIHAB SAID[1], SLIM BECHIKH[1,2], ALI LOUATI[3], ABDULAZIZ ALDAEJ[3], AND LAMJED BEN SAID[1]

[1]Strategies for Modelling and Artificial inTelligence (SMART) Laboratory, ISG, University of Tunis, Tunis 2000, Tunisia
[2]LMVSR, Kennesaw State University, Kennesaw, GA 30144, USA
[3]Department of Information Systems, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Corresponding author: Slim Bechikh (slim.bechikh@fsegn.rnu.tn)

ABSTRACT Many decision making situations are characterized by a hierarchical structure where a lower-level (follower) optimization problem appears as a constraint of the upper-level (leader) one. Such kind of situations is usually modeled as a BLOP (Bi-Level Optimization Problem). The resolution of the latter usually has a heavy computational cost because the evaluation of a single upper-level solution requires finding its corresponding (near) optimal lower-level one. When several objectives are optimized in each level, the BLOP becomes a multi-objective task and more computationally costly as the optimum corresponds to a whole non-dominated solution set, called the PF (Pareto Front). Despite the considerable number of recent works in multi-objective evolutionary bi-level optimization, the number of methods that could be applied to the combinatorial (discrete) case is much reduced. Motivated by this observation, we propose in this paper an Indicator-Based version of our recently proposed Co-Evolutionary Migration-Based Algorithm (CEMBA), that we name IB-CEMBA, to solve combinatorial multi-objective BLOPs. The indicator-based search choice is justified by two arguments. On the one hand, it allows selecting the solution having the maximal marginal contribution in terms of the performance indicator from the lower-level PF. On the other hand, it encourages both convergence and diversity at the upper-level. The comparative experimental study reveals the outperformance of IB-CEMBA on a multi-objective bi-level production-distribution problem. From the effectiveness viewpoint, the upper-level hyper-volume values and inverted generational distance ones vary in the intervals [0.8500, 0.9710] and [0.0072, 0.2420], respectively. From the efficiency viewpoint, IB-CEMBA has a good reduction rate of the Number of Function Evaluations (NFEs), lying in the interval [30.13%, 54.09%]. To further show the versatility of our algorithm, we have developed a case study in machine learning, and more specifically we have addressed the bi-level multi-objective feature construction problem.

INDEX TERMS Combinatorial bi-level multi-objective optimization, computational cost, indicator-based evolutionary algorithms, population decomposition, migration schemes.

I. INTRODUCTION

Bi-level optimization, as the name reflects, deals with the minimization or the maximization of two interconnected hierarchical levels: (1) a leader, called the upper-level problem, and (2) a follower, called the lower-level problem. The objective is to find the optimum of an upper-level problem, subject to the optimality of the corresponding lower-level problem. The two levels of the problem have their own objective(s), variable(s), and constraint(s). Several problems from the domain of transportation [1], logistics [2], engineering [3], industry [4], and economics [5] have inherent nested structure which needs them to be modeled as bi-level problems. In addition to complexity and increasing size of such problems, other difficulties could occur such as conflecting relationship between objectives of the two levels, non-linearity and/or multimodality in one or both levels, NP-hardness of BLOPs [6], deceptiveness of

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

the lower-level fitness landscape, etc. The BLOPs resolution dates back to the 1970s. Indeed, a variety of methods have been proposed such as Karush-Kuhn-Tucker method [7], the Branch-and-Bound [8], the trust region method [9], etc. However, all these methods are sensitive to the mathematical features of constraint and objective functions such as dimensionality, non-linearity, non-differentiability, etc. Encouraged by the fact that Evolutionary Algorithms (EAs) are less sensitive to these features, researchers have used them for BLOPs resolution such as [10]–[12], and [13]. We assume that, from an effectiveness viewpoint, the results are promising. However, from an efficiency viewpoint, the computational cost is very high, and a large NFEs is needed [14]. Minimizing the computational overhead of the follower EA, and reducing the required number of evaluations have been the most important key pursuits in this domain, and a variety of works have been proposed using different techniques such as hybridization with local search [15], the quadratic approximation method [16], the use of surrogates [17], multi-parametric programming [18], and the use of multicriteria optimization principle [19]. In fact, many decision making situations such as road pricing network problems [20], stone industrial park location problems [21], construction site layout and security planning problems [22], and distribution networks problems with grid-connected microgrid [23]; the upper-level and/or the lower-level involve the optimization of more than one objective. However, few papers have considered BLOPs with multiple objectives [24]–[27], and even the majority of proposed works tackled the continuous case of Multi-Objective Bi-level optimization Problems (MOBPs). Indeed, the reason behind this lack of works is that the bi-level problem becomes both computationally and mathematically intractable even by simplifying assumptions like convexity, continuity, etc. In a MOBP, when evaluating a single upper-level solution, all the lower-level Pareto optimal solutions are considered at the upper-level problem. As follows, the leader will choose a follower solution randomly from the received lower-level Pareto set, which cannot ensure diversity and convergence aspects of the upper-level PF (cf. definition 4 in Appendix C).

Recently, we have proposed a new EA to tackle single-objective BLOPs called CEMBA [28]. Our proposed approach has demonstrated its effectiveness and efficiency in generating good solutions, best upper-level fitness, best lower-level reaction, and less NFEs as described in Appendix A. Motivated by this idea, we propose in this paper, a new version of our CEMBA approach [28] for MOBPs. Our algorithm is termed IB-CEMBA, as it uses, in each level, two populations, and an indicator-based approach. In fact, the reason behind using indicator-based approaches is that these latter ones help the algorithm to approximate the upper-level front and to return the lower-level solution with the best marginal contribution in terms of a multi-objective quality indicator that will be sent to the upper-level. In other words, the solution sent from the follower to the leader is the one with the best indicator contribution and it is not chosen randomly, which is not the case for existing approaches. Indeed, each leader population works with its correponding follower one. Also, a migration scheme is applied in order to guarantee the existence of optimal solutions. The main contributions of this paper are the following:

1) Proposing an efficient approach called IB-CEMBA for combinatorial MOBPs;
2) Preserving the efficiency of the baseline algorithm CEMBA in terms of the NFEs for the multi-objective case (cf. details in Appendix A);
3) Reporting and analyzing comparative results of IB-CEMBA on the Multi-objective Bilevel Production-Distribution Planning problem model with Equilibrium between Supply and Demand, and Workload Balance objective (MBPDPESDWB);
4) Illustrating the versatility of IB-CEMBA on a machine learning case study consisting in solving the Multi-objective Bi-level Feature Construction problem.

II. RELATED WORK

A. MOBP: BASIC DEFINITIONS

Formally, a BLOP consists in optimizing the leader objective function, under some constraints, where one of these constraints represents a nested optimization problem, called the follower. In many practical problem situations, the leader and the follower of a bi-level problem might face multiple objectives. This gives rise to the multi-objective case of bi-level optimization problems. As a definition, the MOBP is a problem with two levels of multi-objective optimization problems in which the lower-level optimal solution represents a feasible space for the upper-level optimization problem. Fig. 1 illustrates the two levels of a MOBP. It explains the fact when all the pareto-optimal solutions of the lower-level are considered at the upper-level. A general MOBP is defined as follows [27]:

$$\begin{aligned}
&Min_{x_u,x_l} F(x) = (F_1(x), \ldots, F_M(x)) \\
&s.t\ G(x) \le 0, H(x) = 0, \\
&\quad x_l \in argmin_{x_l}, \\
&\quad \{f(x) = (f_1(x), \ldots, f_m(x)) | g(x) \le 0, h(x) = 0\}, \\
&\quad x_i^{(L)} \ge x_i \ge x_i^{(U)}, \quad i = 1, \ldots, n.
\end{aligned} \tag{1}$$

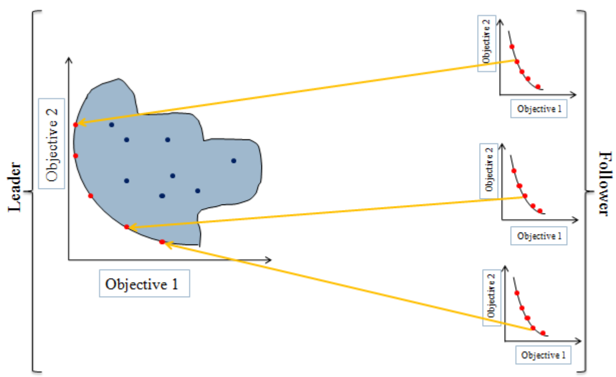


FIGURE 1. Illustration of the two levels of a bi-level optimization problem with two objectives at both levels (inspired by [29]).

Each level has its corresponding objectives, variables, and constraints. The manuscripts u and l are used for the upper-level problem, and the lower-level one, respectively. Consequently, the upper-level variables are represented by x_u, whereas, x_l denotes the lower-level variables. In the previous formulation, F denotes the set of upper-level objective functions that contains M objectives, and f represents the m follower objective functions. In fact, the two levels cannot have common objectives since each level corresponds to an actor. For example, in game theory applications (Stackelberg games), the two actors are in competition. In such situation, one of the actors has more power than the other and acts as a clear leader [14], [30]. For this reason, even if the actors have the same objective function (e.g., cost minimization), these objectives do not have the same meaning since the two actors are in competition. The follower inequality and equality constraints are given by g, and h, recpectively, whereas, G, and H denotes the leader inequality and equality constraints, respectively. It should be noted that the follower optimization problem is optimized using the x_u variable vector as a fixed parameter with respect to the x_l variables. The upper-level problem is optimized by taking the lower-level pareto-optimal set as feasible solutions, and determining its pareto-optimal set by using its objective functions set F and its constraints. We mention here that an explanation of multi-objective optimization aspects is given in Appendix C.

B. EXISTING MULTI-OBJECTIVE BI-LEVEL OPTIMIZATION METHODS

Few papers have considered bi-level optimization problems with multiple objectives on both levels. These works can be classified into two categories: (1) classical approaches, and (2) evolutionary approaches.

1) CLASSICAL APPROACHES

Some of the MOBP studies which exist in literature are mostly directed towards development of classical methods. Reference [24] solved a simple multi-objective bi-level problem using a numerical optimization technique at the lower-level, and an adaptive exhaustive search method at the upper-level. The proposed method is a nested one, in which the lower-level problem is solved to Pareto-optimality. Unfortunately, this solution is non-scalable to large-scale problems. Another method was proposed by [31] which consists in transforming the MOBP into an ϵ-constraint bi-level problem. Indeed, the decision maker chooses the ϵ-parameter, and after that the lower-level problem is replaced by its Karush-Kuhn-Tucker (KKT) conditions. These classical approaches require several executions with several parameters to obtain new pareto fronts. Also, the majority of these approaches tackled the continuous non-linear case and not the discrete one.

2) EVOLUTIONARY APPROACHES

With the surge in the power of computation, different evolutionary approaches have been proposed for MOBP resolution. Reference [32] proposed an evolutionary algorithm for a MOBP with a single-objective at the lower-level, and multiple objectives at the upper-level. The suggested approach is a genetic algorithm for a transportation planning and management problem. Reference [33] proposed a new evolutionary approach for MOBP. Indeed, they use a specialized linear Particle Swarm Optimization (PSO) multi-objective approach in order to solve a multi-component chemical system. Reference [34] proposed an adaptive scalarization based-approach in order to solve a bi-level problem with multiple objectives at both levels. Also, a surrogate-assistance technique is used to minimize the computational cost. Most of the proposed evolutionary approaches cannot be used for the discrete case as they require the gradient information computation.

3) SHORTCOMINGS OF EXISTING APPROACHES

Fewer are works that tackled the combinatorial multi-objective case of BLOPs due to computational and decision making complexities. This observation is explained by the fact that combinatorial BLOPs are known to be strongly NP-hard, and even the evaluation of a solution represents also an NP-hard task [16]. Table 1 summarizes existing multi-objective bi-level optimization approaches that could be applied for the combinatorial case since they do not require the gradient information. We mention here that the cited works are classified into four types of approaches: (1) EXact methods (EX), (2) Trajectory-based Metaheuristics (TM), (3) Evolutionary Metaheuristics (EM), and (4) Hybrid Metaheuristics. It is shown from the table that there is a lack of trajectory metaheuristics and the majority of existing approaches are following a nested structure. For this reason, they are very computationally expensive. The main issue in MOBPs is that the lower-level Pareto-optimal solution set is considered in the upper-level optimization problem. In this case, existing approaches select a lower-level solution randomly from the follower Pareto-optimal set. This solution will be used for the upper-level problem. However, the fact of choosing a solution randomly cannot ensure convergence and diversity of the upper-level PF. In summary, the research gaps in existing combinatorial multi-objective bi-level approaches are the following: (1) the nested structure of algorithms that makes these latter ones very computationally expensive and (2) the choice of the lower-level solution from the follower PF is randomly. We mention here that existing exact methods transform the bi-level problem to a single-level one. For this reason, the solution selection is Not Applied (N/A). In fact, the difference between our proposed approach and existing ones is explained as follows. Our algorithm structure is not a nested one, but it is a decomposition-based co-evolutionary structure. Also, the lower-level solution is not chosen randomly, but we are choosing the lower-level solution having the best contribution in terms of an indicator quality. In summary, in our proposed approach, the computational cost is reduced using decomposition, co-evolution, and migration schemes, while diversity and convergence aspects are ensured using indicator-based approaches.

TABLE 1. Existing multi-objective bi-level optimization approaches that could be used to solve combinatorial MOBPs.

Reference	Algorithm	Method type				Algorithmic structure	Solution selection from the follower PF
		EX	TM	EM	HM		
[32]	Finite string-based genetic algorithm			×		nested	randomly
[33]	LMO PSO: Linear MultiObjective Particle Swarm Optimization			×		nested	randomly
[35]	Bi-objective adaptive ϵ-constraint algorithm using Branch-and-Bound	×				single-level transformation	N/A
[36]	HESA: Hybrid Evolutionary Simulated Annealing				×	nested	randomly
	HGA: Hybrid Genetic Algorithm				×	nested	randomly
	HABC: Hybrid Artificial Bee Colony algorithm				×	nested	randomly
[37]	IMHGA: Improved Multi-objective Hierarchical Genetic Algorithm			×		nested	randomly
[38]	Bi-objective Bilevel algorithm based on explicit enumeration	×				single-level transformation	N/A

III. PROPOSED ALGORITHM

A. MAIN IDEA AND MOTIVATIONS

The main motivation behind this work is to exploit the efficiency of our previously proposed single-objective bi-level algorithm CEMBA [28] to solve MOBPs. The main challenge is how to extend the algorithmic scheme of CEMBA when passing to the multi-objective bi-level case, while preserving its efficiency in terms of the NFEs (Appendix A illustrates the efficiency mechanism of CEMBA). To achieve this goal, we need two kinds of multi-objective metaheuristic algorithms:

1) For the upper-level, we need a multi-objective EA that is able to approximate the PF with enough convergence and diversity (cf. Appendix C); while
2) For the lower-level, we need a multi-objective algorithm that is able to return the solution having the best contribution in terms of a multi-objective quality indicator to the corresponding upper-level problem.

Motivated, by the ability of indicator-based algorithms in transforming a multi-objective vector into a *scalar value* [39] expressing its *marginal contribution* in terms of quality with respect to a set (population) of solutions, we choose to use IBEA (Indicator-Based Evolutionary Algorithm) [39] for PF approximation at the upper-level and IBMOLS (Indicator-Based Multi-Objective Local Search) [40] (i.e. a population-based local search) at the lower-level. In this way, the evaluation of each upper-level solution works in two steps: (1) The upper-level variables are first sent to the lower-level and then (2) IBMOLS approximates the solution having the maximum marginal contribution in terms of the multi-objective quality indicator and send it to the upper-level to be able to terminate the quality evaluation of the considered upper-level solution. We notice here that the indicator-based approach allows respecting the bi-level nature of the optimization problem since it returns a single solution to the upper-level instead of whole set of non-dominated solution (i.e. a whole Pareto Front). As, in this work, the neighborhood size is negligible (much smaller) with respect to (than) the population size, both IBEA [39] and IBMOLS [40] have the same time complexity that is $O(N^2)$, where N is the population size. This further motivates the need to an efficient bi-level algorithmic scheme that significantly reduces the NFEs as possible. This algorithmic design choice allows IB-CEMBA to preserve the population decomposition and migration strategies of the baseline CEMBA [28], which makes it able to solve combinatorial MOBPs with the least possible NFEs. In details, IB-CEMBA uses two populations in each level where each leader population works with its corresponding follower one. The migration scheme is periodically applied in order to guarantee the existence of the optimal follower solutions in the corresponding lower-level population and not in the mismatched one (cf. Appendix A for further details).

B. IB-CEMBA ALGORITHMIC SCHEME

In this subsection, we present the flowchart of IB-CEMBA in Fig. 2, and we describe the IB-CEMBA working principle using a number of steps as follows:

1) **Step 0: Upper and lower population initialization:** Initialize the upper-level population (*UP*) and the lower-level population (*LP*) using DSDM (Discrete Space Decomposition Method) [41] (cf. details in Appendix B) two times in order to obtain, at each level, two initial populations (*UP1*, *UP2*), (*LP1*, *LP2*). The reason behind using a decomposition method is to ensure a uniform coverage of the decision space, and to obtain, as possible, a uniformly distributed solutions set in each level decision space.
2) **Step 1: Lower-level fitness assignment:** In a BLOP, the evaluation of each upper-level solution requires running a whole lower-level algorithm, and this fact represents the main difficulty of BLOP. Motivated by this observation, and in order to handle this issue, we decompose each level of the problem by using two populations. To evaluate lower solutions, the lower-level algorithm of each LP_i (with i belongs to {1,2}) uses the leader solutions from the corresponding UP_i. In other words, we assign the fitness of LP_i solutions with respect to the UP_i solutions.
3) **Step 2: Local search procedure:** As we use the IBMOLS [40] principle for the lower-level algorithm, we apply the local search procedure for each lower-level population LP_i (with i belongs to {1, 2}) as illustrated in Fig. 3. Indeed, we start first by the

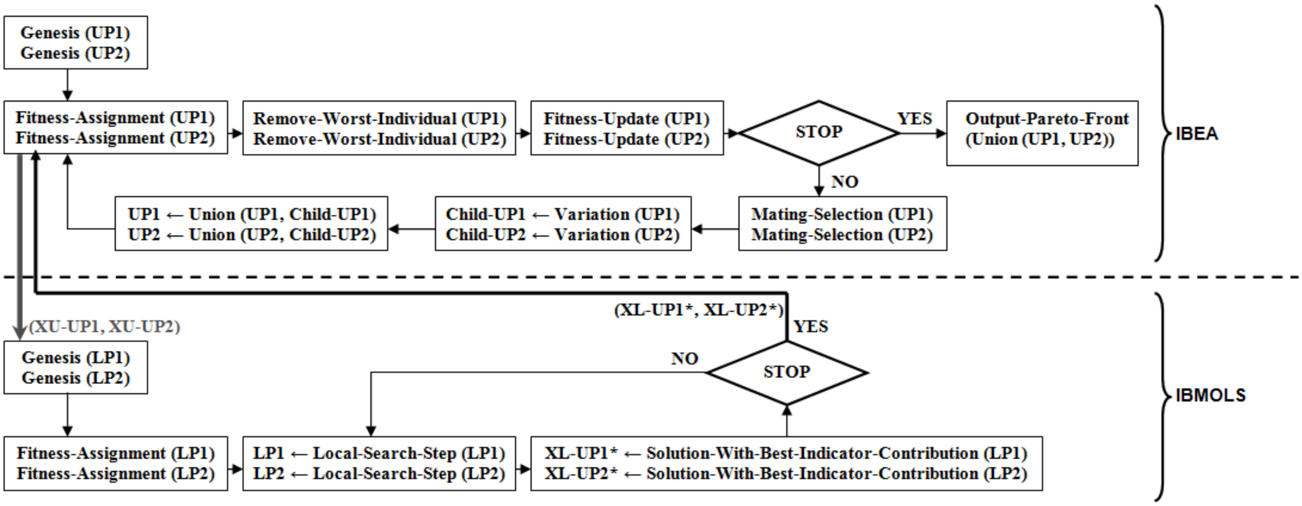


FIGURE 2. The main algorithmic scheme of IB-CEMBA.

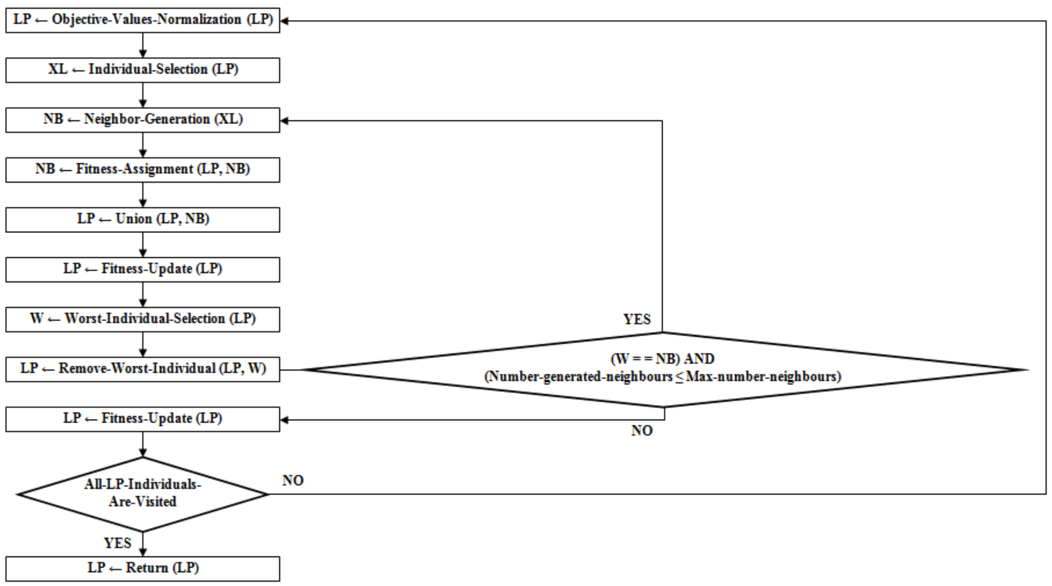


FIGURE 3. Local-Search-Step iteration of IBMOLS at the lower-level of IB-CEMBA.

computation of the normalized objective function values. Consequently, we generate a neighborhood for each lower solution, then, we compute its fitness value based on an indicator I, and the normalized values of the objective function. After that, we update the fitness values, we remove the worst solution, and we update the fitness values again. We mention here that the generation of neighborhood stops in two cases: (1) when the entire solution neighborhood is explored, or (2) when an improving neighboring solution (with respect to I) is found. The entire local search pocedure is terminated when all the lower-level members are visited. We mention here that the use of this indicator-based approach helps to obtain a follower solution with the best indicator contribution which is not the case for existing approaches. In other words, instead of randomly choosing a solution from the optimal lower front, the IBMOLS helps the algorithm to obtain a single lower solution (with the best indicator contribution) that will be sent to the upper-level.

4) **Step 3: Best indicator contribution lower-level solution determination:** As the evaluation of leader solutions from UP_i (with i belongs to $\{1,2\}$) necessitates the approximation of each corresponding follower solutions from LP_i, the lower-level solutions are evaluated with respect to the upper-level members.
5) **Step 4: Upper-level indicator-based procedure:** After receiving the lower-level solutions with the best

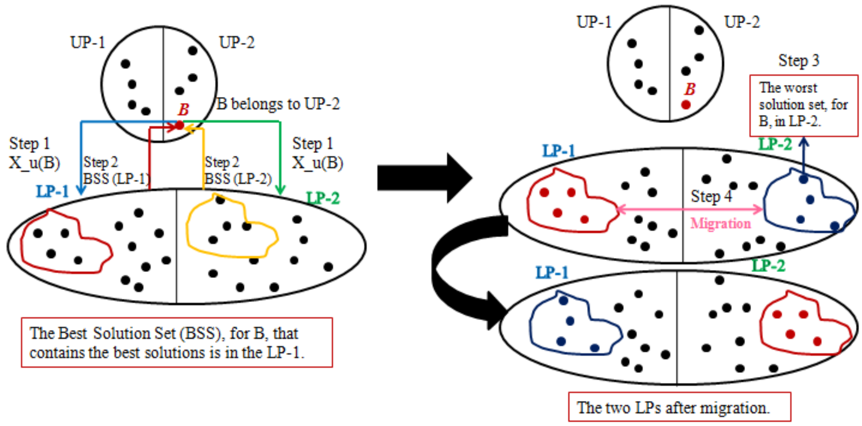


FIGURE 4. Illustration of the migration strategy in IB-CEMBA.

indicator contribution from the follower problem, each upper-level population UP_i (with i belongs to {1,2}) executes its algorithm based on the IBEA principle [39]. Indeed, we determine the individual with the smallest fitness value, we remove it, then we update the remaining individuals fitness values until we reach the stopping criterion. After that, we apply the mating selection, and the variation. We mention here that the use of IBEA helps the algorithm to approximate the optimal upper front.

6) **Step 5: Migration strategy (each α generations):** Since we use two leader populations, and two follower populations, the best lower-level solution set could be in the mismatched follower population as illustrated in Fig. 4. For this reason, we apply a migration strategy after a fixed number of generations. Thus, we use a β parameter in order to select a solution set which contains β solutions from the follower objective space. This selected solution set is used in the migration step. In other words, we determine for each upper-level solution, its best solution set in each LP_i population (with i belongs to {1,2}). Then, we compare the two solution sets, if the best follower solution set for a fixed leader solution belongs to the Mismatched follower Population (*MP*), this solution set is migrated to the Corresponding lower Population (*CP*). To keep balance between the two lower-level populations, we determine the worst follower solution set for the fixed upper-level solution, and we migrate it from *CP* to *MP*.

We mention here that our proposed algorithm is implemented using a multithreading (pseudo-parallel) mechanism in which the two sub-populations work in a parallel manner by the use of multiple threads.

IV. EXPERIMENTAL STUDY

To investigate the performance of our proposed IB-CEMBA, a set of experiments has been performed on a new model of the one proposed by [42]. In this section, we first describe the benchmark problems, the performance metrics, and the algorithms under comparison. Second, we describe the parameter settings, the statistical test method, and the obtained results.

A. RESEARCH QUESTIONS

The objective of this experimental study is to answer to three main questions:

1) **RQ1:** How does IB-CEMBA perform compared to other bi-level optimization algorithms regarding the upper-level?
2) **RQ2:** How does IB-CEMBA perform compared to other bi-level optimization algorithms regarding the lower-level?
3) **RQ3:** How does IB-CEMBA perform compared to other bi-level optimization algorithms regarding the NFEs?

B. BENCHMARK PROBLEMS

The Multi-objective Bilevel Production-Distribution Planning Problem with Equilibrium between Supply and Demand (MBPDPESD) is a problem that describes the relationship between a distribution company, a manufacturing plant, and retailers. As a bi-level problem, the leader is represented by the distribution company, while the manufacturing plant is the follower. MBPDPESD was described by [42] with one objective at the upper-level, and two objectives at the lower-level. Indeed, the manufacturing company is composed from several plants that produce several types of products. These productions are based on the orders which are given by the distribution companies in order to satisfy the retailers demands. For this reason, the production company (follower) has two objectives to be optimized: (1) the minimization of the overall production cost which is the cost of manufacturing the product types in plants, and (2) the minimization of the storage cost of products in plants. For the upper-level, the distribution company (leader) has to serve a number of retailers, geographically dispersed, with their demands. In fact, it needs to purchase products from plants, transport them to their depots, and then, deliver these products to retailers. According to [42], the distribution company has just one objective to be optimized which is the minimization of the transportation cost. In our experimental study, we have chosen to evaluate our IB-CEMBA on a multi-objective bi-level problem that contains two objectives at each level. For this reason, we have proposed a new version of the MBPDPESD [42], that we named MBPDPESDWB (Multi-objective Bilevel Production-Distribution Planning Problem with Equilibrium between Supply and Demand, and Workload Balance) in which we add to the leader problem a second objective. This added leader objective is the workload balance proposed, recently, by [43]. In fact, the ideal case in the previous objective is when the workload is equally distributed in plants and depots, respectively. That means, all the plants are at the same capacity level, and each plant production is proportional to its capacity, while the retailers' demands are almost distributed among all the depots. As following, the distribution company has two objectives in the new MBPDPESDWB problem: (1) the minimization of the transportation cost, and (2) the balance of the total workload assigned to plants and depots. Despite the fact that, to balance the total workload, all depots must be used, it could be desired to use only a depots subset in order to minimize the transportation cost. Table 2 illustrates the MBPDPESDWB notations. The new MBPDPESDWB problem is formulated in Pages 8 and 9. The interaction between the two levels of the proposed model is given by Appendix F. The two levels of the MBPDPESDWB model are given separately. In fact, Equations (2), as shown at the bottom of the page, and (3), as shown at the bottom of the next page, represent the upper-level problem (the distribution company), while Equations (4) and (5), as shown at the bottom of the next page represent the lower-level problem (the manufacturing plant). Indeed, the leader objective functions are given by Equation (2), while its constraints are represented by Equation (3). For the lower-level problem, Equation (4) represents the two follower objective functions, while Equation (5) explains the different constraints for the lower-level problem. We mention here that the feasibility of our proposed model is ensured by the constraint handling technique described in Appendix D.

To experiment the IB-CEMBA efficiency, the compared algorithms were tested by utilizing ten instances which have been previously used in literature in order to solve similar production-distribution planning problems [43], [44]. The description of these instances is given by Table 3 where *NV* represents the number of vehicules and *C* is their capacities.

C. PERFORMANCE METRICS

In order to evaluate the performance of the different algorithms used in our comparison, we choose two performance metrics as follows:

- **The HyperVolume (*HV*)** [45]: The *HV* indicator measures the hypercube volume which is covered by a non-dominated set of solutions noted *Q*. The *HV* indicator expression is given as follows [46]:

$$HV = Volume(\cup_{i=1}^{|Q|} v_i) \tag{6}$$

  where i represents a solution that belongs to the non-dominated set *Q* and it represents the diagonal corners of the hypercube V_i which is constructed with a reference point noted *W*. To compute the *HV* metric value for our bi-objective case, a reference point $W = (w, w)$ is needed. It is worth noting that the reference point value should be properly and precisely defined as it heavily

$$min_x F(x,y) = (F_1(x,y), F_2(x,y)) \begin{cases} F_1(x,y) = \sum_{d\in D}(\sum_{p\in P}\sum_{t\in T} fee_{pdt}\ y_{pdt} + \sum_{r\in R}\sum_{t\in T} s_{drt}\ x_{drt}) + \sum_{d\in D}\sum_{p\in P}\sum_{t\in T} o_{pt}\ y_{pdt} \\ \Rightarrow \text{ It explains the first upper-level objective corresponding to the minimization of the} \\ \text{transportation cost between: (1) retailers and depots, and (2) depots and plants.} \\ F_2(x,y) = BP(y) + \gamma BD(x) \\ \Rightarrow \text{It explains the second upper-level objective that is the workload balance objective.} \\ \text{Such that:} \begin{cases} BP(y) = \max_{p\in P} \left| \dfrac{\prod_{p_i}^{max}}{\sum_{p_i\in P}\prod_{p_i}^{max}} \sum_{r\in R} b_r - \sum_{d\in D} y_{pd} \right| \\ \Rightarrow \text{It computes the maximum deviation for the plants.} \\ BD(x) = \max_{d\in D} \left| \dfrac{\sum_{r_z\in R} b_{r_z}}{|D|} - \sum_{v\in V_d}\sum_{r_j\in R_v}\sum_{r_z\in R_v} x_{dr_j}\ e^{v}_{r_j r_z} \right| \\ \Rightarrow \text{It computes the maximum deviation for the depots.} \\ \gamma \text{ is a numeric value that is used to standardize the units in the sum.} \end{cases} \end{cases} \tag{2}$$

$$
s.t.\begin{cases}
\sum_{d\in D} x_{drt} \geq m_{rt};\ r\in R;\ t\in T\\
\Rightarrow \text{It explains the limit of each retailer' demand: the products of type } t \text{ delivered to retailer } r\\
\ \text{must satisfy the } r \text{ demand.}\\
\sum_{r\in R} c_{dt}\ x_{drt} \leq q_{dt};\ d\in D;\ t\in T\\
\Rightarrow\ \text{It describes the limit of each depots' volume: for the depot } d\text{, the volume occupied by the}\\
\ \text{products cannot exceed the available volume of the depot } l.\\
x_{drt} \geq 0;\ d\in D;\ t\in T;\ r\in R\\
\Rightarrow \text{It explains the non-negativity constraint of the upper-level variables.}\\
\sum_{d\in D} y_{pd} \leq \overset{max}{\underset{p\in P}{\Pi}};\ p\in P\\
\Rightarrow \text{It represents the fact that, for each plant, the produced quantity does not exceed its maximum}\\
\ \text{manufacturing capacity.}\\
\sum_{p\in P} y_{pd} \geq \sum_{v\in V_d}\sum_{r\in R} b_r;\ d\in D\\
\Rightarrow \text{It garantees the fact that the demand of retailers assigned to each depot is satisfied.}\\
\sum_{r\in R_d} e^{v}_{dr} \leq 1;\ v\in V_d;\ d\in D\\
\Rightarrow \text{It represents the retailer assignment to the different depots.}\\
\sum_{d\in D} e^{v}_{d_k d_l} = 0;\ v\in V_d;\ d_k, d_l\in D\\
\Rightarrow \text{It forbids the inter-flow between depots. In other words, the shipments between depots are}\\
\ \text{not allowed.}\\
\sum_{r_j\in R_v} x_{dr} \sum_{r_z\in R_v} e^{v}_{r_j r_z} \leq \overset{max}{\underset{v}{Q}};\ v\in V\\
\Rightarrow \text{It represents the constraints of each vehicle's capacity: the vehicle } v \text{ serving two retailers}\\
\ \text{respects its maximum capacity.}\\
y_{pd} \geq 0;\ p\in P;\ d\in D;\\
\Rightarrow \text{It explains the non-negativity of the manufacturing variables.}
\end{cases}
\tag{3}
$$

$$
min_y f(x,y) = (f_1(x,y), f_2(x,y))\begin{cases}
f_1(x,y) = \sum_{p\in P}\sum_{d\in D}\sum_{t\in T} g_{pt}\ y_{pdt}\\
\Rightarrow\ \text{It explains the first lower-level objective corresponding to the}\\
\ \text{minimization of the production cost.}\\
f_2(x,y) = \sum_{p\in P}(\sum_{t\in T}\sum_{d\in D} w_{pt}\ y_{pdt})\\
\Rightarrow \text{It explains the second lower-level objective corresponding to the}\\
\text{minimization of the storage cost of the product of type } t \text{ in palnt } p.
\end{cases}
\tag{4}
$$

$$
s.t.\begin{cases}
\sum_{t\in T}\sum_{d\in D} n_{pt}\ y_{pdt} \leq h_p;\ p\in P\\
\Rightarrow \text{It explains the volume limit at each plant: the volume used in plant } p\\
\ \text{cannot exceed the } t \text{ limit volume.}\\
\sum_{d\in D}\sum_{p\in P}\sum_{t\in T} y_{pdt} = \sum_{r\in R}\sum_{d\in D}\sum_{t\in T} x_{drt}\\
\Rightarrow\ \text{It explains the equilibrium between the retailers' demands, and the}\\
\text{amount of products manufactured by plants: the total amount of products}\\
\text{for the plant } p \text{ should be delivered to retailers in order to satisfy the}\\
\text{demands.}\\
y_{pdt} \geq 0;\ p\in P;\ d\in D;\ t\in T\\
\Rightarrow \text{It explains the non-negativity constraints of the lower-level variables.}
\end{cases}
\tag{5}
$$

TABLE 2. Notations used in the model.

Symbol	Description	Location
	System actors	
P	The set of plants (where any plant $p \in P$)	lower-level
D	The set of depots (where any depot $d \in D$)	upper-level
R	The set of retailers (where any retailer $r \in R$)	upper-level
T	The set of product types (where any type $t \in T$)	both levels
V	The set of vehicles (where any vehicle $v \in V$)	both levels
V_d	The set of vehicles serving depot d	upper-level
R_v	The set of retailers served by the vehicle v	upper-level
R_d	The set of retailers served by the depot d	upper-level
	Decision variables	
x_{drt}	A leader decision variable indicating the amount of products of type t delivered by depot d to retailer r (such that $d \in D, r \in R, t \in T$)	upper-level
y_{pdt}	A follower decision variable indicating the amount of products of type t produced by plant p to depot d (such that $p \in P, d \in D, t \in T$)	lower-level
e^v_{dr}	A binary variable that takes 1 when the vehicle v serves retailer r from depot d	upper-level
$e^v_{d_k d_l}$	A binary variable that takes 0 to forbid the inter-flow between the two depots d_k and d_l	upper-level
$e^v_{r_j r_z}$	A binary variable that takes 1 when the vehicle v moves from retailer r_j to retailer r_z	upper-level
	Objective functions (to be minimized)	
F_1	The transportation cost	upper-level
F_2	The balance corresponding to the sum of BP and BD where: -BP is the maximum deviation from the mean workload for plants -BD is the maximum deviation from the mean workload for depots	upper-level
f_1	The production cost	lower-level
f_2	The cost of products storage in plants	lower-level
	Constraints' variables	
m_{rt}	The demand amount in terms of product t from retailer r	upper-level
c_{dt}	The amount of volume used by a unit of the product of type t at the depot d	upper-level
q_{dt}	The available volume at the depot d to store the product of type t	upper-level
b_r	The demand of the retailer r	upper-level
h_p	The available storage volume at the plant p	lower-level
n_{pt}	The volume needed for the storage of a unit of the product of type t in the plant p	lower-level
	Constants	
$\pi^{max}_{p_i}$	The maximum manufacturing capacity of the plant p_i	lower-level
$Q^{max}_{v_a}$	The maximum capacity of the vehicle v_a	both levels
fee_{pdt}	The transportation cost of delivering the product of type t from the plant p to the depot d	both levels
s_{drt}	The transportation cost of delivering the product of type t from depot d to retailer r	upper-level
o_{pt}	The price of a unit of the product of type t in the plant p	lower-level
g_{pt}	The production cost of manufacturing a unit of the product of type t at the plant p	lower-level
w_{pt}	The storage cost for a unit of the product of type t in the plant p	lower-level
	Indices	
k, l	Two indices used to indicate the k^{th} depot and l^{th} one, respectively	upper-level
j, z	Two indices used to indicate the j^{th} retailer and z^{th} one, respectively	upper-level
i	The index used to mention the i^{th} plant	lower-level

TABLE 3. Description of the MBPDPESDWB benchmarks.

Instance	Retailer	Depot	Plant	(*NV*, *C*)	Product Type
1	48	4	4	(5, 100)	1
2	96	4	4	(5, 180)	2
3	144	4	4	(5, 185)	3
4	192	4	4	(5, 190)	4
5	240	4	4	(5, 195)	5
6	288	4	4	(5, 249)	1
7	72	6	6	(8, 100)	2
8	144	6	6	(8, 180)	3
9	216	6	6	(8, 190)	4
10	288	6	6	(8, 249)	5

influences the outcome of the *HV* computation. For a fair comparison, we have adopted the recently proposed formulation designed by [47]. In such formulation, $w = 1 + (1/H)$ where H is a user-specified parameter. Based on the recommendations in [48], $H = 5$ and then $w = 1.2$ and $W = (1.2, 1.2)$. The larger the obtained *HV* value is, the better the algorithm performance is. We note here that this measure determines both convergence and diversity.

- **The Inverted Generational Distance (*IGD*)** [49]: The *IGD* gives a measure of the distance between the true Pareto front and its closest individual in an approximation set. The *IGD* measures not only the diversity but also the convergence. Let F^* be a set of points that are uniformly distributed over the Pareto Front and S represents the obtained solution set. The *IGD* metric can be formulated as follows [50]:

$$IGD = \frac{\sum_{x^* \in F^*} dist(F^*, S)}{|F^*|} \tag{7}$$

In the previous formulation, $dist(F^*, S)$ represents the Euclidean distance between a point from F^* ($x^* \in F^*$) to its nearest solution in S. We mention here that when the *IGD* values are low, then, better sets are obtained.

D. ALGORITHMS UNDER COMPARISON

To validate the IB-CEMBA performance, we conduct a comparative experimental study between the proposed IB-CEMBA, and the Indicator-Based (IB) extensions of the three BLOPs combinatorial algorithms to which we have previously compared our CEMBA algorithm [28] and to a bi-level algorithm that applies the well-known approach NSGA-II [51]. Indeed, the compared algorithms description is given as follows:

- ***An IB-nested approach***: An indicator-based extension of the simple nested algorithm. This algorithm considers the optimality condition of the lower-level as a constraint, and tries to find best variables of the upper-level.
- ***IB-CoBRA***: An Indicator-Based extension of the Co-evolutionary Bi-level method using Repeated Algorithms [52]. The working principle of CoBRA is based on five components: (1) a co-evolutionary scheme, (2) level-specific selection operators, (3) an archiving strategy, (4) a stopping criterion, and (5) two specific algorithms. This combinatorial algorithm seeks to produce solutions with good quality. Then, it archives the best obtained solutions, applies a selection operator, and co-evolves both sub-populations at the final iteration.
- ***IB-CODBA-CRO***: An Indicator-Based extension of the CO-evolutionary Decomposition-Based Algorithm with Chemical Reaction Optimization [53]. The CODBA-CRO lower-level working principle is based on three main mechanisms: (1) the decomposition, (2) the multi-threading, and (3) the co-evolution scheme. This algorithm decompose a molecule population into several well-distributed sub-populations using a decomposition method. After that, these sub-populations evolve using several threads. Then, each thread (cluster) applies chemical reaction optimization in order to find optimal solutions.
- ***N-NSGA-II***: It is a Nested bi-level algorithm that uses the Non-dominated Sorting Genetic Algorithm II [51] at each level of the problem. In fact, NSGA-II is a multi-objective evolutionary algorithm that uses three main mechanisms: (1) recombination operators to generate offsprings, (2) non-dominated sorting to classify the population into several fronts, and (3) niching strategy in order to choose individuals that have the largest crowded distance in the front. The NSGA-II is a method for diversity and an elitist approach.

E. PARAMETER SETTING AND STATISTICAL TEST METHOD

The parameter values can largely influence the performance of an algorithm, for this reason, a good way for tuning parameters must be chosen [54]. On this subject, the different parameter settings used for each algorithm are detailed in the following. We must mention here that a trial-and-error [54] strategy-based Taguchi method [55] for tuning parameters values is used (cf. details in Appendix E). In Table 4, the design parameters for algorithms under comparison are illustrated. In order to achieve more accurate and stable results, the same terminal criterions are considered. When we are testing the *HV* values, the *IGD* values, and the CPU time, we use 5.12 E^{+6} NFEs as a stopping criterion. For the NFEs test, we use, for each instance, the upper reference *HV* value as a stopping criterion. To provide a making quantitative decisions system about the process, a statistical test is needed. For this reason, we used in a pairwise fashion the Wilcoxon rank test [56] in order to check whether there is a statistical difference between the obtained samples results. In fact, the choice of a Wilcoxon rank test is explained by the fact that this test works on the values' ranks and not on the values themselves. In this experimental study, 31 runs were launched for each couple (algorithm, problem) as recommended in [56]. After that, the obtained results were analyzed using the median value as a measure of central tendency. In fact, the median value represents the middle score for a data set, and it is not distorted by the skewed data and the outliers [57]. As follows, the *p-value* of the 2 following hypothesis is computed using the MATLAB rank sum function:

- *H0*: median (Algorithm 1) = median (Algorithm 2).
- *H1*: median (Algorithm 1) $\neq$ median (Algorithm 2).

In fact, we talk about two different cases. In the first case, the obtained *p-value* is equal or less than 0.05, and here *H1* is accepted, while *H0* is rejected. So that, the two algorithms median are different from each other. In the second case, the obtained *p-value* is greater than 0.05, and here *H1* is rejected, while *H0* is accepted, and we cannot say that one of the two algorithms is better than the other. We note here that we have coded our IB-CEMBA and the other used

TABLE 4. Default parameters settings.

Specific parameters	
IB-CEMBA	Upper Population size: UP_1 = 40, UP_2 = 40, Lower Population size: LP_1 = 40, LP_2 = 40, Upper and Lower Generations: UG = 40, LG = 40.
IB-CODBA-CRO	Upper Population size: *UP* = 100, Lower Population size: *LP* = 100, Upper and Lower Generations: *UG* = 26, *LG* = 20. (The 26^{th} upper generation will be interrupted once we arrive to 5.12 E^{+6})
IB-CoBRA	Upper Population size: *UP* = 100, Lower Population size: *LP* = 100, Exchange parameter and Upper Generations: *Pg* = 10, *UG* = 256.
IB-Nested	Upper Population size: *UP* = 100, Lower Population size: *LP* = 100, Upper and Lower Generations: *UG* = 26, *LG* = 20. (The 26^{th} upper generation will be interrupted once we arrive to 5.12 E^{+6})
N-NSGA-II	Upper Population size: *UP* = 100, Lower Population size: *LP* = 100, Upper and Lower Generations: *UG* = 26, *LG* = 20. (The 26^{th} upper generation will be interrupted once we arrive to 5.12 E^{+6})
Commun parameters	
Crossover	Type: RBX crossover, Probability = 0.9.
Mutation	Type: Uniform mutation, Probability = 0.1.
DSDM method	Decomposition parameter = 2.

TABLE 5. Median *HV* values for IB-CEMBA, IB-CODBA, IB-CoBRA, IB-nested algorithm, and N-NSGA-II on MBPDPESDWB benchmarks.

Instance	IB-CEMBA	IB-CODBA-CRO	IB-CoBRA	IB-nested	N-NSGA-II
1	**0.971(++++)**	0.942(+++)	0.871(++)	0.798(-)	0.790
2	**0.950(++++)**	0.890(+++)	0.856(++)	0.785(+)	0.778
3	**0.941(++++)**	0.887(+++)	0.840(++)	0.771(+)	0.768
4	**0.928(++++)**	0.856(+++)	0.811(++)	0.758(+)	0.748
5	**0.917(++++)**	0.831(+++)	0.809(++)	0.737(+)	0.731
6	**0.904(++++)**	0.827(+++)	0.795(++)	0.710(+)	0.704
7	**0.899(++++)**	0.805(+++)	0.783(++)	0.693(+)	0.689
8	**0.892(++++)**	0.800(+++)	0.775(+-)	0.687(+)	0.685
9	0.879(++++)	**0.882(+++)**	0.760(++)	0.624(+)	0.660
10	**0.850(++++)**	0.799(+++)	0.710(++)	0.595(+)	0.591

TABLE 6. Median *IGD* values for IB-CEMBA, IB-CODBA, IB-CoBRA, IB-nested algorithm, and N-NSGA-II on MBPDPESDWB benchmarks.

Instance	IB-CEMBA	IB-CODBA-CRO	IB-CoBRA	IB-nested	N-NSGA-II
1	**0.0072(++++)**	0.0088(+++)	0.0102(++)	0.0121(+)	0.0160
2	**0.0099(++++)**	0.0097(+++)	0.0136(++)	0.0157(-)	0.0171
3	**0.0311(++++)**	0.0620(+++)	0.0702(++)	0.0887(+)	0.0913
4	**0.0460(++++)**	0.0659(+++)	0.0751(++)	0.0901(+)	0.0974
5	**0.0482(++++)**	0.0673(+++)	0.0784(++)	0.0922(+)	0.0992
6	**0.0720(++++)**	0.0911(+++)	0.0995(++)	0.1128(+)	0.1204
7	**0.0829(++++)**	0.1073(+++)	0.1230(++)	0.1530(+)	0.1776
8	**0.1275(++++)**	0.1712(+++)	0.1912(++)	0.2134(+)	0.2284
9	**0.2111(++++)**	0.2536(+++)	0.2845(++)	0.3001(-)	0.3131
10	**0.2420(++++)**	0.2907(+++)	0.3026(++)	0.3210(+)	0.3354

algorithms in Java programming langage, while we have performed the different simulations on the same machine: Intel(R) Core(TM) i5-4210 CPU 2.40GHz,4 GO RAM. The comparison results are represented by (+) (significance) and (−) (no significance). We note here that the (+) symbol indicates that *H0* is rejected while the (−) symbol means that *H0* is accepted.

F. COMPARATIVE UPPER AND LOWER RESULTS

To answer *RQ1*, we detail the IB-CEMBA performance along with four competitive algorithms. In our experimental study, the upper-level performance is tested using the *HV* and the *IGD* metrics. Table 5 illustrates the normalized median *HV* values obtained by the five compared algorithms. We mention here that we use 5.12 E^{+6} NFEs as a stopping criterion. Compared to the indicator-based extensions of CODBA-CRO, CoBRA, and the nested approach, and to the N-NSGA-II algorithm, IB-CEMBA has the best overall performance in terms of *HV* leader values. Indeed, in nine instances out of ten ones, our proposed IB-CEMBA reaches the best *HV* values, and generates best pareto-fronts regarding the used algorithms. For the other instance, IB-CEMBA generates the second best *HV* value. The previous results are confirmed by Table 6 that illustrates the *IGD* values for all algorithms under comparison. We mention here that the *IGD* metric requires a true PF in the calculation. For this reason, we have filtered the non-dominated (Pareto-optimal) solutions obtained by all algorithms, during all the executions. The filtering process is done by eliminating dominated solutions and preserving the non-dominated ones which are used as reference PF to compute the *IGD* value in each experiment. As illustrated in Table 6, our proposed approach has the best *IGD* values compared to other approaches. This observation is explained by the fact that IB-CEMBA consists in applying an indicator-based approach at the lower-level in order to choose the solution with the best marginal contribution in terms of the *HV* value from the obtained lower-level PF. This solution will be sent to the upper-level in order to approximate the leader PF. It is worth mentioning that it is not the case for the other approaches since they randomly choose a lower-level solution from the follower PF which is not an optimal choice. We can say that good upper-level results which are obtained by IB-CEMBA are explained by the advantages of using two leader populations with an indicator-based approach at each upper population. In fact, the use of an indicator-based co-evolutionary scheme, helps IB-CEMBA to choose the best solutions. Tables 5 and 6 clearly demonstrate the ability of IB-CEMBA to ensure convergence and diversity of the upper-level PF.

To study the lower-level behaviour of IB-CEMBA, and to answer *RQ2*, we use the *HV* indicator in order to evaluate the lower-level solutions. In the following, Table 7 illustrates the obtained normalized *HV* values. The statistical results are generated using the NFEs as a stopping criterion, which is fixed to 5.12 E^{+6} NFEs. As expected, in all the used test problems, our IB-CEMBA outperforms the four other algorithms in terms of the *HV* values. As shown in table 7, the worst performance is observed with the N-NSGA-II and the indicator-based version of the nested algorithm compared to IB-CODBA-CRO, and IB-CoBRA. This result explains the fact that N-NSGA-II and the IB-nested algorithm need an efficient method for the lower-level problem, so that, it can produce good responses. In the majority of the test problems, IB-CODBA-CRO outperforms IB-CoBRA. Indeed, all these

TABLE 7. Median *HV* lower values for IB-CEMBA, IB-CODBA, IB-CoBRA, IB-nested algorithm, and N-NSGA-II on MBPDPESDWB benchmarks.

Instance	IB-CEMBA	IB-CODBA-CRO	IB-CoBRA	IB-nested	N-NSGA-II
1	**0.984**(++++)	0.920(+++)	0.886(++)	0.799(+)	0.773
2	**0.972**(++++)	0.884(+++)	0.859(++)	0.775(+)	0.569
3	**0.966**(++++)	0.872(+++)	0.851(++)	0.756(+)	0.743
4	**0.951**(++++)	0.860(+++)	0.837(++)	0.741(+)	0.737
5	**0.937**(++++)	0.848(+++)	0.804(++)	0.729(+)	0.720
6	**0.918**(++++)	0.831(+++)	0.787(++)	0.708(+)	0.700
7	**0.909**(++++)	0.814(+++)	0.779(++)	0.692(+)	0.685
8	**0.881**(++++)	0.809(+++)	0.758(++)	0.681(-)	0.674
9	**0.876**(++++)	0.801(+++)	0.731(++)	0.672(+)	0.667
10	**0.853**(++++)	0.759(+++)	0.712(++)	0.603(+)	0.600

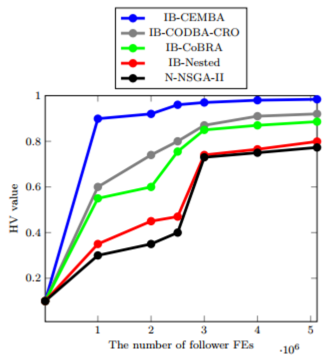


FIGURE 5. The average convergence plots of IB-CEMBA, IB-CODBA-CRO, IB-CoBRA, IB-nested approach, and N-NSGA-II for the instance number 1.

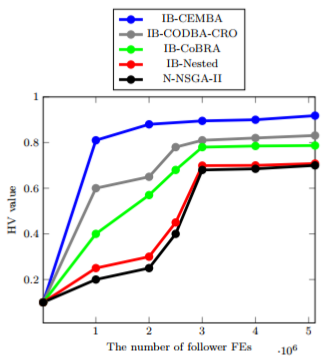


FIGURE 6. The average convergence plots of IB-CEMBA, IB-CODBA-CRO, IB-CoBRA, IB-nested approach, and N-NSGA-II for the instance number 6.

observations show the ability of IB-CEMBA in controlling the directions of search, and approximating the lower-level solution with the maximum marginal contribution in terms of the *HV* metridn summary, the use of a local search procedure at each lower population, helps our proposed IB-CEMBA to generate a single solution instead of a whole Pareto Front, while the used migration scheme guarantees the existence of optimal solutions in the corresponding follower population. In order to clarify more the lower-level informations especially on the convergence characteristic, an average convergence plot on the two lower-level objectives is illustrated using three types of instances: (1) the instance 1 (small size instance) represented by Fig. 5, (2) the instance 6 (medium size instance) represented by Fig. 6, and (3) the instance 10

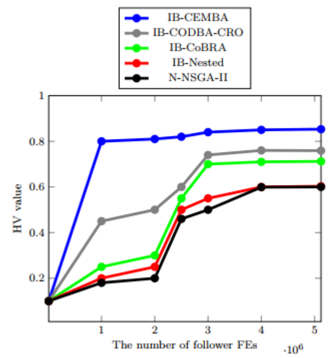


FIGURE 7. The average convergence plots of IB-CEMBA, IB-CODBA-CRO, IB-CoBRA, IB-nested approach, and N-NSGA-II for the instance number 10.

(large size instance) represented by Fig. 7. Indeed, the objective space proportion of the approximation of solutions with maximum marginal contributions in terms of the *HV* indicator is given by each used algorithm for every number of follower evaluations. In fact, by using 5.12 E^{+6} NFEs as a stopping criterion, it is shown that our proposed IB-CEMBA performs better on the global approximation.

G. EFFICIENCY COMPARATIVE RESULTS

In addition to the previous results, we are interested to investigate the performance of the competitive algorithms in terms of the NFEs in order to answer the *RQ3*. Table 8 illustrates the NFEs consumed by IB-CEMBA, IB-CODBA-CRO, IB-CoBRA, IB-nested approach, and N-NSGA-II. We mention here that, for each used instance, we use two different references. Indeed, the first one is the difficult (d) reference upper *HV* value ($HV = 0.8$), and the second one is the easy (e) reference upper *HV* value ($HV = 0.6$). In fact, for each instance, we compute the NFEs consumed by each algorithm in order to reach the difficult, and the easy *HV* values. On the used test problems, we observe that our proposed IB-CEMBA consumes less NFEs than IB-CODBA-CRO, IB-CoBRA, IB-nested approach, and N-NSGA-II. All these results are confirmed by Figs 8, 9, and 10, which describe the NFEs obtained by the confronted algorithms. In the NFEs experiments, we have chosen for brevity three test problems, which are: the instance number 1 (small size instance), the instance number 6 (medium size instance), and

TABLE 8. NFEs for IB-CEMBA, IB-CODBA, IB-CoBRA, IB-nested algorithm, and N-NSGA-II on MBPDPESDWB benchmarks.

Instance	IB-CEMBA ($\times 10^6$)	IB-CODBA-CRO ($\times 10^6$)	IB-CoBRA ($\times 10^6$)	IB-nested ($\times 10^6$)	N-NSGA-II ($\times 10^6$)	Reference HV value
1-d	**6.01(++++)**	6.84(+++)	8.27(++)	10.09(+)	10.70	0.8
1-e	**4.15(++++)**	5.29(+++)	6.03(++)	8.31(+)	8.82	0.6
2-d	**5.41(++++)**	7.05(+++)	7.21(++)	10.24(+)	10.56	0.8
2-e	**3.54(++++)**	5.21(+++)	5.65(++)	6.71(+)	6.98	0.6
3-d	**4.02(++++)**	6.22(+++)	6.98(++)	7.80(+)	8.12	0.8
3-e	**2.81(++++)**	3.74(+++)	3.92(++)	4.21(+)	4.71	0.6
4-d	**5.00(++++)**	5.98(+++)	6.03(++)	6.97(+)	7.36	0.8
4-e	**2.52(++++)**	3.21(+++)	3.93(++)	5.18(+)	5.68	0.6
5-d	**3.92(++++)**	4.87(+++)	5.36(++)	5.90(+)	6.34	0.8
5-e	**2.09(++++)**	2.98(+++)	3.65(++)	4.20(+)	4.84	0.6
6-d	**5.84(++++)**	7.52(+++)	7.92(++)	10.33(+)	10.91	0.8
6-e	**4.27(++++)**	5.63(+++)	5.77(++)	6.84(+)	7.03	0.6
7-d	**4.92(++++)**	5.96(+++)	6.99(++)	7.57(+)	7.99	0.8
7-e	**3.95(++++)**	4.54(+++)	5.11(++)	6.01(+)	6.32	0.6
8-d	**5.03(++++)**	5.57(+++)	7.20(++)	7.72(+)	8.01	0.8
8-e	**4.05(++++)**	4.60(+++)	5.41(++)	6.36(+)	6.77	0.6
9-d	**2.52(++++)**	3.88(+++)	4.01(++)	4.30(+)	4.84	0.8
9-e	**1.95(++++)**	2.39(+++)	2.97(++)	3.50(+)	3.90	0.6
10-d	**1.26(++++)**	2.09(+++)	2.57(++)	3.62(+)	4.03	0.8
10-e	**1.01(++++)**	2.02(+++)	2.13(++)	3.03(+)	3.27	0.6

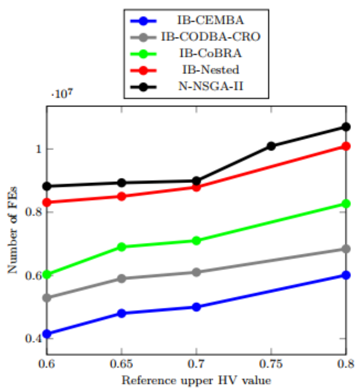


FIGURE 8. The NFEs obtained by all algorithms on the instance 1.

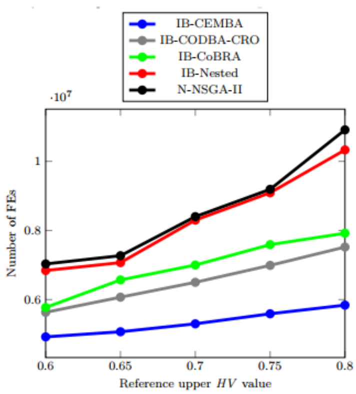


FIGURE 9. The NFEs obtained by all algorithms on the instance 6.

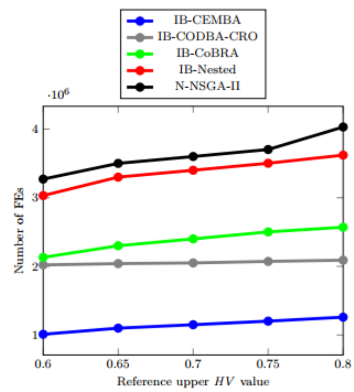


FIGURE 10. The NFEs obtained by all algorithms on the instance 10.

the instance number 10 (large size instance) in which the reference upper HV value is used as a termination criterion. We explain the obtained results by the ability of IB-CEMBA in controlling the directions of search, which can hugely reduce the NFEs. In fact, the decomposition idea that consists in using two upper populations and two lower populations where each UP_i (with i belongs to {1,2}) works with its corresponding LP_i helps IB-CEMBA to evaluate solutions with a minimum NFEs. In summary, IB-CEMBA with the co-evolutionary scheme, and the use of two populations at each level is able to consume a minimum NFEs.

In addition to the previous results, we need to check the ability of IB-CEMBA with migration scheme in reducing the NFEs. For this reason, we have compared two versions of IB-CEMBA in terms of NFEs: (1) the first version is when IB-CEMBA applies a migration scheme and (2) the second one is when IB-CEMBA does not use a migration scheme. In fact, we have chosen for brevity six instances: (1) two small size instances (1 and 2), (2) two medium size instances (3 and 8), and (3) two large size ones (9 and 10). We mention here that we use the upper HV values as a stopping criterion for each experiment. In fact, for each instance, we use two upper-level HV references: (1) an easy upper HV value noted e ($HV = 0.6$) which is easy to be achieved and (2) a difficult upper HV value noted d ($HV = 0.8$) that

TABLE 9. Added value of the migration strategy in terms of the reduction percentage of the NFEs on six instances using two reference upper *HV* values (*d* means a difficult reference and *e* means an easy reference).

Experiment		NFEs for IB-CEMBA without migration ($\times 10^6$)	NFEs for IB-CEMBA with migration ($\times 10^6$)	Reduction Percentage (*RP*)
Instance	**Reference upper *HV* value**			
1-d	0.8	9.98(+)	6.01	39.77 %
1-e	0.6	8.20(+)	4.15	49.39 %
2-d	0.8	10.00(+)	5.41	45.90 %
2-e	0.6	6.52(+)	3.54	45.70 %
3-d	0.8	7.50(+)	4.02	46.40 %
3-e	0.6	4.19(+)	2.81	32.93 %
8-d	0.8	7.20(+)	5.03	30.13 %
8-e	0.6	6.01(+)	4.05	32.61 %
9-d	0.8	4.00(+)	2.52	37.00 %
9-e	0.6	3.05(+)	1.95	36.06 %
10-d	0.8	2.50(+)	1.26	49.60 %
10-e	0.6	2.20(+)	1.01	54.09 %

TABLE 10. Computational time CPU (milliseconds) consumed by IB-CEMBA, IB-CODBA, IB-CoBRA, IB-nested algorithm, and N-NSGA-II on MBPDPESDWB benchmarks.

Instance	IB-CEMBA	IB-CODBA-CRO	IB-CoBRA	IB-nested	N-NSGA-II
1	**10040**(++++)	11331(+++)	20510(++)	26211(+)	26410
2	**9198**(++++)	9879(+++)	14216(++)	31312(+)	31910
3	**11477**(++++)	11610(+++)	25411(++)	25923(+)	26214
4	**14001**(++++)	14123(+++)	24104(++)	34504(+)	34923
5	22247(++++)	**22237**(+++)	31399(++)	34500(+)	34950
6	**19288**(++++)	19411(+++)	28506(++)	30784(+)	31116
7	27930(++++)	**27924**(+++)	43117(++)	48506(+)	49098
8	**29248**(++++)	29944(+++)	45603(++)	72099(+)	72511
9	**29037**(++++)	29610(+++)	48899(++)	63213(+)	63832
10	**29874**(++++)	30642(+++)	51711(++)	69266(+)	69707

is difficult to be achieved. Thus, twelve experiments were performed ((six instances) $\times$ (two *HV* reference values)) to test the impact of using a migration strategy in the reduction of NFEs. Table 9 illustrates the obtained NFEs for the two versions of IB-CEMBA. To show the impact of the migration scheme, we have calculated the Reduction Ratio (*RR*) and the Reduction Percentage (*RP*) values as follows:

$$RR = \frac{\text{NFEs without migration - NFEs with migration}}{\text{NFEs without migration}} \quad (8)$$

$$RR = 1 - \left(\frac{\text{NFEs with migration}}{\text{NFEs without migration}}\right) \quad (9)$$

$$RP = RR \times 100 \quad (10)$$

As illustrated in Table 9, IB-CEMBA with the use of a migration scheme has a good reduction of the NFEs. In fact, the obtained results show that the *RP* ranges vary in the interval [30.13%, 54.09%]. We can say that this reduction is very significant and important because it varies between the third and almost the half.

To further emphasize our experimental study, Table 10 reports the CPU times obtained at each instance for IB-CEMBA, IB-CODBA, IB-CoBRA, IB-nested approach, and N-NSGA-II. We note here that the NFEs is used as a stopping criterion for all the used algorithms. Indeed, the commun NFEs is set to 5.12 E^{+6}. As shown in the table, our proposed IB-CEMBA consumes less CPU time compared to IB-CoBRA, IB-nested algorithm, and N-NSGA-II. However, the CPU times consumed by IB-CEMBA, and IB-CODBA-CRO are close to each other. In fact, the obtained results of IB-CEMBA, and IB-CODBA-CRO demonstrate that the fact of using a pseudo-parallel mechanism, helps these two algorithms to not waste computational effort while determining the pareto-optimal sets, which is not the case for the other algorithms.

V. CASE STUDY: MULTI-OBJECTIVE BI-LEVEL FEATURE CONSTRUCTION FOR THE CASE OF BINARY CLASSIFICATION

The case study discussed here represents the multi-objective bi-level feature construction problem (for the binary classification case). In fact, the bi-level feature construction problem has been tackled only in a single-objective way [58]. Inspired by works from the field of evolutionary feature selection and construction [59], [60], we developed, in this paper, the multi-objective version of the bi-level feature construction problem. In fact, feature selection and construction are important techniques in data mining. Indeed, feature selection consists in selecting a relevant feature subset, while feature construction consists in generating new high-level features (constructed features), where each one of them corresponds to a combination of an original features subset. To reduce the dimensionality and improve the classification performance, feature selection and feature construction are used together [59]. As described in Fig. 11, in a multi-objective bi-level feature construction, the upper-level performs feature selection by minimizing two objectives: (1) the number of

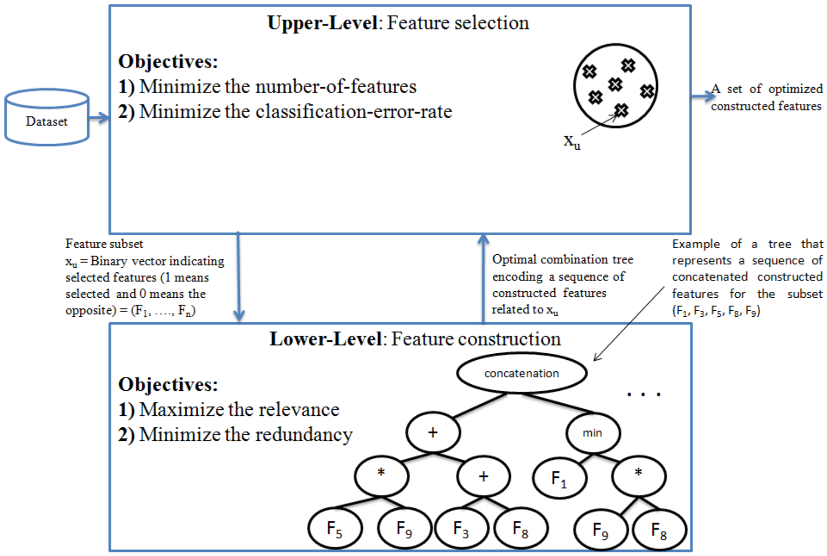


FIGURE 11. Illustrating the multi-objective bi-level feature construction problem.

TABLE 11. Datasets.

Datasets	#Features	#Instances	#Classes
Colon Tumor	2000	60	2
Central Nervous System	7129	60	2

selected features and (2) the classification-error-rate; while the lower-level searches for optimized constructed features (combinations of features) by: (1) maximizing relevance and (2) minimizing redundancy. In the following, each upper-level solution (feature subset represented by a binary vector indicating selected features where "1" means that the corresponding feature is selected and "0" otherwise) is associated with an optimal lower-level solution (tree encoding a constructed feature that represents optimal combinations of feature subset). Both leader objectives could be conflicting as the minimization of the number of features may increase the classification-error-rate due to the removal of relevant features [60]. A similar observation could be seen for the follower objectives as maximizing the relevance (correlation with class labels) may increase the redundancy between the constructed features of the considered combination tree.

Hence, our problem is a combinatorial MOPB with two (possibly) conflicting objectives at the upper-level and two other ones at the lower-level. It is worth noting that for each feature subset, a whole lower-level is optimized in order to find the corresponding tree that encodes a constructed feature. In this way, the bi-level problem will be able to generate a set of optimized constructed features.

To show the versatility of IB-CEMBA, we have used, for brevity, two large-dimensional gene expression datasets described by Table 11 and available at `http://csse.szu.edu.cn/staff/zhuzx/Datasets.html`. The performance of IB-CEMBA is evaluated by comparing it with IB-CODBA-CRO, IB-CoBRA, IB-nested, and N-NSGA-II based on the obtained constructed features. The experiments were conducted using two types of machine learning algorithms: (1) K-Nearest Neighbors (KNN) and (2) Naïve Bayes (NB). To show the effectiveness of our proposed approach, we use two performance metrics: (1) classification accuracy metric and (2) classification-error-rate one. We mention here that the stopping criterion is fixed to 5.12 E^{+6} NFEs. In Table 12, we compare the classification accuracy of the constructed features obtained by IB-CEMBA with those

TABLE 12. Classification accuracy results of the obtained constructed features.

Dataset	Algorithm	NOF	KNN		NB	
			Best	Median	Best	Median
Colon Tumor	IB-CEMBA	27	**90.01(++++)**	**79.03(++++)**	**84.83(++++)**	**82.94(++++)**
	IB-CODBA-CRO	26	87.35(+++)	77.95(+++)	83.50(+++)	78.57(+++)
	IB-CoBRA	29	86.50(++)	77.32(+-)	82.91(+-)	75.39(++)
	IB-nested	32	85.04(+)	76.84(+)	81.36(-)	73.36(+)
	N-NSGA-II	35	84.95	76.41	81.23	73.21
Central Nervous System	IB-CEMBA	51	**89.65(++++)**	**89.42(++++)**	**98.92(++++)**	**94.31(++++)**
	IB-CODBA-CRO	53	88.05(+++)	69.0(+++)	96.30(+++)	70.09(+++)
	IB-CoBRA	48	77.20(++)	65.21(++)	70.02(++)	62.22(++)
	IB-nested	60	60.21(-)	55.34(+)	61.59(+)	53.14(-)
	N-NSGA-II	63	55.20	54.35	61.46	53.01

TABLE 13. Classification-error-rate results of the obtained constructed features.

Dataset	Algorithm	NOF	KNN		NB	
			Best	Median	Best	Median
Colon Tumor	IB-CEMBA	27	**0.0999(++++)**	**0.2097(++++)**	**0.1517(++++)**	**0.1706(++++)**
	IB-CODBA-CRO	26	0.1265(+++)	0.2205(+++)	0.1650(+++)	0.2143(+++)
	IB-CoBRA	29	0.1350(++)	0.2268(+-)	0.1709(+-)	0.2461(++)
	IB-nested	32	0.1496(+)	0.2316(+)	0.1864(-)	0.2664(+)
	N-NSGA-II	35	0.1505	0.2359	0.1877	0.2679
Central Nervous System	IB-CEMBA	51	**0.1035(++++)**	**0.1058(++++)**	**0.0108(++++)**	**0.0569(++++)**
	IB-CODBA-CRO	53	0.1195(+++)	0.3093(+++)	0.0370(+++)	0.2991(+++)
	IB-CoBRA	48	0.2280(++)	0.3479(++)	0.2998(++)	0.3778(++)
	IB-nested	60	0.3979(-)	0.4466(+)	0.3841(+)	0.6486(-)
	N-NSGA-II	63	0.4480	0.4565	0.3854	0.4699

produced by IB-CODBA-CRO, IB-CoBRA, IB-nested, and N-NSGA-II. Table 13 displays the obtained constructed features for all algorithms in terms of the classification-error-rate metric. In fact, the Number of Original Features is given by column NOF, while the best and median results obtained by KNN and NB are illustrated by columns "Best" and "Median", respectively. We mention here that the NOF is the same for the two tables. Indeed, we are using two tables separately in order to have an idea about the classification accuracy and the classification error-rate. Compared with IB-CODBA-CRO, IB-CoBRA, IB-nested, and N-NSGA-II, the constructed features by IB-CEMBA have higher performance on all datasets in terms of classification-error-rate and classification accuracy. We can say that constructed features by IB-CEMBA help the two classification algorithms obtain higher improvement in all datasets due to the decomposition scheme and the migration strategy.

VI. CONCLUSION AND FUTURE WORKS

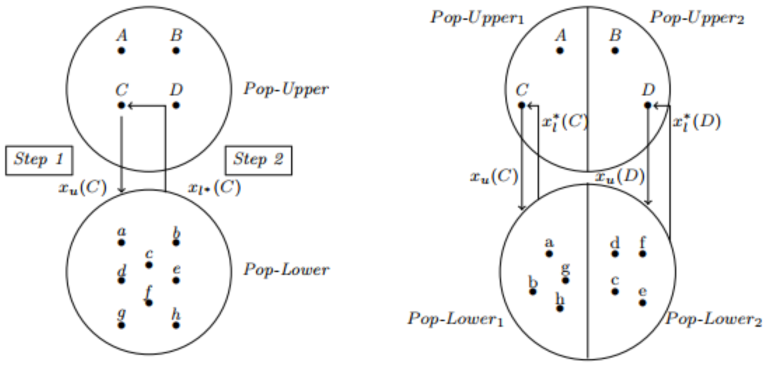
In this paper, we have proposed a new indicator-based co-evolutionary migration-based algorithm for combinatorial multi-objective BLOPs. Due to the lack of works proposed for multi-objective bi-level problems, we have compared our proposed IB-CEMBA to the extensions of three recently proposed combinatorial bi-level algorithms and to a nested bi-level algorithm that uses NSGA-II at both levels. The obtained experimental results show that our proposed IB-CEMBA algorithm provides competitive results regarding the used algorithms. In this section, we would like to discuss the possible threats to validity. These threats could be classified into three different categories: (1) construct validity, (2) internal validity, and (3) external one. We mention here that necessity to construct validity appears because we considered the *HV* performance metric [45] and the *IGD* one [49], we did not use other metrics for the evaluation. As follows, we plan to use additional performance metrics to evaluate the performance of the proposed algorithm. The stochastic behavior of our proposed algorithm represents the major internal threats. In our experimental study, we used the trial-and-error strategy [45] based Taguchi method [55] to set the parameter tuning of our proposed IB-CEMBA. In fact, the obtained results are promising, however, the design of a parameter control strategy for our IB-CEMBA would be a challenging perspective. Concerning the generalizability of the obtained results which refers to the external threats, we have used a set of benchmark test problems that presents various challenging difficulties. In fact, IB-CEMBA has the best overall performance with respect to the compared algorithms. However, it would be interesting to assess IB-CEMBA on other benchmark problems.

APPENDIX A
A COMPARISON BETWEEN THE NESTED APPROACH AND THE CEMBA APPROACH

This appendix explains the difference between a simple nested approach and our proposed CEMBA [28] as illustrated in Fig. 12. Starting by the nested approach which is explained by Fig. 12(a), the fitness evaluation of the upper-level solution C necessitates approximating its corresponding optimal lower-level solution noted x_l^* (C). In other words, the leader variable subvector $x_u(C)$ is passed as a fixed parameter to the follower problem. Then, the latter faces a new lower-level optimization problem that is parametrized by the received $x_u(C)$. The follower algorithm approximates the lower-level optimal solution x_l^* (C) and then sends this approximation to the leader in order to compute the fitness value of the upper-level solution C (i.e., $F(C) = (x_u(C), x_l^*(C))$). As follows, for the nested approach, we repeat the two steps (step 1, and step 2) for each upper-level solution. However, as illustrated in Fig. 12(b), in our proposed CEMBA [28], these two steps are repeated for each leader population with its corresponding follower population. In fact, *Pop-Lower1* individuals contribute in the evaluation of *Pop-Upper1* ones, while *Pop-Lower2* individuals contribute in the evaluation of *Pop-Upper2* ones.

APPENDIX B
A COMPARISON BETWEEN THE DAS-&-DENNIS AND DSDM METHODS

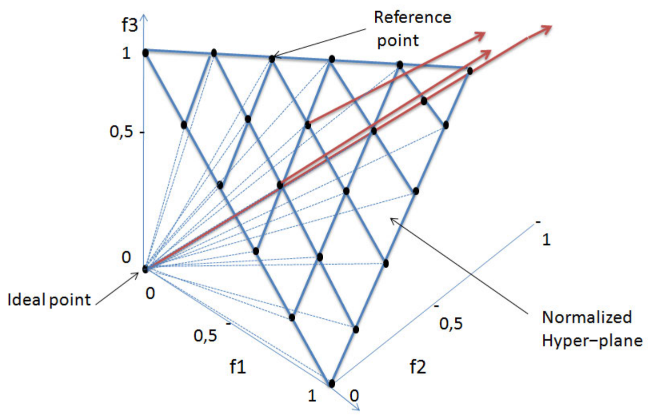
This second appendix explains the difference between the Das and Dennis [61] and the DSDM [41] methods as shown in Fig. 13. For a continuous search space, the Das-&-Dennis method could be used. However, for a discrete search space, the Das-&-Dennis method is inapplicable. The DSDM method is a variant of the Das-&-Dennis method, and it is used in order to generate a set of points over the



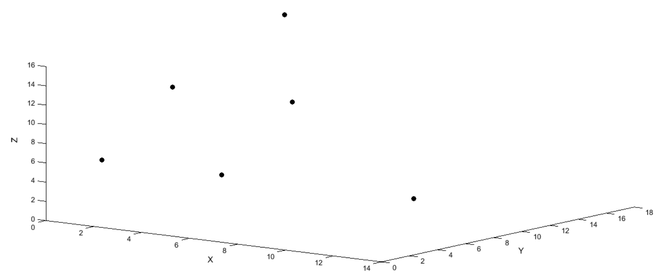
(a) Nested approach: Number of FEs= 4*8 =32

(b) Our CEMBA approach: Number of FEs= (2*4) + (2*4)=16

FIGURE 12. Illustration how our CEMBA decomposition strategy consumes the half of the NFEs of the nested approach.



(a) Illustration of the Das-&-Dennis method with $M = 3$, $P = 5$, and $H = 21$ [61].



(b) Illustration of the DSDM method for three decision variables x_1, x_2, and x_3 where the domains are: D_{x1} = [0,2,5,13], D_{x2} = [4,7,9,17], and D_{x3} = [5,8,11,16] [41].

FIGURE 13. Illustration of the difference between the two decomposition methods: (a) the Das-&-Dennis and (b) the DSDM.

whole decision (search) space that are uniformly distributed as possible. Fig. 13(a) illustrates the distribution of the reference points for the Das-&-Dennis method with 3-objective optimization problem ($M = 3$) and a spacing of $\delta = 0.2$ ($P = 5$). In fact, 21 reference points ($H = 21$) are generated in a normalized hyper-plane. We mention here that lines that are constructed from the origin to each of these reference points represent the reference directions. Fig. 13(b) illustrates the obtained results for the DSDM method with three decision variables where the domains are: $D_{x1} = [0,2,5,13]$, $D_{x2} = [4,7,9,17]$, and $D_{x3} = [5,8,11,16]$. In a discrete space, the DSDM method needs a uniform spacing noted δ_i in order to generate the coordinates of the reference points, this δ_i is calculated for each decision variable as follows: $\delta_i = max_i/P$ (i is the decision variable index, P is a fixed parameter based on the dimension of the problem). For this reason, the DSDM method starts by determining the highest value in each decision variable domain D_{xi} as follows: $max_1 = 13$, $max_2 = 17$, and $max_3 = 16$. After that, the uniform spacing values δ_i are computed (for a $P = 3$) as follows: $\delta_1 = 13/3 = 4$, $\delta_2 = 17/3 = 5$, and $\delta_3 = 16/5 = 5$. We mention here that each axis division is not the same for other axis because the division number varies between axes. Based on the obtained δ_i values, a range of values set (R_i) is generated for each decision variable by performing two steps: (1) the first step consists in adding the first value of D_{xi} to the corresponding R_i, and (2) the second step consists in determining the following members by calculating the sum of: $\delta_i +$ the latest value added to R_i. If the obtained value belongs to D_{xi}, this value will be added to R_i; otherwise, it will be replaced by its closest value that belongs to the corresponding D_{xi}. To clarify the previous idea, we generate the range set R_1 for the corresponding D_{x1} by performing the following steps. First of all, the first value in D_{x1} is added to R_1

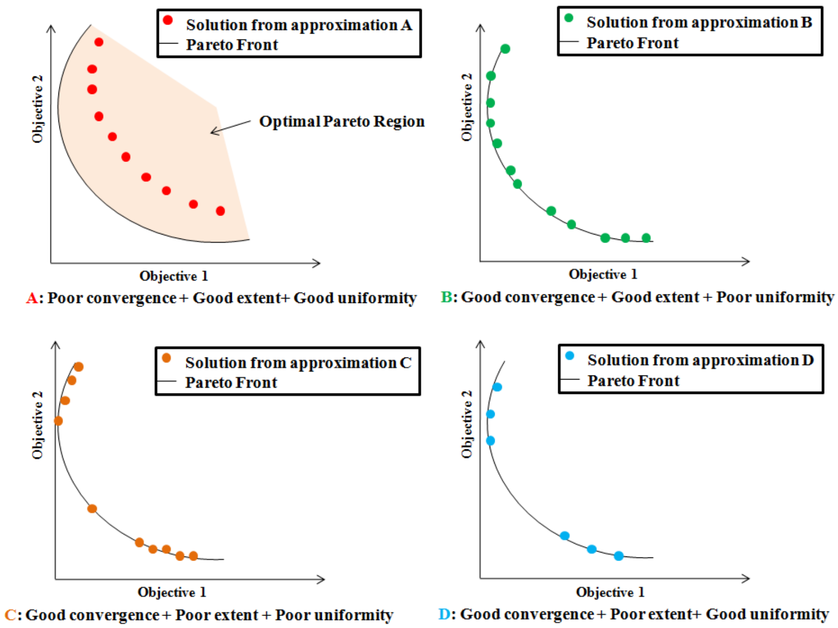


FIGURE 14. Illustration of convergence and diversity concepts in a MOP.

(i.e. $R_1 = [0]$). After that, we determine the following member (i.e. $\delta_1 + 0 = 4$). However, the obtained value does not belong to D_{x1}. For this reason, we replace it by the closest value to 4 that belongs to D_{x1} which is 5 (i.e. $R_1 = [0, 5]$). The same idea is repeated until we reach the upper bound of D_{xi} (i.e. $\delta_1 + 5 = 4 + 5 = 9$ which does not belong to D_{x1}, thus, it is replaced by 13 that represents its closest value in the D_{x1}). At this stage, all the values in R_1 are generated because we have reached 13 which is the upper bound of D_{x1} (i.e. $R_1 = [0, 5, 13]$). Consequently, the coordinates of the generated reference points for D_{x2} and D_{x3} are $R_2 = [4, 9, 17]$, and $R_3 = [5, 11, 16]$, respectively. Finally, the obtained solutions for the DSDM example are: (0,4,5), (0,9,11), (0,17,16), (5,4,5), (5,9,11), and (13,4,5). It is shown from Fig. 13 that the number of solutions generated by the DSDM method is less than the one generated by the Das-&-Dennis method. This observation is explained by the fact that the DSDM method uses the rounding of values because it is applied in a discrete search space. We mention here, that having less number of solutions compared with the Das-&-Dennis method is not important because the goal behind using the decomposition method DSDM is to cover the discrete search space.

APPENDIX C
MULTI-OBJECTIVE OPTIMIZATION

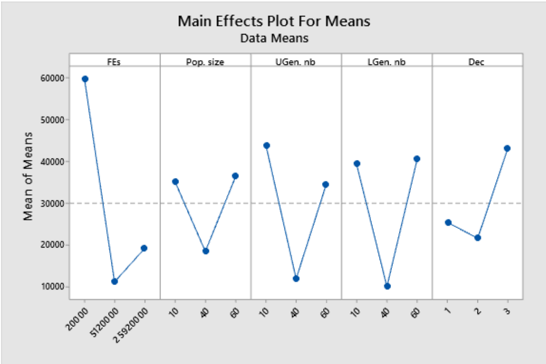
A Multi-Objective optimization Problem (MOP) involves the minimization or the maximization of two or more conflicting objectives simultaneously. In a MOP, we have two goals to pursue which are: (1) convergence towards the optimal PF and (2) diversity along the PF [50]. Indeed, a good convergence means that the obtained non-dominated solutions are very close to the optimal PF. A good diversity is based on two factors: (1) a good uniformity of solution distribution, in other words, the distances separating neighboring solutions are almost equal; and (2) a good extent along the PF, meaning that the obtained non-dominated solutions cover the whole optimal PF. Convergence and diversity concepts are explained by Fig. 14. In the following, four cases are represented: (A) poor convergence + good extent + good uniformity, (B) good convergence + good extent + poor uniformity, (C) good convergence + poor extent + poor uniformity, and (D) good convergence + poor extent + good uniformity. To clarify more the multi-objective aspect, some definitions are given as follows [62]:

Definition 1 (Pareto Optimality): A solution noted x^* is Pareto optimal, if $\forall x \in \Omega$ and $I = \{1, \ldots, M\}$ either $\forall m \in I$, we have: $f_m(x) = f_m(x^*)$ or there is at least one $m \in I$ such that $f_m(x) > f_m(x^*)$.

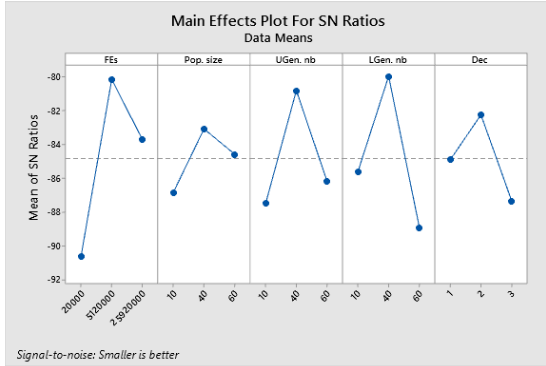
Definition 2 (Pareto Dominance): A solution noted $u = (u_1, \ldots, u_n)$ is said to dominate the solution $v = (v_1, \ldots, v_n)$ (it is denoted by $f(u) \preceq f(v)$) if and only if $f(u)$ is partially less than $f(v)$. In other words, $f_m(u) \leq f_m(v)\ \forall m \in \{1, \ldots, M\}$ and $\exists m \in \{1, \ldots, M\}$ where $f_m(u) < f_m(v)$.

Definition 3 (Pareto Optimal Set): For a given multi-objective optimization problem $f(x)$, the Pareto optimal set is noted P^*: $P^* = \{x \in \Omega | \neg\exists x' \in \Omega, f(x') \preceq f(x)\}$. Where Ω represents the feasible search space, and $f(x') \preceq f(x)$ explains the fact that x' dominates x.

Definition 4 (Pareto Optimal Front): For a given multi-objective optimization problem and its corresponding Pareto optimal set noted P^*, the Pareto Front is PF^*, where $PF^* = \{f(x), x \in P^*\}$.

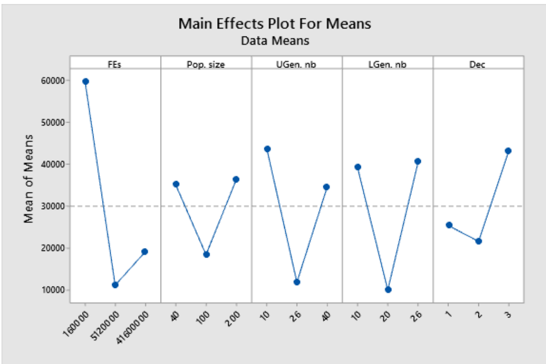


(a) The mean upper-level value for IB-CEMBA.

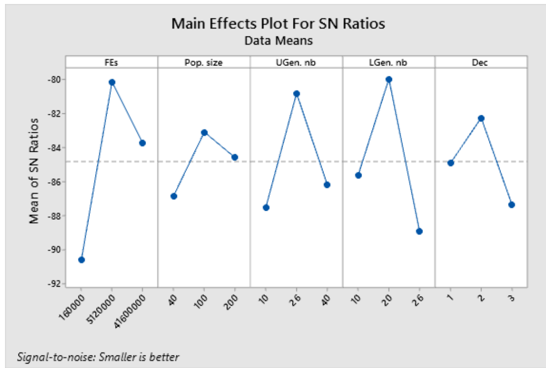


(b) The SNR plot for IB-CEMBA.

FIGURE 15. Results on Taguchi method for parameter tuning of IB-CEMBA.



(a) The mean upper-level value for IB-CODBA-CRO, IB-CoBRA, IB-nested, and N-NSGA-II.



(b) The SNR plot for IB-CODBA-CRO, IB-CoBRA, IB-nested, and N-NSGA-II.

FIGURE 16. Results on Taguchi method for parameter tuning of IB-CODBA-CRO, IB-CoBRA, IB-nested, and N-NSGA-II.

APPENDIX D
THE USED CONSTRAINT HANDLING STRATEGY

This appendix explains the used constraint handling strategy. In fact, we use a constraint handling technique for indicator-based multi-objective evolutionary algorithms which is inspired by [63]. The used constraint handling strategy works as follows:

- **Step 1:** Divide the population P into two sub-populations SP_i (with i belongs to $\{1, 2\}$): (1) a feasible solution set SP_1 (where $SP_1 = \{ p_i \in P \mid$ Constraint Violation $(p_i) = 0\}$), and (2) an infeasible solution set SP_2 (where $SP_2 = \{ p_i \in P \mid$ Constraint Violation $(p_i) > 0\}$).
- **Step 2:** If the number of feasible solutions is equal or greater than the number of individuals N ($|SP_1| \geq N$), then, we apply IBEA [39] in order to select N individuals out of SP_1, we sort the N selected individuals according to their fitness values, and we put the N sorted individuals into P_{t+1}.
  If the number of feasible solutions is less than the number of individuals N ($|SP_1| < N$), then, we preserve all SP_1 individuals, we sort the SP_1 individuals according to their fitness values, and we put the SP_1 sorted individuals into P_{t+1}. After that, we sort the SP_2 individuals according to their constraint violation degree and we preserve the least infeasible (N - $|SP_1|$) solutions in P_{t+1}.

APPENDIX E
THE USED PARAMETER SETTING METHOD

This appendix explains the used method for tuning parameters for all algorithms under comparison. In fact, the Taguchi method [55] is a sophisticated kind of the trial-and-error one [54]. In order to clarify more and verify the proposed parameter tuning values, we have applied the Taguchi method in which the Signal-to-Noise Ratio (SNR) parameter is calculated as follows:

$$SNR = -log_{10}(\frac{1}{N}\sum_{i=1}^{N}(objective\ \ function)_i^2) \qquad (11)$$

where N represents the number of performed runs. The SNR parameter reflects the variability and the mean of the experimental data. The used parameters for tuning are the following: (1) NFEs (*FEs*), (2) population size (*Pop. size*), (3) upper generation number (*UGen. nb*), (4) lower generation number (*LGen. nb*), and (5) decomposition parameter for the DSDM method (*Dec*). The considered levels for each parameter are given by Table 14, while the corresponding orthogonal

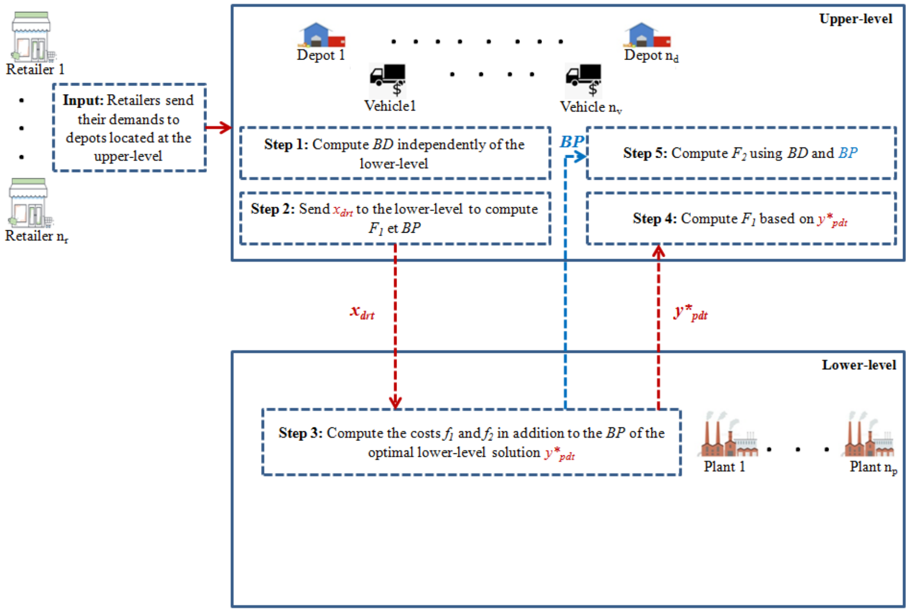


FIGURE 17. The interaction of the lower and the upper levels to evaluate the upper-level objective functions F_1 and F_2 for a single upper-level solution.

TABLE 14. Design parameters and their levels.

IB-CEMBA					
	Parameters				
Levels	A	B	C	D	E
	FEs	Pop. size	UGen. nb	LGen. nb	Dec
Low	20000	10	10	10	1
Medium	5120000	40	40	40	2
Large	25920000	60	60	60	3
IB-CODBA-CRO, BI-CoBRA, IB-nested, and N-NSGA-II					
	Parameters				
Levels	A	B	C	D	E
	FEs	Pop. size	UGen. nb	LGen. nb	Dec
Low	160000	40	10	10	1
Medium	5120000	100	26	20	2
Large	41600000	200	40	26	3

TABLE 15. The orthogonal array L_{27}.

Experiments	A	B	C	D	E
1	A(1)	B(1)	C(1)	D(1)	E(1)
2	A(1)	B(1)	C(1)	D(1)	E(2)
3	A(1)	B(1)	C(1)	D(1)	E(3)
4	A(1)	B(2)	C(2)	D(2)	E(1)
5	A(1)	B(2)	C(2)	D(2)	E(2)
6	A(1)	B(2)	C(2)	D(2)	E(3)
7	A(1)	B(3)	C(3)	D(3)	E(1)
8	A(1)	B(3)	C(3)	D(3)	E(2)
9	A(1)	B(3)	C(3)	D(3)	E(3)
10	A(2)	B(1)	C(2)	D(3)	E(1)
11	A(2)	B(1)	C(2)	D(3)	E(2)
12	A(2)	B(1)	C(2)	D(3)	E(3)
13	A(2)	B(2)	C(3)	D(1)	E(1)
14	A(2)	B(2)	C(3)	D(1)	E(2)
15	A(2)	B(2)	C(3)	D(1)	E(3)
16	A(2)	B(3)	C(1)	D(2)	E(1)
17	A(2)	B(3)	C(1)	D(2)	E(2)
18	A(2)	B(3)	C(1)	D(2)	E(3)
19	A(3)	B(1)	C(3)	D(2)	E(1)
20	A(3)	B(1)	C(3)	D(2)	E(2)
21	A(3)	B(1)	C(3)	D(2)	E(3)
22	A(3)	B(2)	C(1)	D(3)	E(1)
23	A(3)	B(2)	C(1)	D(3)	E(2)
24	A(3)	B(2)	C(1)	D(3)	E(3)
25	A(3)	B(3)	C(2)	D(1)	E(1)
26	A(3)	B(3)	C(2)	D(1)	E(2)
27	A(3)	B(3)	C(2)	D(1)	E(3)

array ($L_{27}(3^5)$ where we have 27 experiments, 5 variables, and 3 levels) is given by Table 15. Fig. 15(b) displays the obtained SNR results for IB-CEMBA. The optimal level with respect to SNR values is: A(2), B(2), C(2), D(2), and E(2). Moreover, Fig. 15(a) displays computed results for IB-CEMBA in terms of mean fitness values of the upper-level in Taguchi experimental analysis, which confirmed the achieved optimal levels using SNR parameter. For the other compared algorithms, Fig. 16(b) displays their optimal SNR values which are: A(2), B(2), C(2), D(2), and E(2). In fact, the computed mean upper-level fitness values (cf. Fig. 16(a)) confirmed the acheived optimal level using SNR parameter.

APPENDIX F
THE INTERACTION OF THE LOWER AND UPPER LEVELS TO EVALUATE THE UPPER-LEVEL OBJECTIVE FUNCTIONS F_1 AND F_2 FOR A SINGLE UPPER-LEVEL SOLUTION

To clarify the interaction of the lower and the upper levels when evaluating the upper-level objective functions F_1 and F_2 for a single upper-level solution, we give details in Fig. 17. We mention here that n_d and n_v represents the depots number and the vehicules number, respectively, while n_r and n_p are the number of retailers and plants, respectively. In a MBPDPESDWB problem, the leader receives demands from a set of retailers. First of all, the leader starts by computing the maximum deviation for depots *BD* located at the upper-level independly from the lower-level. However, the maximum deviation for plants *BP* and the F_1 values depend on the lower-level problem. For this reason, the upper-level sends x_{drt} to the lower-level. At this stage, the follower computes f_1 and f_2 costs in addition to the *BP* value of the optimal lower-level solution y^*_{pdt}. After that, the *BP* value and the y^*_{pdt} solution are sent to the upper-level. In the following, the leader computes F_1 using y^*_{pdt} and F_2 using *BD* and *BP* values. The previous process is applied in order to evaluate the upper-level objective functions F_1 and F_2 for a single upper-level solution.

REFERENCES

[1] A. Migdalas, "Bilevel programming in traffic planning: Models, methods and challenge," *J. Global Optim.*, vol. 7, no. 4, pp. 381–405, Dec. 1995.

[2] H. Sun, Z. Gao, and J. Wu, "A bi-level programming model and solution algorithm for the location of logistics distribution centers," *Appl. Math. Model.*, vol. 32, no. 4, pp. 610–616, Apr. 2008.

[3] C. Kirjner-Neto, E. Polak, and A. Der Kiureghian, "An outer approximations approach to reliability-based optimal design of structures," *J. Optim. Theory Appl.*, vol. 98, no. 1, pp. 1–16, Jul. 1998.

[4] N. Gupta, M. Khosravy, N. Patel, and T. Senjyu, "A bi-level evolutionary optimization for coordinated transmission expansion planning," *IEEE Access*, vol. 6, pp. 48455–48477, 2018.

[5] A. Sinha, P. Malo, A. Frantsev, and K. Deb, "Multi-objective Stackelberg game between a regulating authority and a mining company: A case study in environmental economics," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancun, Mexico, Jun. 2013, pp. 478–485.

[6] R. G. Jeroslow, "The polynomial hierarchy and a simple model for competitive analysis," *Math. Program.*, vol. 32, no. 2, pp. 146–164, Jun. 1985.

[7] J. F. Bard and J. E. Falk, "An explicit solution to the multi-level programming problem," *Comput. Oper. Res.*, vol. 9, no. 1, pp. 77–100, Jan. 1982.

[8] J. F. Bard and J. T. Moore, "A branch and bound algorithm for the bilevel programming problem," *SIAM J. Sci. Stat. Comput.*, vol. 11, no. 2, pp. 281–292, Mar. 1990.

[9] B. Colson, P. Marcotte, and G. Savard, "Bilevel programming: A survey," *4OR*, vol. 3, no. 2, pp. 87–107, Jun. 2005.

[10] X. Li, S. Ma, and Y. Wang, "Multi-population based ensemble mutation method for single objective bilevel optimization problem," *IEEE Access*, vol. 4, pp. 7262–7274, 2016.

[11] A. Koh, "Solving transportation bi-level programs with differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, Sep. 2007, pp. 2243–2250.

[12] E.-G. Talbi, *Metaheuristics for Bi-Level Optimization*, vol. 482. Berlin, Germany: Springer, 2013.

[13] K. Deb and A. Sinha, "Evolutionary bilevel optimization (EBO)," in *Proc. Conf. Companion Genetic Evol. Comput. Companion (GECCO)*, Vancouver BC, Canada, 2014, pp. 857–876.

[14] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: From classical to evolutionary approaches and applications," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 276–295, Apr. 2018.

[15] M. Mokhlesian and S. H. Zegordi, "Application of multidivisional bi-level programming to coordinate pricing and inventory decisions in a multi-product competitive supply chain," *Int. J. Adv. Manuf. Technol.*, vol. 71, nos. 9–12, pp. 1975–1989, Apr. 2014.

[16] A. Sinha, P. Malo, and K. Deb, "Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping," *Eur. J. Oper. Res.*, vol. 257, no. 2, pp. 395–411, Mar. 2017.

[17] M. M. Islam, H. K. Singh, and T. Ray, "A surrogate assisted approach for single-objective bilevel optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 681–696, Oct. 2017.

[18] S. Avraamidou and E. N. Pistikopoulos, "A multi-parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems," *Comput. Chem. Eng.*, vol. 125, pp. 98–113, Jun. 2019.

[19] J. Fliege and L. N. Vicente, "Multicriteria approach to bilevel optimization," *J. Optim. Theory Appl.*, vol. 131, no. 2, pp. 209–225, Dec. 2006.

[20] J. Y. T. Wang, M. Ehrgott, K. N. Dirks, and A. Gupta, "A bilevel multi-objective road pricing model for economic, environmental and health sustainability," *Transp. Res. Procedia*, vol. 3, pp. 393–402, 2014.

[21] J. Gang, Y. Tu, B. Lev, J. Xu, W. Shen, and L. Yao, "A multi-objective bi-level location planning problem for stone industrial parks," *Comput. Oper. Res.*, vol. 56, pp. 8–21, Apr. 2015.

[22] Z. Li, W. Shen, J. Xu, and B. Lev, "Bilevel and multi-objective dynamic construction site layout and security planning," *Autom. Construction*, vol. 57, pp. 1–16, Sep. 2015.

[23] T. Lv, Q. Ai, and Y. Zhao, "A bi-level multi-objective optimal operation of grid-connected microgrids," *Electr. Power Syst. Res.*, vol. 131, pp. 60–70, Feb. 2016.

[24] G. Eichfelder, "Multiobjective bilevel optimization," *Math. Program.*, vol. 123, no. 2, pp. 419–449, Jun. 2010.

[25] A. Sinha, "Bilevel multi-objective optimization problem solving using progressively interactive EMO," in *Proc. Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Ouro Preto, Brazil, 2011, pp. 269–284.

[26] A. Sinha, P. Malo, and K. Deb, "Approximated set-valued mapping approach for handling multiobjective bilevel problems," *Comput. Oper. Res.*, vol. 77, pp. 194–209, Jan. 2017.

[27] K. Deb and A. Sinha, "Solving bilevel multi-objective optimization problems using evolutionary algorithms," in *Proc. Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Nantes, France, 2009, pp. 110–124.

[28] R. Said, S. Bechikh, M. Elarbi, and L. B. Said, "Solving combinatorial bi-level optimization problems using multiple populations and migration schemes," *Oper. Res. Int. J.*, to be published.

[29] S. Bechikh, R. Datta, and A. Gupta, *Recent Advances in Evolutionary Multi-Objective Optimization*, vol. 20. Cham, Switzerland: Springer, 2016.

[30] A. Frantsev, A. Sinha, and P. Malo, "Finding optimal strategies in multi-period Stackelberg games using an evolutionary framework," *IFAC Proc. Volumes*, vol. 45, no. 25, pp. 33–38, 2012.

[31] X. Shi and H. S. Xia, "Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers," *J. Multi-Criteria Decis. Anal.*, vol. 10, no. 1, pp. 27–34, 2001.

[32] Y. Yin, "Genetic-algorithms-based approach for bilevel programming models," *J. Transp. Eng.*, vol. 126, no. 2, pp. 115–120, Mar. 2000.

[33] W. Halter and S. Mostaghim, "Bilevel optimization of multi-component chemical systems using particle swarm optimization," in *Proc. IEEE Int. Conf. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jun. 2006, pp. 1240–1247.

[34] A. Gupta and Y.-S. Ong, "An evolutionary algorithm with adaptive scalarization for multiobjective bilevel programs," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 1636–1642.

[35] W. J. Gutjahr and N. Dzubur, "Bi-objective bilevel optimization of distribution center locations considering user equilibria," *Transp. Res. E, Logistics Transp. Rev.*, vol. 85, pp. 1–22, Jan. 2016.

[36] E. Miandoabchi, F. Daneshzand, W. Y. Szeto, and R. Z. Farahani, "Multi-objective discrete urban road network design," *Comput. Oper. Res.*, vol. 40, no. 10, pp. 2429–2449, Oct. 2013.

[37] H. Zhang, H. Ge, R. Pan, and Y. Wu, "Multi-objective bi-level programming for the energy-aware integration of flexible job shop scheduling and multi-row layout," *Algorithms*, vol. 11, no. 12, p. 210, Dec. 2018.

[38] S. P. Parvasi, R. Tavakkoli-Moghaddam, A. A. Taleizadeh, and M. Soveizy, "A bi-level bi-objective mathematical model for stop location in a school bus routing problem," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 1120–1125, 2019.

[39] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nature (PPSN)*, Birmingham, U.K., 2004, pp. 832–842.

[40] M. Basseur and E. K. Burke, "Indicator-based multi-objective local search," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, Sep. 2007, pp. 3100–3107.

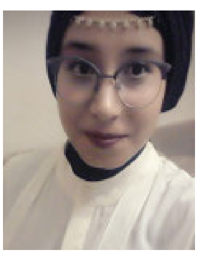
[41] A. Chaabani, S. Bechikh, and L. B. Said, "A new co-evolutionary decomposition-based algorithm for bi-level combinatorial optimization," *Int. J. Speech Technol.*, vol. 48, no. 9, pp. 2847–2872, Sep. 2018.

[42] L. Jia, Y. Wang, and L. Fan, "Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm," *Integr. Comput.-Aided Eng.*, vol. 21, no. 1, pp. 77–90, Jan. 2014.

[43] M.-S. Casas-Ramírez, C.-V. Martha-Selene, J.-F. Camacho-Vallejo, R. G. González-Ramírez, J.-A. Marmolejo-Saucedo, and J.-M. Velarde-Cantú, "Optimizing a biobjective production-distribution planning problem using a GRASP," *Complexity*, vol. 2018, pp. 1–13, Jan. 2018.

[44] H. I. Calvete, C. Galé, and M.-J. Oliveros, "Bilevel model for production–distribution planning solved by using ant colony optimization," *Comput. Oper. Res.*, vol. 38, no. 1, pp. 320–327, Jan. 2011.

[45] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Proc. Int. Conf. Evol. Multi Criterion Optim. (EMO)*, Matsushima, Japan, 2007, pp. 862–876.
[46] J. J. Durillo, Y. Zhang, E. Alba, and A. J. Nebro, "A study of the multi-objective next release problem," in *Proc. 1st Int. Symp. Search Based Softw. Eng. (SSBSE)*, Windsor, U.K., May 2009, pp. 49–58.
[47] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evol. Comput.*, vol. 26, no. 3, pp. 411–440, Sep. 2018.
[48] K. Shang, H. Ishibuchi, and X. Ni, "R2-based hypervolume contribution approximation," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 185–192, Feb. 2020.
[49] C. A. R. Villalobos and C. A. C. Coello, "A new multi-objective evolutionary algorithm based on a performance assessment indicator," in *Proc. 14th Int. Conf. Genetic Evol. Comput. Conf. (GECCO)*, Philadelphia, PA, USA, 2012, pp. 505–512.
[50] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5, 2nd ed. Boston, MA, USA: Springer, 2007.
[51] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[52] F. Legillon, A. Liefooghe, and E.-G. Talbi, "CoBRA: A cooperative coevolutionary algorithm for bi-level optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Brisbane, QLD, Australia, Jun. 2012, pp. 1–8.
[53] A. Chaabani, S. Bechikh, and L. B. Said, "A co-evolutionary decomposition-based chemical reaction algorithm for bi-level combinatorial optimization problems," *Procedia Comput. Sci.*, vol. 112, pp. 780–789, Jan. 2017.
[54] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 19–31, Mar. 2011.
[55] P. J. Ross, *Taguchi Techniques for Quality Engineering: Loss Function, Orthogonal Experiments, Parameter and Tolerance Design*. New York, NY, USA: McGraw-Hill, 1988.
[56] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
[57] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Evanston, IL, USA: Routledge, 1977.
[58] M. Hammami, S. Bechikh, A. Louati, M. Makhlouf, and L. B. Said, "Feature construction as a bi-level optimization problem," in *Proc. Neural Comput. Appl.*, Feb. 2020, pp. 1–22.
[59] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.
[60] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100663.
[61] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, Aug. 1998.
[62] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Hoboken, NJ, USA: Wiley, 2001.
[63] Z.-Z. Liu, Y. Wang, and B.-C. Wang, "Indicator-based constrained multiobjective evolutionary algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Dec. 5, 2019, doi: 10.1109/TSMC.2019.2954491.



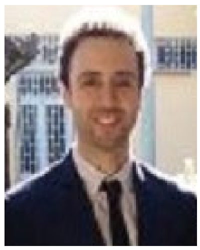
RIHAB SAID received the B.Sc. and M.Sc. degrees in computer science and business from the ISG, University of Tunis, Tunis, Tunisia, in 2015 and 2018, respectively, where she is currently pursuing the Ph.D. degree with the Strategies for Modelling and Artificial inTelligence (SMART) Laboratory.

Her current research interests include bi-level optimization, multiobjective optimization, evolutionary computation, and metaheuristics and their applications.

SLIM BECHIKH received the B.Sc., M.Sc., Ph.D., and Habilitation degrees in computer science and business from the ISG, University of Tunis, Tunis, Tunisia, in 2006, 2008, 2013, and 2015, respectively.

He is currently an Associate Professor with the FSEG, University of Carthage, Nabeul, Tunisia. He is also the Research Director of the Strategies for Modelling and Artificial inTelligence (SMART) Laboratory. He published over 70 articles. His current research interests include multiobjective optimization, evolutionary computation, machine learning, and business analytics.

Dr. Bechikh was a recipient of the Best Paper Award from the ACM SAC, Switzerland, in 2010. He is a Reviewer for several international journals, such as the IEEE Transactions on Evolutionary Computation and the IEEE Transactions on Cybernetics, in addition to several conferences such as the IEEE Congress on Evolutionary Computation and the ACM Genetic and Evolutionary Computation Conference.



ALI LOUATI received the B.Sc. and M.Sc. degrees in computer science and business from the University of Sfax, in 2010 and 2013, respectively, and the Ph.D. degree in computer science and business from the ISG, University of Tunis, Tunis, Tunisia, in 2018.

He held a one-year position of a Researcher with LIMOS, CNRS, UMR, Clermont-Ferrand, France, in 2013 before moving to King Saud University where he spent four years from 2014 to 2018. Since 2018, he has been an Assistant Professor with Prince Sattam Bin Abdulaziz University, Saudi Arabia, and he is a Researcher with the Strategies for Modelling and Artificial inTelligence (SMART) Laboratory, ISG, University of Tunis. He has contributed to several research projects in Tunisia, France, and Saudi Arabia. His main research interests are artificial immune systems, machine learning, data mining, and optimization. He serves as a Reviewer for many international journals such as *European Transportation Research Review*, the *Journal of Supercomputing*, and *Applied Soft Computing*.

ABDULAZIZ ALDAEJ received the M.Sc. and Ph.D. degrees in information systems from the University of Surrey, U.K., in 2010 and 2015, respectively.

He is currently an Assistant Professor with the Department of Information Systems, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Saudi Arabia. His research interests include semantic Web technologies, network security, AI, the IoT, and human–computer interaction.

LAMJED BEN SAID received the B.Sc. degree in computer science and business from the ISG, University of Tunis, Tunis, Tunisia, in 1998, the M.Sc. and Ph.D. degrees in computer science from the University of Paris VI, Paris, France, in 1999 and 2003, respectively, and the Habilitation degree from the ISG, University of Tunis, in 2011.

He is currently a Full Professor and the Dean of the ISG, University of Tunis, where he is also the Head of the Strategies for Modelling and Artificial inTelligence (SMART) Laboratory. He published over 130 articles. His current research interests include multiagent simulation, evolutionary computation, supply chain management, and Industry 4.0. He is a reviewer for several artificial intelligence journals and conferences.