Constructing Test Problems for Bilevel Evolutionary Multi-Objective Optimization

Kalyanmoy Deb and Ankur Sinha
Department of Business Technology
Helsinki School of Economics
PO Box 1210, FIN-00101, Helsinki, Finland
{Kalyanmoy.Deb, Ankur.Sinha}@hse.fi

KanGAL Report Number 2008010

***Abstract*— Many real-world problems demand a feasible solution to satisfy physical equilibrium, stability, or certain properties which require an additional lower level optimization problem to be solved. Although such bilevel problems are studied somewhat in the context of a single objective in each level, there are not many studies in which multiple conflicting objectives are considered in each level. Bilevel multi-objective optimization problems offer additional complexities, as not every lower level Pareto-optimal front has a representative solution to the upper level Pareto-optimal front and that only a tiny fraction of participating lower level fronts make it to the upper level front. A couple of recent studies by the authors have suggested a viable EMO method. In this paper, we analyze the difficulties which a bilevel EMO procedure may face in handling such problems and present a systematic construction procedure for bilevel optimization test problems. Based on the suggested principles, we propose five test problems which are scalable in terms of number of variables and objectives, and which enable researchers to evaluate different phases of a bilevel problem solving task. The test problem construction procedure is interesting and may motivate other researchers to extend the idea to develop further test problems.**

I. Introduction

Bilevel optimization problems appear often in practice due to inherent equilibrium or stability requirements of a physically realizable solution. This requires every feasible solution of the optimization problem (we refer as an 'upper level optimization problem') becomes an optimal solution with respect to a different lower level optimization problem which enforces physical conditions involving equilibrium, stability or flow and energy balance. Despite the existence of this so-called bilevel optimization problems in practice, they are mostly avoided by researchers simply due to the complexities and computational burden associated in solving such problems. Practitioners often derive approximate or mathematical optimality conditions (wherever possible) corresponding to the lower level optimization problem and use the conditions as constraints for the upper level optimization problem. This converts a truly bilevel problem to a single level optimization problem. But there exist many problem scenarios, for which such a conversion is difficult or unacceptable and both level optimization problems must be handled as they are.

In the context of a single objective in each level, there exist studies on mathematical optimality conditions [9] and on numerical algorithm development [2], [16]. However, there are not much studies involving multiple objectives in each level of a bilevel optimization problem. In the context of evolutionary studies on bilevel multi-objective optimization, a previous study emphasized the importance on launching such studies [14]. In the recent past [6], [7], authors have suggested the first EMO-based algorithm for handling bilevel multi-objective optimization problems and demonstrated its working on a few test problems, including a couple borrowed from a previous study which proposed a naive enumerative scheme [10].

In this paper, we identify problem difficulties associated with bilevel problem solving and suggest a systematic procedure for constructing test problems. These problems are scalable in number of variables and can be extended to higher objectives. An extension of the principle can be made to develop higher level test problems as well, but we do not pursue it here. Based on the construction principle, we suggest five specific test problems and discuss why these problems may offer difficulties to an optimization algorithm. We then apply our previously-proposed BLEMO algorithm to solve all five test problems. The study demonstrates that the proposed test problems offer adequate challenges to the existing algorithm and emphasize more research in the development of bilevel multi-objective optimization algorithms.

II. Description of Bilevel Multi-Objective Optimization Problem

A bilevel multi-objective optimization problem has two levels of multi-objective optimization problems such that the optimal solution of the lower level problem determines the feasible space of the upper level optimization problem. In general, the lower level problem is associated with a variable vector $\mathbf{x}_l$ and a fixed vector $\mathbf{x}_u$. However, the upper level problem usually involves all variables $\mathbf{x} = (\mathbf{x}_u, \mathbf{x}_l)$, but we refer here $\mathbf{x}_u$ exclusively as the upper level variable vector. A general bilevel multi-objective optimization problem can be described as follows:

$$\begin{array}{rl} \min._{(\mathbf{x}_u,\mathbf{x}_l)} & \mathbf{F}(\mathbf{x}) = (F_1(\mathbf{x}),\ldots,F_M(\mathbf{x}))\,, \\ \text{s.t.} & \mathbf{x}_l \in \text{argmin}_{(\mathbf{x}_l)} \left\{\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})) \,\middle|\, \right. \\ & \quad \left. \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\right\}, \\ & \mathbf{G}(\mathbf{x}) \geq \mathbf{0}, \mathbf{H}(\mathbf{x}) = \mathbf{0}, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1,\ldots,n. \end{array} \tag{1}$$

In the above formulation, $F_1(\mathbf{x}),\ldots,F_M(\mathbf{x})$ are the upper level objective functions, and $\mathbf{G}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ are upper level inequality and equality constraints, respectively. The objectives $f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})$ are the lower level objective functions, and functions $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are lower level inequality and equality constraints, respectively. It should be noted that the lower level optimization problem is optimized only with respect to the variables $\mathbf{x}_l$ and the variable vector $\mathbf{x}_u$ is kept fixed. The Pareto-optimal solutions of a lower level optimization problem become feasible solutions to the upper level problem. The Pareto-optimal solutions of the upper level problem are determined by objectives $\mathbf{F}$ and constraints $\mathbf{G}$, and restricting the search among the lower level Pareto-optimal solutions.

III. Existing Approaches

Several studies exist in determining the optimality conditions for a Pareto-optimal solution to the upper level problem. The difficulty arises due to the existence of the lower level optimization problems. Usually the KKT conditions of the lower level optimization problems are used as constraints in formulating the KKT conditions of the upper level problem. As discussed in [9], although KKT optimality conditions can be written mathematically, the presence of many lower level Lagrange multipliers and an abstract term involving coderivatives makes the procedure difficult to be applied in practice.

Fliege and Vicente [11] suggested a mapping concept in which a bilevel single-objective optimization problem can be converted to an equivalent four-objective optimization problem with a special cone dominance concept. Although the idea can be, in principle, extended for bilevel multi-objective optimization problems, the number of objectives to be considered is large and moreover handling constraints seems to introduce additional difficulties in obtaining resulting objectives. In the context of bilevel single-objective optimization problems, there exists a number of studies, including some useful reviews [2], [16], test problem generators [1], and even some evolutionary algorithm (EA) studies [15], [13], [18], [12], [17]. However, there does not seem to be too many studies on bilevel multi-objective optimization. A recent study by Eichfelder [10] suggested a refinement based strategy in which the upper level optimization is an exhaustive search. This does not make the algorithm extendible to solving large-dimensional problems.

The greatest challenge in handling bilevel optimization problems seems to lie in the fact that unless a solution is optimal for the lower level problem, it cannot be feasible for the overall problem. This requirement, in some sense disallow any approximate optimization algorithm (including an EA or an EMO) to be used for solving the lower level task. But from all practical point of view near-optimal or near-Pareto-optimal solutions are often acceptable and it is in this spirit for which EA and EMO may have a great potential for solving bilevel optimization problems. EA or EMO has another advantage. Unlike the classical point-by-point approach, EA/EMO uses a population of points in their operation. By keeping two interacting populations, a coevolutionary algorithm can be developed so that instead of a serial and complete optimization of lower level problem for every upper level solution, both upper and lower level optimization tasks can be pursued simultaneously through iterations. In a couple of recent studies by the authors [6], [7], for the first time, a viable bilevel evolutionary multi-objective optimization (BLEMO) algorithm was suggested. The algorithm is applied to a couple of problems from the Eichfelder's [10] study, and successful simulation results are reported.

IV. Principle of Bilevel Test Problem Development

Bilevel multi-objective optimization problems are different from single-level multi-objective optimization problems in that the Pareto-optimal solutions for the lower level multi-objective optimization problems are feasible solutions to the upper level problem. Thus, while developing a bilevel multi-objective test problem, we should have ways to test an algorithm's ability to handle complexities in both lower and upper level problems independently and collectively. In addition, we would also like to have have a knowledge of the exact location (and relationships) of Pareto-optimal points of each problem. We summarize in the following some properties which we would like to have in a bilevel multi-objective test problem:

1) Exact Pareto-optimal in both lower and upper level problems are easy to know. This will enable an user to evaluate the performance of an algorithm easily.
2) Problems are scalable with respect to number of variables. This will enable an user to test whether an algorithms scales well with number of variables in both lower and upper levels.
3) Problems are scalable with respect to number of objectives in both lower and upper level problems. This will enable an user to evaluate an algorithm's scalable performance with number of objectives.
4) Lower level problems are difficult to solve to Pareto-optimality. If lower level Pareto-optimal fronts are not found exactly, upper level solutions are not feasible. Therefore, these problems will test an algorithm's ability to converge to the correct Pareto-optimal front (we can borrow ideas here from single level EMO test problems). The shape (convex, non-convex, disjointedness) of the Pareto-optimal front plays an important role in this respect.
5) There is a conflict between lower and upper level problem solving tasks. In these problems, dominated

lower level solutions may be allowed to dominate upper level Pareto-optimal points. These problems will test an algorithm's ability to converge to the lower level Pareto-optimal fronts, despite not enough emphasis by the upper level problem to do so.
6) Desirably, problems are scalable to multiple levels (at most up to three). On the other hand, it will be ideal to have the problems which degenerate to single level test problems as well.
7) Different lower level problems contribute differently to the upper level front. These test problems will test an algorithm's coordinating ability between upper and lower levels in finding the correct upper level Pareto-optimal front.

Various difference principles are possible to construct test problems following above guidelines. Here, we suggest a simple procedure for two-objective optimization problems in both lower and upper levels. Our approach is a bottom-up approach, in which both lower and upper level problems are designed starting with a pre-specified Pareto-optimal front. We begin the construction process from the lower level problem.

We choose a standard bi-objective test problem [19], [8] as a parameterized lower level problem here, which results in a parametric representation of the Pareto-optimal front $(f_1^*(t), f_2^*(t))$ for $t \in [t_{\min}, t_{\max}]$. For example, if ZDT1 is chosen, the Pareto-optimal front is given by: $(f_1^*(t) = t,\ f_2^*(t) = 1 - t^2)$ for $t \in [0, 1]$. Thus, the bi-objective front is always a one-parameter curve in the case of a bi-objective problem. Since all these solutions are feasible for the upper level problem and since only a subset (or none) of these solutions may qualify as the upper level Pareto-optimal solutions, we can now map these solutions to another one-dimensional non-dominated front in the upper level problem. We illustrate this aspect in Figure 1. The lower level Pareto-

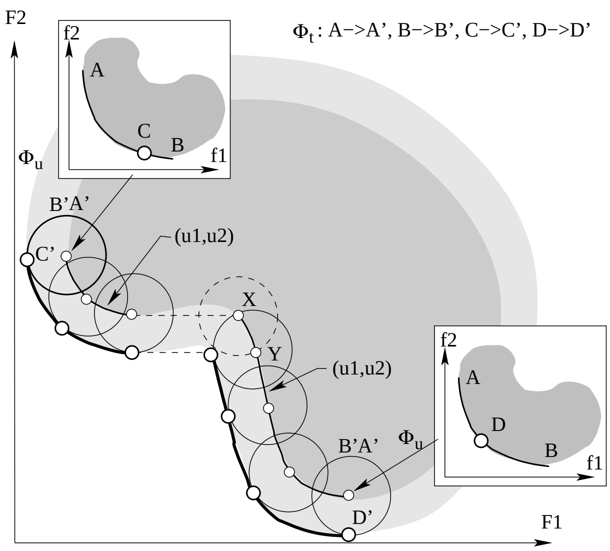

Fig. 1. The construction principle of bilevel test problems.

optimal front (AB) in the f_1-f_2 space is mapped to a complete circle (A′B′) in the F_1-F_2 space. Different lower level Pareto-optimal fronts are mapped and placed at different locations on the upper level front by using a parameter v. The figure shows some such mappings at a few locations with circles. Many simple ideas for the mapping are possible and the figure demonstrates one such simple mapping.

Next, we choose a parameterized, trade-off relationship between two terms u_1 and u_2 as a function of v in the upper level objective space, such that (u_1^*, u_2^*) corresponds to the non-dominated relationship among u_1 and u_2. Many different values of v are possible and for every v, the lower level Pareto-optimal front is $(f_1(t(v)), f_2(t(v)))$ for $t \in [t_{\min}(v), t_{\max}(v)]$. These lower level solutions get mapped using ($\Phi_u : (f_1, f_2) \rightarrow (u_1, u_2)$) on a circle to the upper level objective space having the point v (on u_1-u_2 curve) as the center with a pre-specified radius. Another mapping Φ_t is used to map the parameter t to t'. The following is commutative diagram of both mappings:

$$\begin{array}{ccc} t(v) & \longrightarrow & (f_1^*, f_2^*)(v) \\ \Big\downarrow{\scriptstyle \Phi_t} & & \Big\downarrow{\scriptstyle \Phi_u} \\ t'(v) & \longrightarrow & (F_1, F_2)(v) \end{array}$$

The final upper level Pareto-optimal front can then be derived with the non-dominated solutions from all circles in the F_1-F_2 space, or mathematically

$$(F_1^*, F_2^*) = \mathrm{ND}\left[\cup_{v \in \mathrm{argmin}_{(v)}(u_1,u_2)} (F_1, F_2)(v)\right].$$

Figure 1 depicts one such scenario, in which the lower level front is convex. The mapping Φ_u is such that the lower level Pareto-optimal points (A to B) in f_1-f_2 space is mapped to a full circle (A′ to B′) on the F_1-F_2 space. Thus only a portion of the lower level Pareto-optimal points are non-dominated in the upper level. The mapping Φ_t is chosen to be linear here (linearly mapping AB to to A′B′), but any other mapping can also be chosen to vary the density of points on the non-dominated part of the front in the upper level. The upper level trade-off information on v is non-convex (marked as u_1-u_2 relationship) in the example problem. It is now clear that every upper level Pareto-optimal front comes from a different v and is a different point in the corresponding lower level Pareto-optimal front. Interestingly, some non-dominated v values (in the range XY) do not lead to upper level Pareto-optimal front.

To solve the above problem to upper level Pareto-optimality, an algorithm must first find the exact lower level Pareto-optimal front for every value of the upper level variable v. The upper level optimization then must sort out and locate all possible v which contribute to the upper level Pareto-optimal front. Not only does the algorithm find all such v values, for each such value it should also preserve a well distributed set of lower level Pareto-optimal points which are essential to span the upper level Pareto-optimal front. In the example of the figure, only one solution comes from each lower level front, but a different mapping Φ_u may cause a set of solutions from some lower level fronts to lie on the upper level Pareto-optimal front.

To make each level problem somewhat difficult, additional terms as a function of additional independent variables can be added in each level. The optimization algorithm must now find minimum values of the additional terms to be on the Pareto-optimal front. The approaches to such fronts can be made difficult by choosing suitable functional forms, thereby making the problem more difficult to solve.

V. Proposed Test Suite

In this section, we propose five test problems based on the above principle of bilevel test problem construction. Two objectives at each level are proposed here, but they can be extended to higher objectives with some more consideration. For the following test problems we refer the variable vector $\mathbf{x}_l$ as $\mathbf{x}$ and the variable vector $\mathbf{x}_u$ as $\mathbf{y}$.

A. Problem 1

We start with a problem having linear Pareto-optimal fronts in both levels. Problem 1 has $K+L+1$ variables, which are all real-valued.

$$\begin{array}{l}
\text{minimize} \quad \mathbf{F}(\mathbf{x},\mathbf{y}) = \\
\left\{ \begin{array}{l} (1-x_1)(1+\sum_{j=2}^{K} x_j^2)y_1 \\ x_1(1+\sum_{j=2}^{K} x_j^2)y_1 \end{array} \right\}, \\
\text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})} \\
\quad \left\{ \mathbf{f}(\mathbf{x}) = \left(\begin{array}{l} (1-x_1)(1+\sum_{j=K+1}^{K+L} x_j^2)y_1 \\ x_1(1+\sum_{j=K+1}^{K+L} x_j^2)y_1 \end{array} \right) \right| \\
\qquad g_1(\mathbf{x}) = (1-x_1)y_1 + \frac{1}{2}x_1 y_1 - 1 \geq 0 \Big\}, \\
\quad -1 \leq x_1 \leq 1, \quad 1 \leq y_1 \leq 2, \\
\quad -(K+L) \leq x_i \leq (K+L), \quad \text{for } i = 2,\ldots,(K+L).
\end{array} \tag{2}$$

For a fixed y_1, the lower level Pareto-optimal front corresponds to the line $f_1 + f_2 = y_1$, as shown in Figure 2. The front is restricted at one end by the constraint and at the other end by the x-axis. The constraint boundary does not allow the lower lower level front to continue further which otherwise would have extended until it intercepts the y-axis. At the lower level front, $x_i = 0$ for $i = K+1,\ldots,K+L$ and $x_1 \in [0, 2(1-\frac{1}{y_1})]$, the upper bound depends on the intersection point of the lower level front and the constraint boundary.

The functional relationship for the upper level front is $F_1 + \frac{1}{2}F_2 = 1$, which is the constraint boundary. For the upper level front $x_i = 0$ for $i = 2,\ldots,K$, every Pareto-optimal point corresponds to a different y_1 value in $[1,2]$ and a different x_1 value in $[0,1]$. By increasing K, the number of variables can be increased at the upper level and by increasing L the number of variables can be increased at the lower level. Thus, the complexity at each level can be increased independently by varying K and L. For this test problem we have taken $K = 3$ and $L = 2$ (6-variable test problem).

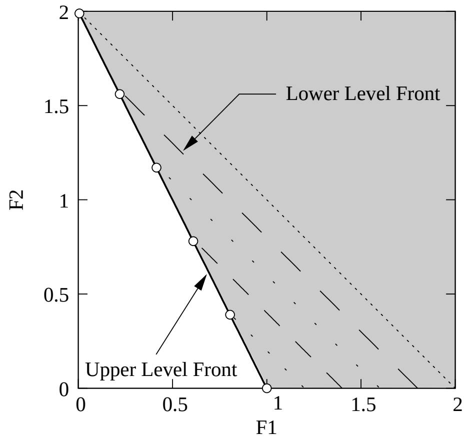

Fig. 2. Pareto-optimal front for problem 1.

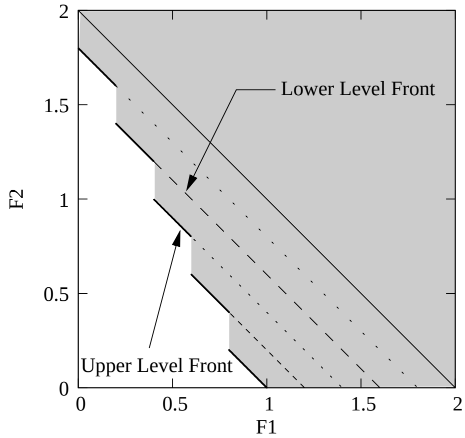

Fig. 3. Pareto-optimal front for problem 2.

B. Problem 2

This problem is similar to problem 1 except that the upper level Pareto-optimal front is constructed from multiple points from a few lower level Pareto-optimal front. There are $K+L+1$ real-valued variables in this problem.

$$\begin{array}{l}
\text{minimize} \quad \mathbf{F}(\mathbf{x},\mathbf{y}) = \\
\left\{ \begin{array}{l} (1-x_1)(1+\sum_{j=2}^{K} x_j^2)y_1 \\ x_1(1+\sum_{j=2}^{K} x_j^2)y_1 \end{array} \right\}, \\
\text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})} \\
\quad \left\{ \mathbf{f}(\mathbf{x}) = \left(\begin{array}{l} (1-x_1)(1+\sum_{j=K+1}^{K+L} x_j^2)y_1 \\ x_1(1+\sum_{j=K+1}^{K+L} x_j^2)y_1 \end{array} \right) \right| \\
\qquad g_1(\mathbf{x}) = (1-x_1)y_1 + \frac{1}{2}x_1 y_1 - 2 + \frac{1}{5}\,[5(1-x_1)y_1 \\
\qquad\quad +0.2] \geq 0 \Big\}, [\cdot] \text{ denotes greatest integer function}, \\
\quad -1 \leq x_1 \leq 1, \quad 1 \leq y_1 \leq 2, \\
\quad -(K+L) \leq x_i \leq (K+L), \quad \text{for } i = 2,\ldots,(K+L).
\end{array} \tag{3}$$

This test problem is similar to the test problem 1 except the constraint at the lower level. The lower level constraint in this test problem is a discontinuous step function which allows multiple points from each lower level front to enter the upper level front. In this problem also the lower level front is given by $f_1 + f_2 = y_1$. At the lower level front $x_i = 0$ for $i = K+1, \ldots, K+L$ and $x_1 \in [0, a]$, where $a \in \{0.2, 0.4, 0.6, 0.8, 1\}$. The upper bound ($a$) of x_1 is determined by the intersection point of the lower level front and the constraint boundary.

The functional relationship for the upper level front is $F_1 + \frac{1}{2}F_2 = 2 - \frac{1}{5}[F_1 + 0.2]$ which is the constraint boundary. For the upper level front, $x_i = 0$ for $i = 2, \ldots, K$ and every Pareto-optimal point corresponds to $y_1 \in \{0, 0.2, 0.4, 0.6, 0.8\}$ and a different x_1 value in $[0, 1]$ (Figure 3). For this test problem we have taken $K = 3$ and $L = 2$ (6-variable test problem).

C. Problem 3

The next problem has $2K$ variables with K real-valued variables each for $\mathbf{x}$ and $\mathbf{y}$:

$$
\begin{aligned}
&\text{minimize} \quad \mathbf{F}(\mathbf{x}, \mathbf{y}) = \\
&\left\{ \begin{array}{c} (1 + r - \cos(\alpha\pi y_1)) + \sum_{j=2}^{K}(y_j - \frac{j-1}{2})^2 \\ +\tau \sum_{i=2}^{K}(x_i - y_i)^2 - r\cos\left(\frac{\pi}{2}\frac{x_1}{y_1}\right) \\ (1 + r - \sin(\alpha\pi y_1)) + \sum_{j=2}^{K}(y_j - \frac{j-1}{2})^2 \\ +\tau \sum_{i=2}^{K}(x_i - y_i)^2 - r\sin\left(\frac{\pi}{2}\frac{x_1}{y_1}\right) \end{array} \right\}, \\
&\text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})} \\
&\quad \left\{ \mathbf{f}(\mathbf{x}) = \left(\begin{array}{c} x_1^2 + \sum_{i=2}^{K}(x_i - y_i)^2 \\ + \sum_{i=2}^{K} 10(1 - \cos(4\pi(x_i - y_i))) \\ \sum_{i=1}^{K}(x_i - y_i)^2 \\ + \sum_{i=2}^{K} 10|\sin(4\pi(x_i - y_i)| \end{array} \right) \right\}, \\
&\quad -K \le x_i \le K, \quad \text{for } i = 1, \ldots, K, \\
&\quad 1 \le y_1 \le 4, \quad -K \le y_j \le K, \quad \text{for } j = 2, \ldots, K.
\end{aligned} \tag{4}
$$

The lower level Pareto-optimal front for a given $\mathbf{y}$ vector corresponds to $x_i = y_i$ for $i = 2, \ldots, K$ and $x_1 \in [0, y_1]$. The objectives are related as follows: $f_2^* = (\sqrt{f_1^*} - y_1)^2$. Let us first consider $\tau = 1$. The upper level Pareto-optimal front corresponds to $y_i = (j-1)/2$ for $j = 2, \ldots, K$. The parametric functional relationship is $u_1 = 1 + r - \cos(\alpha\pi y_1)$ and $u_2 = 1 + r - \sin(\alpha\pi y_1)$. This is a circle of radius one and center at $((1+r), (1+r))$ in the F_1-F_2 space. Thus, the non-dominated portion is the third quadrant of this circle and this happens for $y_1 \in (2p + [0, 0.5])/\alpha$, where p is an integer including zero. For $\alpha = 1$ and for $1 \le y_1 \le 4$, this happens for $y_1 \in [2, 2.5]$ and $y_1 = 4$. Accumulating the non-dominated portions of all circles of radius r at every optimal y_1, we have the overall upper level Pareto-optimal front defined as a circle of radius $(1+r)$ and center at $((1+r), (1+r))$, as shown in Figure 4.

For this test problem, we use $K = 3$ (6-variable problem), $r = 0.1$, and $\alpha = 1$. By increasing K the number of variables can be increased without affecting the Pareto-optimal frontier. The effect of r is not that significant for a real-valued variable y_1. However, if a discrete y_1 is chosen, an increase in r will have a range of lower level Pareto-optimal solutions to qualify as upper level Pareto-optimal solutions. Also, an increase in α causes multi-modal y_1 solutions to appear on the upper level Pareto-optimal front.

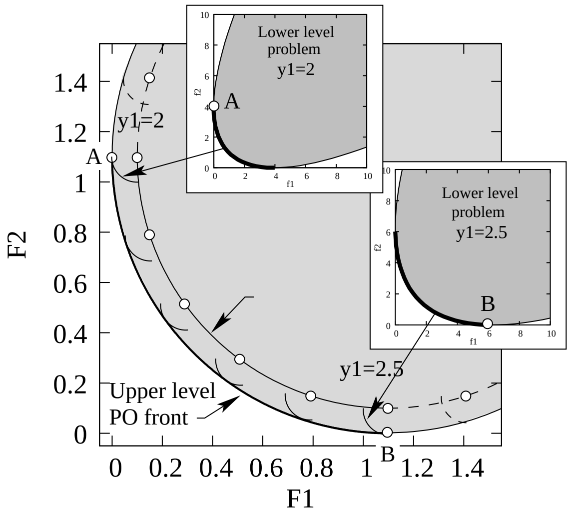

Fig. 4. Pareto-optimal front for problem 3.

For example, for $\alpha = 2$, $y_1 \in [1, 1.25] \cup [2, 2.25] \cup [3, 3.25]$ will correspond to the upper level Pareto-optimal front.

An interesting scenario happens when $\tau = -1$ is set. Now, if any $\mathbf{x}$ is not Pareto-optimal to a lower level problem, some positive amount is deducted in the upper level objectives, thereby allowing points to exist below the Pareto-optimal curve of the upper level problem. These points are infeasible to the upper level due to their non-optimal property at the lower level problems and if allowed to exist in the upper level will dominate the true upper level Pareto-optimal points. Thus, this problem will be difficult to solve than the problem with $\tau = 1$, as the upper level problem does not motivate the solutions to become Pareto-optimal at the lower level and the overall procedure will end up finding infeasible solutions.

D. Problem 4

The next problem consists of $2K$ real-valued variables, but the upper level Pareto-optimal front corresponds to a few values of y_1:

$$
\begin{aligned}
&\text{minimize} \quad \mathbf{F}(\mathbf{x}, \mathbf{y}) = \\
&\left\{ \begin{array}{c} u_1(y_1) + \sum_{j=2}^{K}\left[y_j^2 + 10(1 - \cos(4\pi y_i))\right] \\ -r\cos\left(2\pi\frac{x_1}{y_1}\right) \\ u_2(y_1) + \sum_{j=2}^{K}\left[y_j^2 + 10(1 - \cos(4\pi y_i))\right] \\ -r\sin\left(2\pi\frac{x_1}{y_1}\right) \end{array} \right\}, \\
&\text{subject to} \quad (\mathbf{x}) \in \\
&\quad \text{argmin}_{(\mathbf{x})} \left\{ \mathbf{f}(\mathbf{x}) = \left(\begin{array}{c} x_1^2 + \sum_{i=2}^{K}(x_i - y_i)^2 \\ \sum_{i=1}^{K} i(x_i - y_i)^2 \end{array} \right) \right\}, \\
&\quad -K \le x_i \le K, \quad \text{for } i = 1, \ldots, K, \\
&\quad 0.001 \le y_1 \le K, \quad -K \le y_j \le K, \quad \text{for } j = 2, \ldots, K,
\end{aligned} \tag{5}
$$

where

$$
\begin{aligned}
u_1(y_1) &= \begin{cases} \cos(0.2\pi)y_1 + \sin(0.2\pi)\sqrt{|0.02\sin(5\pi y_1)|}, \\ \quad \text{for } 0 \le y_1 \le 1, \\ y_1 - (1-\cos(0.2\pi)), \quad y_1 > 1 \end{cases} \\
u_2(y_1) &= \begin{cases} -\sin(0.2\pi)y_1 + \cos(0.2\pi)\sqrt{|0.02\sin(5\pi y_1)|}, \\ \quad \text{for } 0 \le y_1 \le 1, \\ 0.1(y_1 - 1) - \sin(0.2\pi), \quad \text{for } y_1 > 1. \end{cases}
\end{aligned} \tag{6}
$$

For every $\mathbf{y}$ vector, the lower level front occurs for $x_i = y_i$ for $i = 2, \ldots, K$ and $x_1 \in [0, y_1]$ and the functional relationship is $f_2^* = (\sqrt{f_1^*} - y_1)^2$. Although a number of different values of y_1 correspond to a non-dominated trade-off with respect to (u_1, u_2), the upper level Pareto-optimal front corresponds to only six discrete values of y_1 (=0.001, 0.2, 0.4, 0.6, 0.8 and 1). Here we suggest using $r = 0.25$. In this problem, a lower level front is mapped to a circle which is centered at $(u_1(y_1), u_2(y_1))$ and has a radius of 0.25. The resulting Pareto-optimal front with six circular arcs are shown in Figure 5. It is clear that the upper level Pareto-optimal front is constructed only from different non-dominated regions from each of the six y_1 values. Interestingly, despite y_1 taking any real value within $[0.001, K]$, no other y_1 value contributes to the upper level Pareto-optimal frontier.

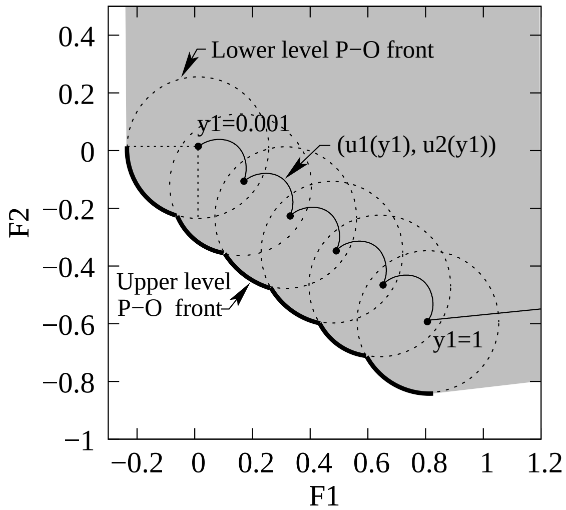

Fig. 5. Pareto-optimal front for problem 4.

Like in Problem 3, we can add the term $\tau \sum_{i=2}^{K}(x_i - y_i)^2$ on both F_1 and F_2 with $\tau = -1$ in order to make the problem more difficult.

E. Problem 5

The next problem has a discrete variable y_1 and other variables are all real-valued:

$$
\begin{aligned}
&\text{minimize} \quad \mathbf{F}(\mathbf{x}, \mathbf{y}) = \\
&\left\{ \begin{array}{l} y_1 + \sum_{j=3}^{K}(y_j - j/2)^2 - R(y_1)\cos\left(4\tan^{-1}\left(\frac{y_2 - x_2}{y_1 - x_1}\right)\right) \\ y_2 + \sum_{j=3}^{K}(y_j - j/2)^2 - R(y_1)\sin\left(4\tan^{-1}\left(\frac{y_2 - x_2}{y_1 - x_1}\right)\right) \end{array} \right\}, \\
&\text{subject to} \quad (\mathbf{x}) \in \text{argmin}_{(\mathbf{x})} \\
&\quad \left\{ \mathbf{f}(\mathbf{x}) = \left(\begin{array}{c} x_1 + \sum_{i=3}^{K}(x_i - y_i)^2 \\ x_2 + \sum_{i=3}^{K}(x_i - y_i)^2 \end{array} \right) \middle| \right. \\
&\qquad\qquad \left. g_1(\mathbf{x}) = (x_1 - y_1)^2 + (x_2 - y_2)^2 \le r^2 \right\}, \\
&\quad G(\mathbf{y}) = y_2 - (1 - y_1^2) \ge 0, \\
&\quad -K \le x_i \le K, \quad \text{for } i = 1, \ldots, K, \\
&\quad 0 \le y_j \le K, \quad \text{for } j = 1, \ldots, K, \\
&\quad y_1 \text{ is a multiple of } 0.1.
\end{aligned} \tag{7}
$$

Here we suggest a periodically changing radius: $R(y_1) = 0.1 + 0.15|\sin(2\pi(y_1 - 0.1)|$ and $r = 0.2$. The corresponding Pareto-optimal solutions for the lower level problem are given as follows:

$$
\begin{aligned}
&\{(\mathbf{x}) \in \mathbb{R}^K | (x_1 - y_1)^2 + (x_2 - y_2)^2 = r^2, \\
&\quad x_1 \le y_1, x_2 \le y_2, x_i = y_i, \forall i \ge 3\}.
\end{aligned} \tag{8}
$$

For the upper level Pareto-optimal points, $y_i = j/2$ for $j \le 3$. The variables y_1 and y_2 take values satisfying constraint $G(\mathbf{y}) = 0$. For each such combination, variables x_1 and x_2 take values according to satisfying equation 8 and on the third quadrant of the circle in the F_1-F_2 space. Notice in Figure 6 how lower level Pareto-optimal solutions for $y_1 = 0.1$ and 0.2 mapped to corresponding circles in the upper level problem get dominated by that for $y_1 = 0$ and 0.3. Every lower level front has an unequal contribution to the upper level Pareto-optimal front.

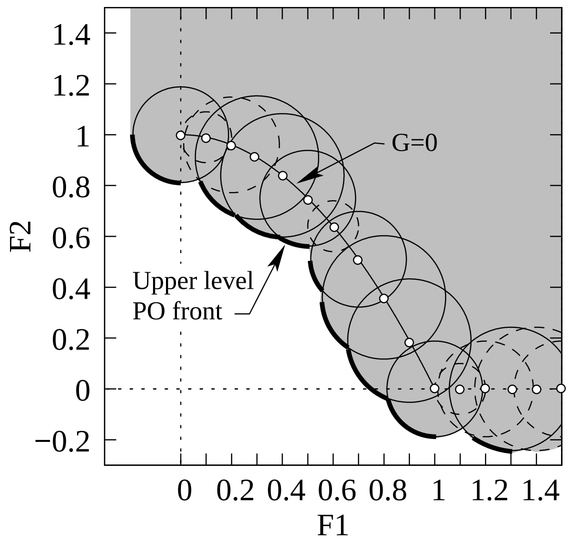

Fig. 6. Pareto-optimal front for problem 5.

Like before, we can add the term $\tau \sum_{i=3}^{K}(x_i - y_i)^2$ (with

$\tau = -1$) on both objectives F_1 and F_2 to make the problem more difficult. It is interesting to note that although lower level dominated solutions may lie below the upper level Pareto-optimal front those solutions may still satisfy the upper level constraint $G()$. Thus, a negative value of τ will test an algorithm's ability to converge to the Pareto-optimal front of a lower level problem, despite the apparent emphasis of dominated solutions of the lower level problem by the upper level problem.

VI. Proposed Procedure (BLEMO)

The proposed method uses the elitist non-dominated sorting GA or NSGA-II [5] in both levels. The upper level population (of size N_u) uses NSGA-II operations for T_u generations with upper level objectives $(\mathbf{F})$ and constraints $(\mathbf{G})$ in determining non-dominated rank and crowding distance values of each population member. However, the evaluation of a population member calls a lower level NSGA-II simulation with a population size of N_l for T_l generations. The upper level population has $n_s = N_u/N_l$ subpopulations of size N_l each. Each subpopulation has the same $\mathbf{x}_u$ variable vector. This mechanism helps to coordinate the upper and lower level simulations in an easier way. The solutions in the best non-dominated upper level are saved in an archive, which in turn participate in genetic operations in subsequent generations. The details of BLEMO procedure can be found in another study by the authors [7]. Overall computational complexity of the proposed BLEMO algorithm is bounded by $N_u(2T_u + 1)(T_l + 1)$.

VII. Results

We use the following parameter settings: $N_u = 400$, $T_u = 100$, $N_l = 40$, and $T_l = 40$ for problems 1 and 2. $N_u = 400$, $T_u = 200$, $N_l = 40$, and $T_l = 40$ for problems 3, 4 and 5. The other NSGA-II parameters are set as follows: for SBX crossover, $p_c = 0.9$, $\eta_c = 15$ [4] and for polynomial mutation operator, $p_m = 0.1$, and $\eta_m = 20$ [3].

A. Problem 1

For this problem we are using $K = 3$ and $L = 2$ which makes it a 6-variable problem. The algorithm is able to converge to the Pareto-front as shown in Figure 7. The algorithm did not have any problem in finding the end point of each lower level front which corresponds to the upper level Pareto-optimal front. This test problem poses a challenge for the existing algorithms to find a particular point from the lower level which corresponds to the upper level solution. If the algorithm is able to solve the lower level problem completely but is unable to find the point corresponding to the upper level front then the lower level run turns out to be of no use.

B. Problem 2

For this problem we are using $K = 3$ and $L = 2$ which makes it a 6-variable problem. The algorithm was able to converge to the Pareto-front as shown in Figure 8. This problem required to find particular y_1 values to form the upper level front, which the algorithm was able to figure out efficiently. This test problem tests the ability of the algorithm to do a search over $\mathbf{y}$ and locate the correct lower level fronts which correspond to the upper level frontier..

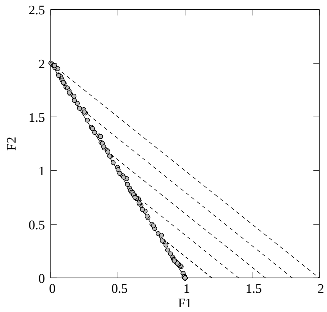

Fig. 7. Obtained Pareto-optimal front for problem 1 (6-variable).

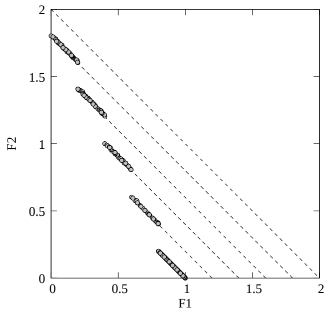

Fig. 8. Obtained Pareto-optimal front for problem 2 (6-variable).

C. Problem 3

This test problem was solved for $K = 3$ (6-variables) and $K = 4$ (8-variables) with $\tau = 1$. Figure 9 shows the obtained Pareto-optimal front for the 6-variable problem and Figure 10 shows the Pareto-optimal front for the 8-variable problem. The algorithm was able to solve the problem for $K = 3$ but its performance deteriorated when run for $K = 4$. The algorithm also faced problems when this test problem was run for $\tau = -1$. It failed for this problem when the non-optimal lower level solutions were made to dominate the upper level solutions by using a negative value of τ. This is an important test criteria to check the performance of the existing algorithms and evaluates their ability to coordinate the upper and the lower level tasks.

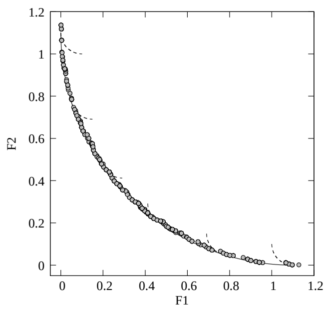

Fig. 9. Obtained Pareto-optimal front for problem 3 (6-variable).

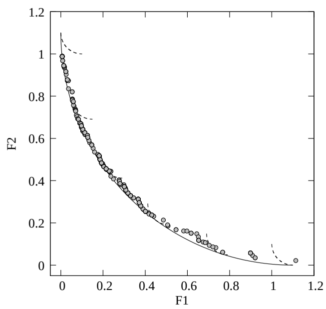

Fig. 10. Obtained Pareto-optimal front for problem 3 (8-variable).

D. Problem 4

This test problem was solved for $K = 2$ (4-variables) and $K = 3$ (6-variables). Figure 11 shows the obtained Pareto-optimal front for the 4-variable problem and Figure 12 shows the Pareto-optimal front for the 6-variable problem. The algorithm was able to get close to the front for $K = 2$ but the performance sharply deteriorated when tried for $K = 3$. This test problem provides local fronts at the upper level which can misguide the algorithm to a local convergence. The number of local fronts at the upper level increases exponentially with the increase in the number of variables. It evaluates an algorithm's ability to do an extensive search before converging to a front. For this problem too (as test problem 2), the Pareto-optimal front corresponds to finite values of y so it also checks the competence of an algorithm in doing a thorough and radical search over $\mathbf{y}$.

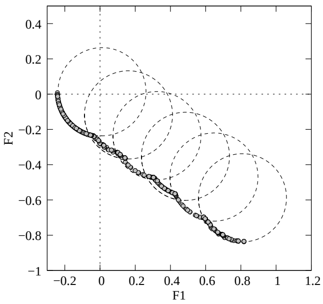

Fig. 11. Obtained Pareto-optimal front for problem 4 (4-variable).

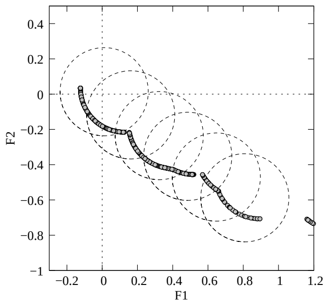

Fig. 12. Obtained Pareto-optimal front for problem 4 (6-variable).

E. Problem 5

This problem was solved for $K = 3$ (6-variables) and $K = 5$ (10-variables). Figure 13 shows the obtained Pareto-optimal front for the 6-variable problem and Figure 14 shows the Pareto-optimal front for the 10-variable problem. The algorithm was able to converge to the front for both the cases, though a degradation in the performance can be observed for higher number of variables. This test problem uses a discreet variable which would make the task tough for the classical approaches to handle this problem.

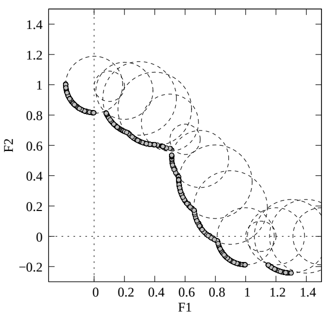

Fig. 13. Obtained Pareto-optimal front for problem 5 (6-variable).

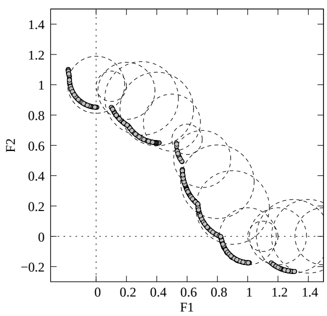

Fig. 14. Obtained pareto-optimal front for problem 5 (10-variable).

VIII. Conclusions

Optimization researchers and practitioners have so far shown a lukewarm interest in solving bilevel multi-objective optimization problems, which have a considerable importance in practice. In the hope of revamping a research and application interest in this area, in this paper, we have suggested a systematic procedure of constructing test problems for adequately evaluating the efficiency of possible optimization algorithms for solving bilevel multi-objective optimization problems. The construction procedure has been used to suggest five test problems which are scalable in number of objectives and variables. Each test problem offers different aspects of complexity which a bilevel optimization problem may possess. An application of our previously proposed BLEMO algorithm to these test problems has shown that these problems are too difficult to be solved for more than four or six variables. In this respect, the suggested test problems seem to be challenging and this study emphasizes an urgent need of more studies in the development of BLEMO algorithms.

Acknowledgments

Authors wish to thank Academy of Finland and Foundation of Helsinki School of Economics for their support of this study.

References

[1] P. H. Calamai and L. N. Vicente. Generating quadratic bilevel programming test problems. *ACM Trans. Math. Software*, 20(1):103–119, 1994.

[2] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operational Research*, 153:235–256, 2007.

[3] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

[4] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[6] K. Deb and A. Sinha. Solving bilevel multi-objective optimization problems using evolutionary algorithms. Technical Report KanGAL Report No. 2008005, Kanpur, Indian Institute of Technology, India, 2008.

[7] K. Deb and A. Sinha. Towards understanding evolutionary bilevel multi-objective optimization algorithm. Technical Report KanGAL Report No. 2008006, Kanpur, Indian Institute of Technology, India, 2008.

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization*, pages 105–145. London: Springer-Verlag, 2005.

[9] S. Dempe, J. Dutta, and S. Lohse. Optimality conditions for bilevel programming problems. *Optimization*, 55(5âĂŞ6):505âĂŞ–524, 2006.

[10] G. Eichfelder. Soving nonlinear multiobjective bilevel optimization problems with coupled upper level constraints. Technical Report Preprint No. 320, Preprint-Series of the Institute of Applied Mathematics, Univ. Erlangen-NÃijrnberg, Germany, 2007.

[11] J. Fliege and L. N. Vicente. Multicriteria approach to bilevel optimization. *Journal of Optimization Theory and Applications*, 131(2):209–225, 2006.

[12] A. Koh. Solving transportation bi-level programs with differential evolution. In *2007 IEEE Congress on Evolutionary Computation (CEC-2007)*, pages 2243–2250. IEEE Press, 2007.

[13] R. Mathieu, L. Pittard, and G. Anandalingam. Genetic algorithm based approach to bi-level linear programming. *Operations Research*, 28(1):1–21, 1994.

[14] K. Miettinen, K. Deb, J. Jahn, W. Ogryczak, K. Shimoyama, and R. Vetschera. Future challenges. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective optimization, (Lecture Notes in Computer Science, 5252)*, pages 435–461. Berlin: Springer, 2008.

[15] V. Oduguwa and R. Roy. Bi-level optimisation using genetic algorithm. In *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAISâĂŹ02)*, pages 322–327, 2002.

[16] L. N. Vicente and P. H. Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, 5(3):291–306, 2004.

[17] Y. Wang, Y.-C. Jiao, and H. Li. An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(2):221–232, 2005.

[18] Y. Yin. Genetic algorithm based approach for bilevel programming models. *Journal of Transportation Engineering*, 126(2):115–120, 2000.

[19] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal*, 8(2):125–148, 2000.