Solving Bilevel Multi-Objective Optimization Problems Using Evolutionary Algorithms

Kalyanmoy Deb and Ankur Sinha

Department of Mechanical Engineering
Indian Institute of Technology Kanpur
PIN 208016, India
deb@iitk.ac.in, ankur.sinha@hse.fi

KanGAL Report Number 2008005

Abstract. Bilevel optimization problems require every feasible upper-level solution to satisfy optimality of a lower-level optimization problem. These problems commonly appear in many practical problem solving tasks including optimal control, process optimization, game-playing strategy development, transportation problems, and others. In the context of a bilevel single objective problem, there exists a number of theoretical, numerical, and evolutionary optimization results. However, there does not exist too many studies in the context of having multiple objectives in each level of a bilevel optimization problem. In this paper, we address bilevel multi-objective optimization issues and propose a viable algorithm based on evolutionary multi-objective optimization (EMO) principles. Proof-of-principle simulation results bring out the challenges in solving such problems and demonstrate the viability of the proposed EMO technique for solving such problems. This paper scratches the surface of EMO-based solution methodologies for bilevel multi-objective optimization problems and should motivate other EMO researchers to engage more into this important optimization task of practical importance.

1 Introduction

In evolutionary optimization, a few studies have considered solving bilevel programming problems in which an upper level solution is feasible only if it is one of the optimum of a lower level optimization problem. Such problems are found abundantly in practice, particularly in optimal control, process optimization, transportation problems, game playing strategies, reliability based design optimization, and others. In such problems, the lower level optimization task ensures a certain quality or certain physical properties which make a solution acceptable. Often, such requirements come up as equilibrium conditions, stability conditions, mass/energy balance conditions, which are mandatory for any solution to be feasible. For example. in reliability based design optimization, a feasible design must correspond to a certain specified reliability against failures. Solutions satisfying such conditions or requirements are not intuitive to obtain,

rather they often demand an optimization problem to be solved. These essential tasks are posed as lower level optimization tasks in a bilevel optimization framework. The upper level optimization then must search among such reliable, equilibrium or stable solutions to find an optimal solution corresponding to one or more different (higher level) objectives.

Despite the importance of such problems in practice, the difficulty of searching and defining optimal solutions for bilevel optimization problems [7] exists. Despite the lack of theoretical results, there exists a plethora of studies related to bilevel single-objective optimization problems [1, 3, 12, 15] in which both upper and the lower level optimization tasks involve exactly one objective each. Despite having a single objective in the lower level task, usually in such problems the lower level optimization problem has more than one optimum. The goal of a bilevel optimization technique is then to first find the lower level optimal solutions and then search for the optimal solution for the upper level optimization task. In the context of bilevel multi-objective optimization studies, however, there does not exist too many studies using classical methods [8] and none to our knowledge using evolutionary methods, probably because of the added complexities associated with solving each level. In such problems, every lower level optimization problem has a number of trade-off optimal solutions and the task of the upper level optimization algorithm is to focus its search on multiple trade-off solutions which are members of optimal trade-off solutions of lower level optimization problems.

In this paper, we suggest a viable evolutionary multi-objective optimization (EMO) algorithm for solving bilevel problems. We simulate the proposed algorithm on five different test problems, including a bilevel single-objective optimization problem. This proof-of-principle study shows viability of EMO for solving bilevel optimization problems and should encourage other EMO researchers to pay attention to this important class of practical optimization problems.

2 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:

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

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.

3 Classical 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 [7], 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 [9] 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. However, such an idea is interesting and can be pursued in the future.

In the context of bilevel single-objective optimization problems, there exists a number of studies, including some useful reviews [3, 13], test problem generators [1], and even some evolutionary algorithm (EA) studies [12, 11, 15, 10, 14]. However, there does not seem to be too many studies on bilevel multi-objective optimization.

A recent study by Eichfelder [8] suggested a refinement based strategy in which the algorithm starts with a uniformly distributed set of points on $\mathbf{x}_u$. Thereafter, for each $\mathbf{x}_u$ vector, the lower level Pareto-optimal solutions are found using a classical generating based optimization method. The set of such points obtained are said to be an approximation of the induced set. Non-dominated points with respect to the upper level problem are chosen from this set and they provide an approximate idea of the desired upper level Pareto-optimal front. Now, the chosen $\mathbf{x}_u$ vectors are refined in their vicinities and the lower level optimizations are repeated till a good approximation of the Pareto-optimal front is obtained. The difficulty with such a technique is that if the dimension of $\mathbf{x}_u$ is high, generating a uniformly spread $\mathbf{x}_u$ vectors and refining the resulting $\mathbf{x}_u$ vector will be computationally expensive. Definitely, an optimization algorithm

for simultaneous selection of $\mathbf{x}_u$ and corresponding optimal $\mathbf{x}_l$ vectors will be more effective.

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 the following, we suggest one such procedure.

4 Proposed Procedure (BLEMO)

The proposed method uses the elitist non-dominated sorting GA or NSGA-II [6], however any other EMO procedures can also be used instead. 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 a special feature. The population has $n_s = N_u/N_l$ subpopulations of size N_l each. Each subpopulation has the same $\mathbf{x}_u$ variable vector. To start the proposed BLEMO, we create all solutions at random, but maintain the above structure. From thereon, the proposed operations ensure that the above-mentioned structure is maintained from one generation to another. In the following, we describe one iteration of the proposed BLEMO procedure. At the start of the upper level NSGA-II generation t, we have a population P_t of size N_u. Every population member has the following quantities computed from the previous iteration: (i) a non-dominated rank ND_u corresponding to $\mathbf{F}$ and $\mathbf{G}$, (ii) a crowding distance value CD_u corresponding to $\mathbf{F}$, (iii) a non-dominated rank ND_l corresponding to $\mathbf{f}$ and $\mathbf{g}$, and (iv) a crowding distance value CD_l using $\mathbf{f}$. For every subpopulation in the upper level population, members having the best non-domination rank (ND_u) are saved as an 'elite set' which will be used in the recombination operator in the lower level optimization task of the same subpopulation.

Step 1: Apply a pair of binary tournament selections on members ($\mathbf{x} = (\mathbf{x}_u, \mathbf{x}_l)$) of P_t using ND_u and CD_u lexicographically. The upper level variable vectors $\mathbf{x}_u$ of two selected parents are then recombined using the SBX operator [5] to obtain two new vectors of which one is chosen at random. The chosen

solution is mutated by the polynomial mutation operator [4] to obtain a child vector (say, $\mathbf{x}_u^{(1)}$). We then create N_l new lower level variable vectors $\mathbf{x}_l^{(i)}$ by applying selection-recombination-mutation operations on entire P_t. Thereafter, N_l child solutions are created by concatenating upper and lower level variable vectors together, as follows: $c_i = (\mathbf{x}_u^{(1)}, \mathbf{x}_l^{(i)})$ for $i = 1, \ldots, N_l$. Thus, for every new upper level variable vector, a subpopulation of N_l lower level variable vectors are created by genetic operations from P_t. The above procedure is repeated for a total of n_s new upper level variable vectors.

Step 2: For each subpopulation of size N_l, we now perform a NSGA-II procedure using lower level objectives ($\mathbf{f}$) and constraints ($\mathbf{g}$) for T_l generations. It is interesting to note that in each lower level NSGA-II, the upper level variable vector $\mathbf{x}_u$ is not changed. For every mating, one solution is chosen as usual using the binary tournament selection using a lexicographic use of ND_l and CD_l, but the second solution is always chosen randomly from the 'elite set'. The mutation is performed as usual. After the lower level NSGA-II simulation is performed for a subpopulation, the resulting solutions are marked with their non-dominated rank (ND_l) and crowding distance value (CD_l). All N_l members from each subpopulation are then combined together in one population (the child population, Q_t). It is interesting to note that in Q_t, there are at least n_s members having $ND_l = 1$ (at least one coming from each subpopulation). Also, in Q_t, there are exactly n_s different $\mathbf{x}_u$ variable vectors.

Step 3: Each member of Q_t is now evaluated with $\mathbf{F}$ and $\mathbf{G}$. Populations P_t and Q_t are combined together to form R_t. The combined population R_t is then ranked according to non-domination and members within an identical non-dominated rank are assigned a crowding distance computed in the $\mathbf{F}$ space. Thus, each member of Q_t gets a upper level non-dominated rank ND_u and a crowding distance value CD_u.

Step 4: From the combined population R_t of size $2N_u$, half of its members are chosen in this step. First, the members of rank $ND_u = 1$ are considered. From them, solutions having $ND_l = 1$ are noted one by one in the order of reducing crowding distance CD_u, for each such solution the entire N_l subpopulation from its source population (either P_t or Q_t) is copied in an intermediate population S_t. If a subpopulation is already copied in S_t and a future solution from the same subpopulation is found to have $ND_u = ND_l = 1$, the subpopulation is not copied again. When all members of $ND_u = 1$ are considered, a similar consideration is continued with $ND_u = 2$ and so on till exactly n_s subpopulations are copied in S_t.

Step 5: Each subpopulation of S_t is now modified using the lower level NSGA-II procedure applied with $\mathbf{f}$ and $\mathbf{g}$ for T_l generations. This step helps progress each lower level populations towards their individual Pareto-optimal frontiers.

Step 6: Finally, all subpopulations obtained after the lower level NSGA-II simulations are combined together to form the next generation population P_{t+1}.

The evaluation of the initial population is similar to the above. First, members of P_0 are created at random with n_s subpopulations, each having an identical $\mathbf{x}_u$ vector for all its subpopulation members. Thereafter, each subpopulation is sent for an update of $\mathbf{x}_l$ vectors to the lower level NSGA-II (with $\mathbf{f}$ and $\mathbf{g}$) for T_l generations. Every member is assigned corresponding ND_l and CD_l values. The resulting subpopulations (from NSGA-II) are combined into one population (renamed as P_0) and evaluated using $\mathbf{F}$ and $\mathbf{G}$. Every member is then assigned a non-dominated rank ND_u and a crowding distance value CD_u.

The good solutions of every generation is saved in an archive (A_t). Initially, the archive A_0 is an empty set. Thereafter, at the end of every upper level generation, solutions having both $ND_u = 1$ and $ND_l = 1$ from P_t is saved in the archive A_t. The non-dominated solutions (with $\mathbf{F}$ and $\mathbf{G}$) of the archive are kept in A_t and rest members are deleted from the archive.

In the above BLEMO, we have used a simple termination rule based on specified number of generations for both lower and upper level tasks. Every lower level task for each subpopulation requires $N_l(T_l + 1)$ solution evaluations and since there are n_s subpopulations in every generation, this requires $n_s N_l(T_l + 1)$ or $N_u(T_l + 1)$ solution evaluations in Step 2. In the initial population evaluation, a final upper level objective and constraint evaluation of N_u is required, but since a solution evaluation refers to both upper and lower level evaluations, this N_u is not extra. For any other generation, Step 5 above requires another $N_u(T_l + 1)$ solution evaluations, thereby totaling $2N_u(T_l + 1)$ solution evaluations. Thus, considering evaluations involved in all generations from $t = 0$ till $t = T_u$, total solution evaluations needed are $N_u(2T_u + 1)(T_l + 1)$.

5 Test Problems and Pareto-Optimal Solutions

In the context of bilevel single-objective optimization, there exists some studies [3, 1] which suggest linear, quadratic and transport related problems. However, to our knowledge, there does not exist any systematic study suggesting test problems for bilevel multi-objective optimization. In this study, we borrow a couple of problems used in [8] and suggest a small and a large-dimensional version of a new test problem.

5.1 Problem 1

Problem 1 has a total of three variables with x_1, x_2 belonging to $\mathbf{x}_l$ and y belonging to x_u and is taken from [8]:

$$\begin{aligned}
&\text{minimize } \mathbf{F}(\mathbf{x}) = \left\{ \begin{array}{c} x_1 - y \\ x_2 \end{array} \right\}, \\
&\text{subject to } (x_1, x_2) \in \text{argmin}_{(x_1,x_2)} \left\{ \mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| g_1(\mathbf{x}) = y^2 - x_1^2 - x_2^2 \geq 0 \right\}, \\
&\qquad\qquad G_1(\mathbf{x}) = 1 + x_1 + x_2 \geq 0, \\
&\qquad\qquad -1 \leq x_1, x_2 \leq 1, \quad 0 \leq y \leq 1.
\end{aligned} \tag{2}$$

Both the lower and the upper level optimization tasks have two objectives each. A little consideration will reveal that for a fixed y value, the feasible region of the lower-level problem is the area inside a circle with center at origin ($x_1 = x_2 = 0$) and radius equal to y. The Pareto-optimal set for the lower-level optimization task for a fixed y is the bottom-left quarter of the circle:

$$\{(x_1, x_2) \in \mathbb{R}^2 \quad | \quad x_1^2 + x_2^2 = y^2, x_1 \leq 0, x_2 \leq 0\}.$$

The linear constraint in the upper level optimization task does not allow the entire quarter circle to be feasible for some y. Thus, at most a couple of points from the quarter circle belongs to the Pareto-optimal set of the overall problem. Eichfelder [8] reported the following Pareto-optimal solutions for this problem:

$$\mathbf{x}^* = \left\{(x_1, x_2, y) \in \mathbb{R}^3 \quad | \quad x_1 = -1 - x_2, x_2 = -\frac{1}{2} \pm \frac{1}{4}\sqrt{8y^2 - 4}, y \in \left[\frac{1}{\sqrt{2}}, 1\right]\right\}. \tag{3}$$

The Pareto-optimal front in F_1-F_2 space is given in parametric form, as follows:

$$\left\{(F_1, F_2) \in \mathbb{R}^2 \quad | \quad F_1 = -1 - F_2 - t, F_2 = -\frac{1}{2} \pm \frac{1}{4}\sqrt{8t^2 - 4}, t \in \left[\frac{1}{\sqrt{2}}, 1\right]\right\}. \tag{4}$$

Figure 1 shows the Pareto-optimal front of problem 1. Lower level Pareto-optimal fronts of some representative y values are also shown on the figure, indicating that at most two such Pareto-optimal solutions (such as points B and C for $y = 0.9$) of a lower level optimization problem becomes the candidate Pareto-optimal solutions of the upper level problem. It is interesting to note that in this problem there exists a number of lower level Pareto-optimal solutions (such as solution A marked in the figure) which are infeasible to the upper level task. Thus, if the lower level optimization is unable to find critical Pareto-optimal solutions (such as B or C) which correspond to the upper level Pareto-optimal solutions, but finds solutions like A in most occasions, the lower

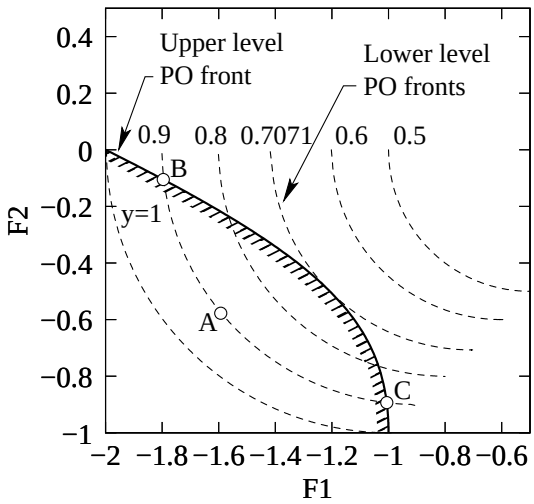

Fig. 1. Pareto-optimal fronts of upper level (complete problem) and some representative lower level optimization tasks are shown for problem 1.

level task becomes useless. This makes the bilevel optimization task challenging and difficult.

5.2 Problem 2

This problem is also taken from [8]:

$$
\begin{aligned}
&\text{minimize} \quad \mathbf{F}(\mathbf{x}) = \left\{ \begin{array}{c} x_1 + x_2^2 + y + \sin^2(x_1+y) \\ \cos(x_2)(0.1+y)(\exp(-\frac{x_1}{0.1+x_2}) \end{array} \right\}, \\
&\text{subject to} \\
&(x_1, x_2) \in \left\{ \begin{array}{l} \text{argmin}_{(x_1,x_2)} \mathbf{f}(\mathbf{x}) = \left(\begin{array}{c} \frac{(x_1-2)^2+(x_2-1)^2}{4} + \frac{x_2 y+(5-y_1)^2}{16} + \sin(\frac{x_2}{10}) \\ \frac{x_1^2+(x_2-6)^4-2x_1y_1-(5-y_1)^2}{80} \end{array} \right) \Big| \\ g_1(\mathbf{x}) = x_2 - x_1^2 \geq 0, \\ g_2(\mathbf{x}) = 10 - 5x_1^2 - x_2 \geq 0, \\ g_3(\mathbf{x}) = 5 - \frac{y}{6} - x_2 \geq 0, \\ g_4(\mathbf{x}) = x_1 \geq 0. \end{array} \right\}, \\
&G_1(\mathbf{x}) \equiv 16 - (x_1 - 0.5)^2 - (x_2 - 5)^2 - (y-5)^2 \geq 0, \\
&0 \leq x_1, x_2, y \leq 10.
\end{aligned}
\tag{5}
$$

For this problem, the exact Pareto-optimal front of the lower or the upper level optimization problem is difficult to derive mathematically. The previous study [8] did not report the true Pareto-optimal front, instead presented a front through their obtained results.

5.3 Problem 3

Next, we create a simplistic bilevel two-objective optimization problem, having $\mathbf{x}_l = (x_1, x_2)$ and $\mathbf{x}_u = (y)$:

$$
\begin{aligned}
&\text{minimize } \mathbf{F}(\mathbf{x}) = \left\{ \begin{array}{c} (x_1-1)^2 + x_2^2 + y^2 \\ (x_1-1)^2 + x_2^2 + (y-1)^2 \end{array} \right\}, \\
&\text{subject to } (x_1, x_2) \in \text{argmin}_{(x_1,x_2)} \left\{ \mathbf{f}(\mathbf{x}) = \left(\begin{array}{c} x_1^2 + x_2^2 \\ (x_1-y)^2 + x_2^2 \end{array} \right) \right\}, \\
&\qquad -1 \leq x_1, x_2, y \leq 2.
\end{aligned}
\tag{6}
$$

For a fixed value of y, the Pareto-optimal solutions of the lower level optimization problem are given as follows: $\{(x_1, x_2) \in \mathbb{R}^2 | x_1 \in [0, y], x_2 = 0\}$. For example, for $y = 0.75$, Figure 2 shows these solutions (points A through B) in the F_1-F_2 space. The points lie on a straight line and are not conflicting to each other. Thus, only one point (point A with $x_1 = y = 0.75$ and $x_2 = 0$) is a feasible solution to the upper level optimization task for a fixed $y = 0.75$. Interestingly, for a fixed y, the bottom-left boundary of the F_1-F_2 space corresponds to the upper bound of x_1 or $x_1 = 1$. However, solutions having $x_1 = 1$ till $x_1 = y$ are not Pareto-optimal for the overall problem. For $y = 0.75$, solutions on line CA (excluding A) are not Pareto-optimal to both lower and upper level problems. Similarly solutions from B upwards on the '$y = 0.75$' line are also not Pareto-optimal for both levels.

When we plot all solutions for which $x_1 = y$ and $x_2 = 0$, we obtain the dotted line marked with '$x_1 = y$' in the figure. Different lower level Pareto-optimal fronts (for different y values) are shown in the figure with dashed straight lines. It is interesting to note that all solutions on this '$x_1 = y$' curve are not Pareto-optimal to the overall problem. By investigating the figure, we observe that the Pareto-optimal solutions to the upper-level problem corresponds to following solutions: $\{(x_1, x_2, y) \in \mathbb{R}^3 | x_1 = y, x_2 = 0, y \in [0.5, 1.0]\}$. This problem does not have any constraint in its lower or upper level. If an algorithm fails to find true Pareto-optimal solutions of a lower level problem and ends up finding a solution below the '$x_1 = y$' curve, such as solution C, it can potentially dominate a true Pareto-optimal point (such as point A) thereby making the task of finding true Pareto-optimal solutions difficult.

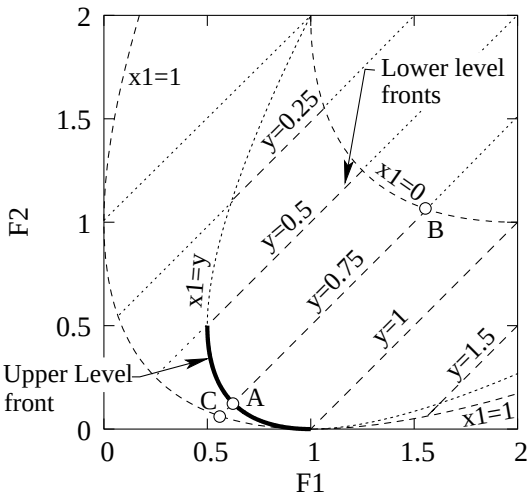

Fig. 2. Pareto-optimal fronts of upper level (complete problem) and some representative lower level optimization tasks are shown for problem 3.

5.4 Problem 4

In this problem, we increase the dimension of the variable vector of problem 3 by adding more variables like x_2:

$$\begin{aligned}
&\text{minimize} \quad \mathbf{F}(\mathbf{x}) = \left\{ \begin{array}{c} (x_1-1)^2 + \sum_{i=1}^{K} x_{i+1}^2 + y^2 \\ (x_1-1)^2 + \sum_{i=1}^{K} x_{i+1}^2 + (y-1)^2 \end{array} \right\}, \\
&\text{subject to} \\
&(x_1, x_2, \ldots, x_{K+1}) \in \text{argmin}_{(x_1,x_2,\ldots,x_{K+1})} \left\{ \mathbf{f}(\mathbf{x}) = \left(\begin{array}{c} x_1^2 + \sum_{i=1}^{K} x_{i+1}^2 \\ (x_1-y)^2 + \sum_{i=1}^{K} x_{i+1}^2 \end{array} \right) \right\}, \\
&-1 \leq x_1, x_2, \ldots, x_{K+1}, y \leq 2.
\end{aligned} \tag{7}$$

This problem has an identical Pareto-optimal front as in problem 3 with $x_i = 0$ for $i = 2, \ldots, (K+1)$, $x_1 = y$ and $y \in [0.5, 1]$. In our simulation here, we use $K = 13$, so that total number of variables is 15.

5.5 Problem 5

To test the proposed BLEMO procedure for bilevel single objective optimization problems, we include one problem from the literature [2] having $\mathbf{x}_l = (x_1, x_2, x_3, x_4)^T$ and $\mathbf{x}_u = (y_1, y_2, y_3, y_4)^T$:

$$\begin{aligned}
&\text{minimize } F(\mathbf{x}) = -(200 - x_1 - x_2)(x_1 + x_3) - (160 - x_2 - x_4)(x_2 + x_4),\\
&\text{subject to}\\
&\quad \mathbf{x}_l \in \text{argmin}_{(\mathbf{x}_l)} \big\{\mathbf{f}(\mathbf{x}) = (x_1 - 4)^2 + (x_2 - 13)^2 + (x_3 - 35)^2 + (x_4 - 2)^2 \big|\\
&\qquad g_1(\mathbf{x}) = 0.4x_1 + 0.7x_2 \le y_1,\ g_2(\mathbf{x}) = 0.6x_1 + 0.3x_2 \le y_2,\\
&\qquad g_3(\mathbf{x}) = 0.4x_3 + 0.7x_4 \le y_3,\ g_4(\mathbf{x}) = 0.6x_3 + 0.3x_4 \le y_4\big\},\\
&\quad G(\mathbf{x}) = y_1 + y_2 + y_3 + y_4 \le 40,\\
&\quad 0 \le y_1 \le 10,\ 0 \le y_2 \le 5,\ 0 \le y_3 \le 15, 0 \le y_4 \le 20,\\
&\quad 0 \le x_1 \le 20,\ 0 \le x_2 \le 20,\ 0 \le x_3 \le 40,\ 0 \le x_4 \le 40.
\end{aligned} \tag{8}$$

The reported solution to this problem [2] is $\mathbf{x}_u^* = (7.36, 3.55, 11.64, 17.45)^T$ and $\mathbf{x}_l^* = (0.91, 10, 29.09, 0)^T$ with $F(\mathbf{x}^*) = -6600.0$ and $f(\mathbf{x}^*) = 57.48$.

6 Proof-of-Principle Results

We use the following parameter settings: $N_u = 400$, $T_u = 200$, $N_l = 40$, and $T_l = 40$. Since lower level search is made interacting with the upper level search, we have run lower level optimization algorithm for a fewer generations and run the upper level simulations longer. The other NSGA-II parameters are set as follows: for SBX crossover, $p_c = 0.9$, $\eta_c = 15$ [5] and for polynomial mutation operator, $p_m = 0.1$, and $\eta_m = 20$ [4].

6.1 Problem 1

Figure 3 shows the obtained solutions using proposed BLEMO. It is clear that the obtained solutions are very close to the theoretical Pareto-optimal solutions of this problem. The lower boundary of the objective space is also shown to indicate that although solutions could have been found lying between the theoretical front and the boundary and dominate the Pareto-optimal points, BLEMO is able to avoid such solutions and find solutions very close to the Pareto-optimal solutions. Also, BLEMO is able to find a good spread of solutions on the entire range of true Pareto-optimal front. Figure 4 shows the variation of $\mathbf{x}$ for these solutions. It is clear that all solutions are close to being on the upper level constraint $G(\mathbf{x})$ boundary ($x_1 + x_2 = -1$) and they follow the relationship depicted in equation 3.

6.2 Balancing Computations Between Lower and Upper Levels

For a fixed overall population size N_u, the number of solution evaluations depends on the product $(2T_u + 1)(T_l + 1)$. Thus, a balance between T_u and T_l

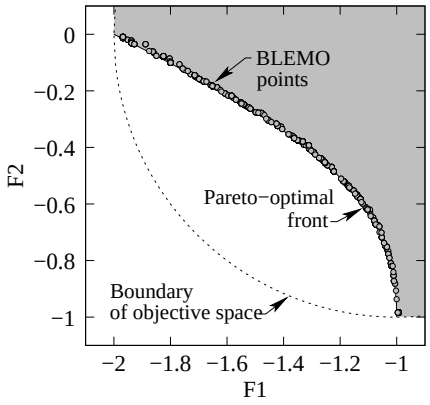

Fig. 3. BLEMO Results for problem 1.

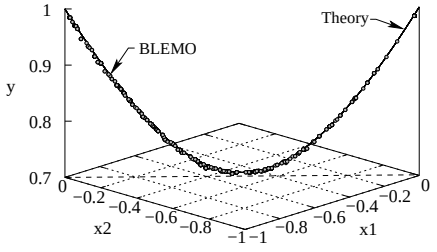

Fig. 4. Variable values of obtained solutions for problem 1. BLEMO solutions are close to theoretical results.

is needed for the overall BLEMO to work well. A too large T_l will ensure near Pareto-optimality of lower level solutions (thereby satisfying the upper level constraint better), but this will be achieved only at the expense of not having adequate upper level generations. On the other hand, a too small value of T_l means inadequate generations for the lower level task for getting close to Pareto-optimal fronts. To understand the effect of this balance between lower level and upper level computational efforts, we perform a number of simulations of our algorithm for different T_u-T_l combinations by keeping the overall solution evaluation constant. Table 1 shows the hypervolume values computed for four other T_u-T_l combinations. To not consider the effect of any non-Pareto-optimal solutions, we eliminate all solutions which lie below the theoretical Pareto-optimal curve before we compute the hypervolume. The reference point used in calculating the hypervolume is $(-1, 0)^T$. The combination $T_l = 40$ and $T_u = 200$ seems

Table 1. Hypervolume values obtained by different T_u-T_l combinations on problem 1.

T_l	T_u	Hypervolume
20	391	0.29851
40	**200**	**0.30268**
100	81	0.29716
200	41	0.28358
400	20	0.23796

Table 2. Hypervolume values obtained by different T_u-T_l combinations on problem 2.

T_l	T_u	Hypervolume
20	391	0.45034
40	**200**	**0.49256**
100	81	0.47164
200	41	0.46157
400	20	0.43145

to perform the best. It is clear that hypervolume degrades with an increase in T_l from 40. To keep the solution evaluations the same as before, T_u must be re-

duced for an increase in T_l. The use of smaller number of upper level generations is detrimental to the overall algorithm. On the other hand, when a smaller T_l (=20) is used, the performance degrades marginally, due to reduced number of lower level generations which did not allow lower level solutions to reach close to their Pareto-optimal sets.

6.3 Problem 2

This problem is more complex than the problem 1 involving non-linearities and periodic functions. We use $N_u = 600$ and $N_l = 60$ for this problem, but use identical termination conditions on generations as before. The number of subpopulation is also the same as in problem 1 and is equal to 600/60 or 10. Figure 5 shows the obtained non-dominated points. For this problem, the exact Pareto-optimal front is not known, but our front is similar to that reported in the previous study [8]. We have also plotted the solutions found by a simulation of the proposed algorithm which is run for an exorbitantly long number of generations ($N_u = 2,000$, $T_u = 400$, $N_l = 100$, $T_l = 100$). Although, our limited generation results are not exactly the same as this 'long run', the solutions are close.

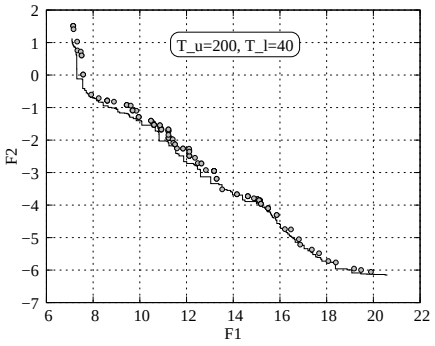

Fig. 5. Results obtained using BLEMO for problem 2.

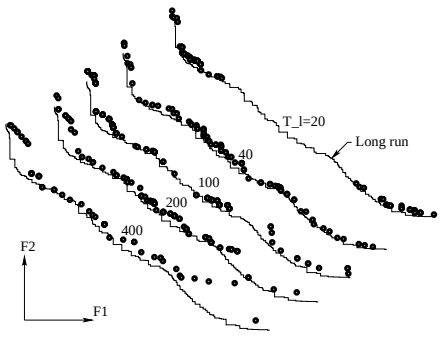

Fig. 6. Results for different T_u-T_l combinations for problem 2.

Table 2 tabulates the hypervolumes obtained using different T_u-T_l combinations. In this case also, we remove all the points which are below the F_1-F_2 points found by the 'long run'. Again, our setting of $T_u = 200$ and $T_l = 40$ is found to perform the best in terms of the hypervolume measure. Figure 6 shows the obtained solutions of different T_u-T_l combinations with respect to the 'long run' (shown in a solid line). In each case, the lower level non-Pareto-optimal solutions which are below the 'long run' front are deleted from the final front owing to being non-Pareto-optimal in the lower level. The distribution and convergence get worse with an increasing T_l value. For $T_l = 20$, there were too many

solutions which were below the 'long run', simply because these solutions were not close to Pareto-optimal front of corresponding lower level problem. However, 40 generations for the lower level search seems adequate with $T_u = 200$ in this problem as well.

6.4 Problem 3

Figure 7 shows the obtained BLEMO points on problem 3. Although solutions in between this front and the feasible boundary of objective space could have been found for an apparently better non-dominated front, these solutions would be non-Pareto-optimal with respect to the lower level problems and our algorithm has succeeded in eliminating them to appear on the final front. The figure shows that BLEMO is able to find a good distribution of solutions on the entire range of the true Pareto-optimal front. Figure 8 shows that for obtained optimal solutions, the relationship $y = x_1$ in the range $x_1 \in [0.5, 1]$ holds. Additionally, we observed that $x_2 = 0$ for all obtained solutions. These observations match with the theoretical Pareto-optimal solutions on this problem discussed in the previous section.

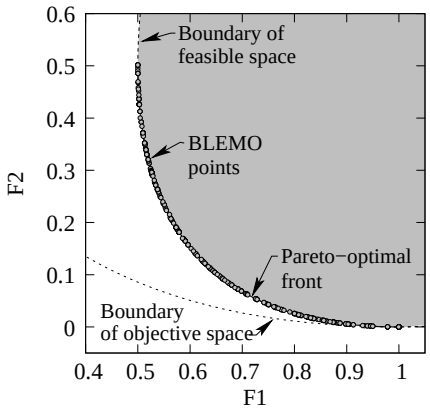

Fig. 7. Results obtained using BLEMO for problem 3.

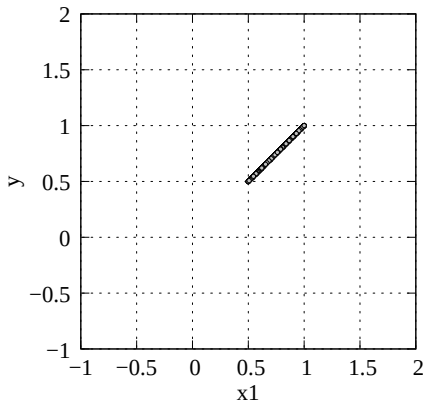

Fig. 8. Variable values of obtained solutions for problem 3.

6.5 Problem 4

In this problem, we have 15 variables. Figure 9 shows the obtained BLEMO solutions. An identical Pareto-optimal front to that in problem 3 is obtained here. For all solutions, we observed that $x_i = 0$ for $i = 2, \ldots, 14$. Again, $y = x_1$ in the range $x_1 \in [0.5, 1]$ relationship is obtained for BLEMO solutions.

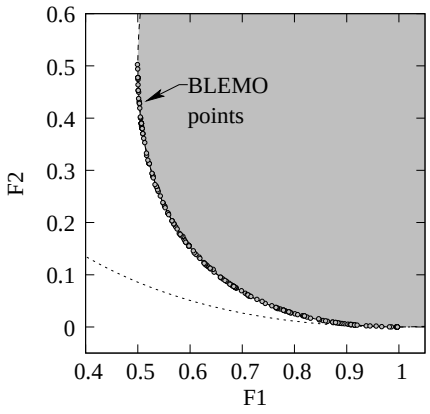

Fig. 9. Results obtained using BLEMO for problem 4.

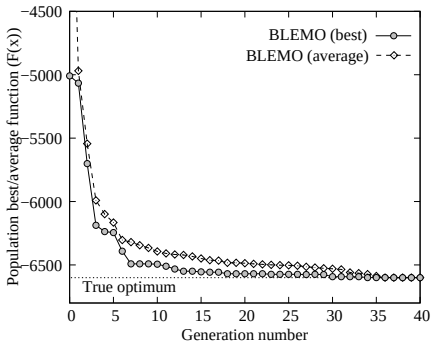

Fig. 10. Average and best function values for $F(\mathbf{x})$ are plotted with generation for problem 5.

6.6 Problem 5

For this problem, we have chosen the following parameter setting: $N_u = 400$, $T_u = 40$, $N_l = 40$ and $T_l = 40$. The obtained solution is -6599.996. The optimal variable vector is $\mathbf{x}_l = (0.9125, 9.9996, 29.0918, 0.0002)^T$ and $\mathbf{x}_u = (7.3601, 3.5516, 11.6400, 17.4520)^T$. Figure 10 shows the best and average $F(\mathbf{x})$ value with generation. This problem shows that the proposed BLEMO is able to degenerate its multi-objective operations to suit the solution of a bilevel single objective optimization problem.

7 Conclusions

In this paper, we have proposed and simulated a bilevel evolutionary multi-objective optimization (BLEMO) algorithm based on NSGA-II applied to both level problems. To coordinate the processing of populations between upper and lower levels we have maintained subpopulations having identical upper level variable values. Although any feasible solution on the upper level must correspond to the Pareto-optimal solutions of the corresponding lower level optimization problem, through simulation studies on a number of problems we have shown that the proposed interactive upper and lower level population processing strategy is able to steer the search close to the correct Pareto-optimal set of the overall problem. In this direction, we have argued and shown through a systematic parametric simulation study that a proper balance between the extent of lower and upper level generations is an important matter for an efficient use of the proposed procedure. Interestingly, we have also shown that the proposed multi-objective algorithm is also able to solve bilevel single-objective optimization problems.

This study has shown one viable way of using an existing EMO methodology for handling bilevel optimization problems. Many other ideas are certainly possible. Hopefully, this study will spur the interest in handling bilevel multi-objective optimization problems more to other interested researchers and practitioners.

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. Bilevel programming with approximation methods: Software guide and test problems. Technical report, Departement of Mathematics, Facultés Universitaires Notre-Dame de la Paix, Brussels, 2002.
3. B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operational Research*, 153:235–256, 2007.
4. K. Deb. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.
5. K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.
6. 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.
7. S. Dempe, J. Dutta, and S. Lohse. Optimality conditions for bilevel programming problems. *Optimization*, 55(56):505–524, 2006.
8. 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-Nrnberg, Germany, 2007.
9. J. Fliege and L. N. Vicente. Multicriteria approach to bilevel optimization. *Journal of Optimization Theory and Applications*, 131(2):209–225, 2006.
10. 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.
11. R. Mathieu, L. Pittard, and G. Anandalingam. Genetic algorithm based approach to bi-level linear programming. *Operations Research*, 28(1):1–21, 1994.
12. V. Oduguwa and R. Roy. Bi-level optimisation using genetic algorithm. In *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS02)*, pages 322–327, 2002.
13. L. N. Vicente and P. H. Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, 5(3):291–306, 2004.
14. 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.
15. Y. Yin. Genetic algorithm based approach for bilevel programming models. *Journal of Transportation Engineering*, 126(2):115–120, 2000.