A New Algorithm for Multilevel Optimization Problems Using Evolutionary Strategy, Inspired by Natural Adaptation

Surafel Luleseged Tilahun[1], Semu Mitiku Kassa[2], and Hong Choon Ong[1]

[1] Universiti Sains Malaysia, School of Mathematical Sciences, 11800, Penang, Malaysia
surafelaau@yahoo.com, hcong@usm.cs.my
[2] Addis Ababa University, Department of Mathematics, 1176, Science faculty, A.A., Ethiopia
smtk@math.aau.edu.et

Abstract. Multilevel optimization problems deals with mathematical programming problems whose feasible set is implicitly determined by a sequence of nested optimization problems. These kind of problems are common in different applications where there is a hierarchy of decision makers exists. Solving such problems has been a challenge especially when they are non linear and non convex. In this paper we introduce a new algorithm, inspired by natural adaptation, using (1+1)-evolutionary strategy iteratively. Suppose there are k level optimization problem. First, the leader's level will be solved alone for all the variables under all the constraint set. Then that solution will adapt itself according to the objective function in each level going through all the levels down. When a particular level's optimization problem is solved the solution will be adapted the level's variable while the other variables remain being a fixed parameter. This updating process of the solution continues until a stopping criterion is met. Bilevel and trilevel optimization problems are used to show how the algorithm works. From the simulation result on the two problems, it is shown that it is promising to uses the proposed metaheuristic algorithm in solving multilevel optimization problems.

Keywords: Multilevel optimization, (1+1)-Evolutionary strategy, metaheuristic algorithms, Natural adaptation.

1 Introduction

Many resource allocation or planning problems require compromises among the objectives of several interacting individuals or agencies; most of the time, arranged in hierarchical administrative structure and can have independent even sometimes conflicting objectives. A planner at one level of the hierarchy may have its objective function determined partly by variables controlled at other levels. Assuming that the decision process has a preemptive nature and having k levels of hierarchy, we consider the decision maker at level 1 to be the *leader* and those at lower levels to be *followers*. These kind of problems can be modeled as a nested optimization problem, referred to as multilevel programming [1]. Mathematical programming models to solve problems of these kind has been studied since 1960s, [2].

P. Anthony, M. Ishizuka, and D. Lukose (Eds.): PRICAI 2012, LNAI 7458, pp. 577–588, 2012.

Multilevel optimization analysis becomes more and more applicable in different fields. Its role in agricultural economics has been studied by Candler et. al [3] . Kocara and Outrata studied its use in engineering design [4]. Its application to transport network was also studied in [5]. Generally whenever there is a hierarchy of decision maker in such a way that each decision maker controls part of the decision variable, multilevel optimization problem model is the one suitable for the situation [6].

Due to its many applications, multilevel programming, in particular bilevel programming, has evolved significantly [7, 8]. In the late nineties Bahatia and Biegler proposed an approach with periodic property [9]. Stochastic programming like method was also proposed by Acevedo and Pistikopoulos [10]. Furthermore Pistikopoulos et. al. and other researchers proposed a new algorithm based on parametric programming theory [8, 11, 12, 13, 14]. Most of the solution methods proposed are mainly for bilevel and trilevel optimization problems with linear or convex property. The search for solution methods still continues, especially methods which are not affected by behavior of the objective functions. Perhaps metaheuristic algorithms are suitable for such purpose. That is why some recent solution methods involve metaheuristic algorithms. Among many metaheuristic algorithms evolution algorithm [15, 16, 17] and particle swarm optimization [18, 19, 20] are used in many researches and application

This paper introduces a new algorithm inspired by natural adaptation and based on (1+1)-evolutionary strategy. The format of the paper is as follows; in the next section basic concepts will be discussed followed by a discussion on the introduced algorithm in section 3. The algorithm will be tested using a bilevel and a trilevel optimization problem in section 4. At last a conclusion will be given in section 5.

2 Preliminaries

2.1 Multilevel Optimization Problem (MLOP)

Optimization problems of the following form, as in equation 1, are called k-level optimization problems.

$$
\begin{aligned}
&\underset{x_1}{Optimize} \quad f_1(x_1, x_2, ..., x_k) \\
&\quad s.t. \quad (x_1, x_2, ..., x_k) \in S_1 \subseteq \Re^n \text{ and } x_2 \text{ solves} \\
&\quad\quad \underset{x_2}{Optimize} \quad f_2(x_1, x_2, ..., x_k) \\
&\quad\quad\quad s.t \quad (x_1, x_2, ..., x_k) \in S_2 \subseteq \Re^n \text{ and } x_3 \text{ solves} \\
&\quad\quad\quad\quad \vdots \\
&\quad\quad\quad\quad\quad \underset{x_k}{Optimize} \quad f_k(x_1, x_2, ..., x_k) \\
&\quad\quad\quad\quad\quad\quad s.t. \quad (x_1, x_2, ..., x_k) \in S_k \subseteq \Re^n \\
&(x_1, x_2, ..., x_k) \in S \subseteq \Re^n \quad \text{common constraint}
\end{aligned} \tag{1}
$$

where $x_i \in \Re^{n_i} \quad \forall i \in \{1,2,...,k\}$, $n = \sum_{j=1}^{k} n_j$ and "Optimize" can be either maximize or minimize.

Generally, multilevel optimization problems (MLOP) are optimization problems which have a subset of their variables constrained to be optimal solutions of other optimization problems parameterized by the remaining variables. Depending on the number of optimization problems in the constraint set a level will be assigned. k-level optimization problem is an optimization problem which has k-1 optimization problems in the constraints.

The first optimization problem, f_1, is called leader's or level one problem and the others are followers' with level number increasing when going down. The decision maker at level j controls only x_j, whereas the other parts of the variables are controlled by the decision makers in other levels.

A point $x^* = (x_1^*, x_2^*, ..., x_k^*)$ is said to be an optimal solution for the multilevel optimization problem if x* is an optimal solution for the leader's problem, satisfying lower level problems as a constraint set. Since we have different levels and different optimization problems in each level there usually will be a conflict of objectives, hence the concept of compromise optimality needs to be defined. A compromise optimal solution is a member of the feasible set for which there doesn't exist another feasible point which does the same in all objectives and better at least in one objective function. For a given multilevel optimization problems it is possible to have many solutions depending on the decision power of the decision maker in each level. Furthermore, unlike single level optimization problems, convexity doesn't guarantee the existence of an optimal solution, and generally it is a non convex problem even when the involved functions are linear. These behaviors make multilevel optimization problems challenging compared to single level optimization problems.

2.2 Evolutionary Strategy

Evolutionary strategy is a metaheuristic algorithm which is inspired by natural evolution. It has an operator which corresponds to the mutation operator in genetic algorithm. Depending on the number of children each solution member gives, we have many kind of evolutionary strategy. In this paper we consider a (1+1)-evolutionary strategy. (1+1)-evolutionary strategy is an evolutionary strategy in which a parent gives a birth to one child [21]. (1+1)-evolutionary strategy has the following main steps:

1. Generate a random set of solutions, $\{x_1, x_2, . . ., x_m\}$
2. Move each solution member x_i, in a randomly generated direction d, $x_i' = x_i + d$. d is from a normal distribution $N(0, \delta)$, where δ is algorithm parameter.
3. Compare the performance of x_i and x_i' according to the objective function; and take the one which does better.
4. If termination criterion is not met go to step (2).

3 Metaheuristic Algorithm for MLOP

A metaheuristic algorithm is an algorithm with randomness property which tries to find a solution for optimization problems by improving the solution set iteratively. Most of these algorithms are inspired by a certain natural phenomenon. Perhaps, it is a good idea to face the challenge of multilevel optimization using metaheuristic solution methods. In this paper we introduce a new metaheuristic algorithm. The algorithm proposed in this paper uses the concept of evolutionary strategy and is inspired by natural adaptation. The leader's problem is solved for all the variables satisfying all constraint sets, as if it controls all the variables. Then that solution will adapt itself according to the objective functions in each level while going through all the levels.

Consider a k-level optimization problem shown below:

$$\begin{aligned}
&\min_{x_1} \; f_1(x_1, x_2, ..., x_k) \\
&\quad s.t. \quad (x_1, x_2, ..., x_k) \in S_1 \subseteq \Re^n \text{ where } x_2 \text{ solves} \\
&\quad\quad \min_{x_2} \; f_2(x_1, x_2, ..., x_k) \\
&\quad\quad\quad s.t \quad (x_1, x_2, ..., x_k) \in S_2 \subseteq \Re^n \text{ where } x_3 \text{ solves} \\
&\quad\quad\quad\quad \cdot \\
&\quad\quad\quad\quad\quad \cdot \\
&\quad\quad\quad\quad\quad\quad \cdot \\
&\quad\quad\quad\quad \min_{x_k} \; f_k(x_1, x_2, ..., x_k) \\
&\quad\quad\quad\quad\quad s.t. \quad (x_1, x_2, ..., x_k) \in S_k \subseteq \Re^n \\
&\quad (x_1, x_2, ..., x_k) \in S \subseteq \Re^n \quad \text{common constraint}
\end{aligned} \tag{2}$$

In the algorithm the leader's problem will be solved for $(x_1^0, x_2^0, ..., x_k^0)$ with all the constraints in all the levels including the common constraint, given as:

$$\begin{aligned}
&\min_{x_1, x_2, ..., x_k} \quad f_1(x_1, x_2, ..., x_k) \\
&\quad s.t. \quad (x_1, x_2, ..., x_k) \in \overline{S_1} \\
&\text{where } \overline{S_1} = S \cap S_1 \cap S_2 \cap ... \cap S_k
\end{aligned} \tag{3}$$

This implies that $f_1(x_1^0, x_3^0, ..., x_k^0) \leq f_1(x_1, x_2, ..., x_k) \quad \forall (x_1, x_2, ..., x_k) \in \overline{S_1}$. Then by fixing x_1^0 we solve the second level problem and update $(x_2^0, x_3^0, ..., x_k^0)$ as shown in equation (4).

$$\begin{aligned}
&\min_{x_2, x_3, ..., x_k} \quad f_1(x_1^0, x_2, ..., x_k) \\
&\quad s.t. \quad (x_1^0, x_2, ..., x_k) \in \overline{S_2} \\
&\text{where } \overline{S_2} = S \cap S_2 \cap S_3 \cap ... \cap S_k
\end{aligned} \tag{4}$$

Generally for any level i we will solve the corresponding problem using evolutionary strategy, as shown in (5).

$$\begin{aligned} \min_{x_i, x_{i+1}, \ldots, x_k} \quad & f_1(x_1^0, x_2^0, \ldots, x_{i=1}^0, x_i, x_{i+1}, \ldots, x_k) \\ s.t. \quad & (x_1^0, x_2^0, \ldots, x_{i=1}^0, x_i, x_{i+1}, \ldots, x_k) \in \overline{S_i} \\ \text{where } \overline{S_i} = & S \cap S_i \cap S_{i+1} \cap \ldots \cap S_k \end{aligned} \tag{5}$$

Once the k^{th} level problem is solved, then $(x_2^0, x_3^0, \ldots, x_k^0)$ will be used as parameters to solve the problem given in equation (6) for x_1^1.

$$\begin{aligned} \min_{x_1} \quad & f_1(x_1, x_2^0, \ldots, x_k^0) \\ s.t. \quad & (x_1, x_2^0, \ldots, x_k^0) \in S_1 \\ & (x_1, x_2^0, \ldots, x_k^0) \in S \end{aligned} \tag{6}$$

Similarly, to solve the second level for x_2^1 we fix $(x_1^1, x_3^0, x_4^0, \ldots, x_k^0)$ and use evolutionary strategy. Once x_2^1 is computed we fix $(x_1, x_2, x_4, \ldots, x_k)$ as $(x_1^1, x_2^1, x_4^0, \ldots, x_k^0)$ and solve for $x_3 = x_3^1$. By continuing in similar way for all the levels down, at last $(x_2^1, x_3^1, \ldots, x_k^1)$ will be fixed to solve for the first level problem for $x_1 = x_1^2$. This process will continue until a termination criterion is fulfilled. It means at j^{th} iteration and optimizing i^{th} level we will have the following optimization problem:

$$\begin{aligned} \min_{x_i} \quad & f_i(x_1^j, x_2^j, \ldots, x_{i-1}^j, x_i, x_{i+1}^{j-1}, \ldots, x_k^{j-1}) \\ s.t. \quad & (x_1^j, x_2^j, \ldots, x_{i-1}^j, x_i, x_{i+1}^{j-1}, \ldots, x_k^{j-1}) \in S'_i \\ & (x_1^j, x_2^j, \ldots, x_{i-1}^j, x_i, x_{i+1}^{j-1}, \ldots, x_k^{j-1}) \in S^i \end{aligned} \tag{7}$$

where S'_i is the i^{th} level constraint set with all the other variables are fixed and S^i is also the common constraint with all the variables, except variable i, are fixed.

At each step (1+1)-evolutionary algorithm will be used with a property of passing the previous solution. It means, suppose we are solving the i^{th} level problem at a particular iteration j. $(x_1^j, x_2^j, \ldots, x_{i-1}^j, x_{i+1}^{j-1}, \ldots, x_k^{j-1})$ is fixed then when evolutionary strategy is used x_i^{j-1} will be taken as a member among the randomly generated initial solution set. This will help the algorithm not to move away from a good solution because of the conflict of objective functions.

The algorithm is summarized in the following tables:

Table 1. The algorithm

Input: $f_i(x_1, x_2, ..., x_k), S, S_i \quad \forall i \in \{1,2,...,k\}$
for p=1:k
  $(x_1^0, x_2^0, \ldots, x_k^0)$ == EvolutionaryStrategy $(f_p(x_1^0, x_2^0, \ldots x_{p-1}^0, x_p, \ldots, x_k), S_p, S, n_p)$
  where $S_p = S_p \cap S_{p+1} \cap \ldots \cap S_k$
end
for j=1:MaxGen
 for i=1:k
  if (i >1)
   x_i^j == EvolutionaryStrategy $(f_i(x_1^j, x_2^j, \ldots, x_{i-1}^j, x_i, x_{i+1}^{j-1} \ldots, x_k^{j-1}), S_i, S, n_i, x_i^{j-1})$
  else
   x_1^j == EvolutionaryStrategy $(f_1(x_1, x_2^{j-1}, \ldots, x_k^{j-1}), S_1, S, n_1, x_1^{j-1})$
  end if
 end for
$(x_1^*, x_2^*, ..., x_k^*) = (x_1^j, x_2^j, ..., x_k^j)$
 terminate if termination criteria is fulfilled
end for
Output: $(x_1^*, x_2^*, ..., x_k^*)$

Table 2. Evolutionary strategy with passing a solution, EvolutionaryStrategy $(f(x), S_1, S, n, x')$

Input: $f(x)$, S_1, S, n, x'
Algorithm Parameter: δ
Do:
  Randomly generate m-1 solutions for x from the feasible region, say x_1, x_2, …, x_{m-1}.
  Put $x_m = x'$ (*if* x' *is given, else generate* x_m *also randomly*)
 for i=1:m
  Generate d from $N(0, \delta)$
  $x_i^* = x_i + d$
   Check feasibility
   if ($f(x_i^*) \le f(x_i)$)
    $x_i = x_i^*$
  end if
end for
Repeat until termination criteria is met
$x^* = x_j$, such that $f(x_j) \le f(x_i) \quad \forall i \in \{1,2,...,m\}$
Output: x^*

4 Simulation Examples

To demonstrate the algorithm we use a bilevel and a trilevel optimization problems.

a) Bilevel Example

The bilevel problem is taken from a book chapter [22]. It is as given in equation (8). After solving the problem using the algorithm, the solution is compared to the solution given in the book. According to the book the solution is (x', y') = (0.609, 0.391, 0, 0, 1.828), with values 0.6429 and 1.6708 for f_1 and f_2, respectively.

$$\begin{aligned}
&\min_{x} f_1(x,y) = y_1^2 + y_3^2 - y_1 y_3 - 4y_2 - 7x_1 + 4x_2 \\
&s.t \quad x_1 + x_2 \le 1 \\
&\qquad \min_{y} f_2(x,y) = y_1^2 + \frac{1}{2}y_2^2 + \frac{1}{2}y_3^2 + y_1 y_2 + (1-3x_1)y_1 + (1+x_2)y_2 \\
&\qquad s.t \quad 2y_1 + y_2 - y_3 + x_1 - 2x_2 + 2 \le 0 \\
&\qquad\qquad x \ge 0 \\
&\qquad\qquad y \ge 0
\end{aligned} \tag{8}$$

The algorithm parameter δ was set to be 1 and number of initial solutions, m, was set to be 50. Furthermore the number of iteration was set to be 30 for the evolutionary strategy and 50 for the algorithm.

First the leader's problem is solved with all the constraints as shown in equation (9)

$$\begin{aligned}
&\min_{x,y} f_1(x,y) = y_1^2 + y_3^2 - y_1 y_3 - 4y_2 - 7x_1 + 4x_2 \\
&s.t \quad x_1 + x_2 \le 1 \\
&\qquad 2y_1 + y_2 - y_3 + x_1 - 2x_2 + 2 \le 0 \\
&\qquad x \ge 0 \\
&\qquad y \ge 0
\end{aligned} \tag{9}$$

Using (1+1)-evolutionary strategy, the solution for the problem in equation (9) is found to be x^0= (0.2756 0.7117) and y^0=(0.0167 1.1095 2.0073) with $f_1(x^0, y^0)$ = 0.4763.

Then by fixing x^0 the second level problem is solved.

$$\begin{aligned}
&\min_{y} f_2(x^0,y) = y_1^2 + \frac{1}{2}y_2^2 + \frac{1}{2}y_3^2 + y_1 y_2 + (1-3x_1^0)y_1 + (1+x_2^0)y_2 \\
&s.t \quad 2y_1 + y_2 - y_3 + x_1^0 - 2x_2^0 + 2 \le 0 \\
&\qquad x_1^0, x_2^0 \ge 0 \\
&\qquad y \ge 0
\end{aligned} \tag{10}$$

Then equation (10) is solved using evolutionary strategy for y^1, in such a way that y^0 will be taken as one of the initial solutions in the evolutionary strategy. After

y^1 is computed, it will be fixed to solve the leader's problem for x^1, again using evolutionary strategy with passing the previous best, x^0, as one of the initial population member.

$$\begin{aligned} \min_{x} \; & f_1(x, y^1) = (y_1^1)^2 + (y_3^1)^2 - y_1^1 y_3^1 - 4y_2^1 - 7x_1 + 4x_2 \\ s.t \quad & x_1 + x_2 \le 1 \end{aligned} \tag{11}$$

After that the second problem will be solved for y^2 by fixing x^1 and passing y^1 as a member of initial solutions in the evolutionary strategy. This pattern repeats itself until a preset iteration number, which in our case is 50, is reached.

After running the program using matlab the optimal solution was found to be (x^*, y^*) = (0.5307 0.4683 0.0012 0.0002 1.5989), with 0.7122 and 1.2778 for f_1 and f_2, respectively. The performance of the algorithm compared to the solution in the book is presented in Figure (1). Furthermore, it compares the performance in terms of the sum of the functional values, $f_1(x',y')+f_2(x',y')$ and $f_1(x^*,y^*)+f_2(x^*,y^*)$.

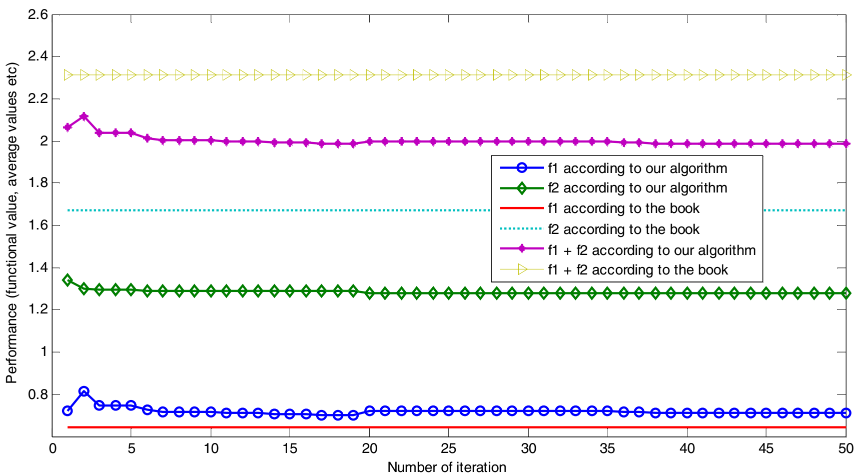

Fig. 1. Performance of the algorithm in the bilevel problem

From the result it is clear that if equal weight is given to the objective functions the algorithm performs better because $f_1(x',y')+f_2(x',y')$ = 2.3134 and $f_1(x^*,y^*)+f_2(x^*,y^*)$ = 1.9900, where x' is the solution from the book and x* is the solution after running the algorithm.

b) Trilevel Example

The second test problem is a trilevel optimization problem taken from a thesis done by Molla [23]. It is given in equation (12)

$$\begin{aligned}
&\min_{x} f_1(x,y,z) = -x + 4y \\
&\quad s.t. \quad x + y \le 1 \quad \text{and y solves} \\
&\qquad \min_{y} f_2(x,y,z) = 2y + z \\
&\qquad s.t. \quad -2x + y \le -z \quad \text{and z solves} \\
&\qquad\quad \min_{z} f_3(x,y,z) = -z^2 + y \\
&\qquad\quad s.t. \quad z \le x \\
&\left.\begin{aligned} 0 \le x \le 0.5 \\ 0 \le y, z \le 1 \end{aligned}\right\} \text{is common constraint}
\end{aligned} \tag{12}$$

The algorithm parameter and number of iterations are the same in the previous case. According to the algorithm first the leader's problem, as shown in equation (13), is solved.

$$\begin{aligned}
&\min_{x,y,z} f_1(x,y,z) = -x + 4y \\
&s.t. \quad x + y \le 1 \\
&\quad -2x + y \le -z \\
&\quad z \le x \\
&\quad 0 \le x \le 0.5 \\
&\quad 0 \le y, z \le 1
\end{aligned} \tag{13}$$

After running the code the solution is found to be (x^0, y^0, z^0) = (0.4999 0.0003 0.1687). Now by fixing x^0 we solve for the problem in equation (14).

$$\begin{aligned}
&\min_{y,z} f_2(x^0,y,z) = 2y + z \\
&s.t. \quad -2x^0 + y \le -z \\
&\quad z \le x^0 \\
&\quad 0 \le x^0 \le 0.5 \\
&\quad 0 \le y, z \le 1
\end{aligned} \tag{14}$$

The solution of equation (14) is 10^{-3}(0.1097 0.2957), with f_2(0.4999, (10^{-3})(0.1097), (10^{-3})(0.2957))=10^{-4} (5.1513). Hence (x^0, y^0, z^0) is updated to (0.4999, (10^{-3})(0.1097), (10^{-3})(0.2957)).

Afterwards we fix (x^0, y^0) and solve the third level problem for z^0.

$$\begin{aligned}
&\min_{z} f_3(x^0,y^0,z) = (-z)^2 + y^0 \\
&s.t. \quad z \le x^0 \\
&\quad 0 \le x^0 \le 0.5 \\
&\quad 0 \le y^0, z \le 1
\end{aligned} \tag{15}$$

z^0 is updated to be 0.4978. Hence (x^0, y^0, z^0) is computed, then y^0 and z^0 will be fixed using evolutionary strategy to solve for x^1, with x^0 as one of the solution candidate for the evolutionary strategy algorithm. After running the matlab code the result becomes $(x^*, y^*, z^*) = (0.5, 0, 0.0095)$. And $f_1(x^*, y^*, z^*) = -0.5$, $f_2(x^*, y^*, z^*) = 0.0127$ and $f_3(x^*, y^*, z^*) = -0.2499$. The performance of the functional values as a function of iteration number is shown in the graph below:

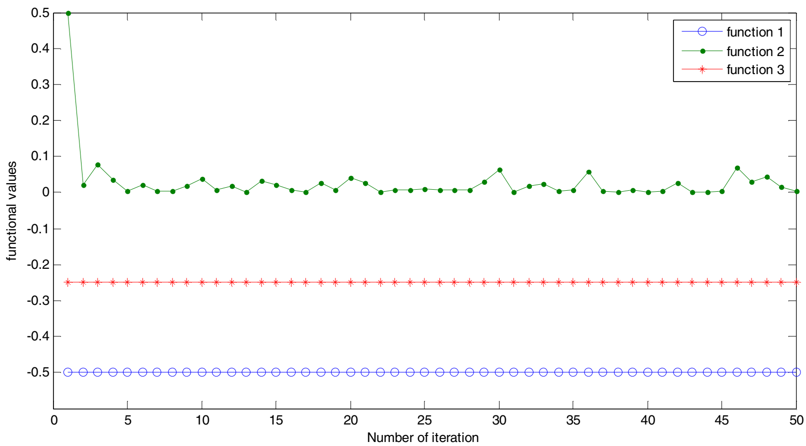

Fig. 2. Performance of the algorithm in the trilevel problem

Our result is better than the result reported by Molla [17], which is $(x_m, y_m, z_m)=(0.5, 1, 1)$. This implies that $f_1(x_m, y_m, z_m)=3.5$, $f_2(x_m, y_m, z_m)=3$ and $f_3(x_m, y_m, z_m)=0$, which are worse in sense of minimization when compared to our result not only in average but also for each individual objective functions.

5 Conclusion

In this paper a new metaheuristic algorithm for multilevel optimization problem which mimic the concept of natural adaptation is introduced. The algorithm uses (1+1)-evolutionary strategy to solve one of the level's optimization problem at once. The leader's problem will be solved for all the variables satisfying all the constraint sets in all the levels. That solution will go through each level iteratively by adapting itself with the corresponding objective function and constraint set of each level. In each iteration the solution will respond to the change in the parameters and tries to update itself compared to the previous solution and the new parameters. (1+1)-evolutionary strategy is used in the updating process with a property of considering the previous best as a candidate solution for the current stage. These updating will continue iteratively until termination criterion is met. From the simulation results on a bilevel and trilevel optimization problem, it is shown that the algorithm gives a promising result for multilevel optimization problems.

Acknowledgements. This work is in part supported by Universiti Sains Malaysia short term grant number 304/PMATHS/6311126. The first author would like to acknowledge a support from TWAS-USM fellowship.

References

1. Bard, J.F., Falk, J.E.: An explicit solution to the multilevel programming problem. Computers and Operations Research 9(1), 77–100 (1982)
2. Dantzig, G.B., Wolfe, P.: Decomposition Principle for Linear Programs. Operations Research 8(1), 101–111 (1960)
3. Candler, W., Fortuny-Amat, J., McCarl, B.: The potential role of multilevel programming in agricultural economics. American Journal of Agricultural Economics 63, 521–531 (1981)
4. Vincent, N.L., Calamai, H.P.: Bilevel and multilevel programming: A bibliography review. Journal of Global Optimization 5, 1–9 (1994)
5. Marcotte, P.: Network design problem with congestion effects: A case of bilevel programming. Mathematical Programming 43, 142–162 (1986)
6. Rao, S.S.: Engineering Optimization: theory and practice, 4th edn. John Wiley & Sons Inc. (2009)
7. Faisca, P.N., Dua, V., Rustem, B., Saraiva, M.P., Pistikopoulos, N.E.: Parametric global optimization for bilevel programming. Journal of Global Optimization 38, 609–623 (2007)
8. Pistikopoulos, N.E., Georgiads, C.M., Dua, V.: Multiparametric Programming: theory, algorithm and application. Wiely-Vich Verlag Gmbh and Co. KGaA (2007)
9. Bahatia, T.K., Biegler, L.T.: Multiperiod design and planning with interior point method. Computers and Chemical Engineering 23(14), 919–932 (1999)
10. Acevedo, J., Pistikopoulos, E.N.: Stochastic optimization based algorithms for process synthesis under uncertainty. Computer and Chemical Engineering 22, 647–671 (1998)
11. Dua, V., Pistikopoulos, E.N.: An algorithm for the solution of parametric mixed integer linear programming problems. Annals of Operations Research 99(3), 123–139 (2000)
12. Dua, V., Bozinis, N.A., Pistikopoulos, E.N.: A multiparametric programming approach for mixed-integer quadratic engineering problem. Computers and Chemical Engineering 26(4/5), 715–733 (2002)
13. Pistikopoulos, E.N., Dua, V., Ryu, J.H.: Global optimization of bilevel pro-gramming problems via parametric programming. Computational Management Science 2(3), 181–212 (2003)
14. Li, Z., Lerapetritou, M.G.: A New Methodology for the General Multi-parametric Mixed-Integer Linear Programming (MILP) Problems. Industrial & Engineering Chemistry Research 46(15), 5141–5151 (2007)
15. Wang, Y., Jiao, Y.C., Li, H.: An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. IEEE Transactions on Systems, Man and Cybernetics Part C 35(2), 221–232 (2005)
16. Deb, K., Sinha, A.: Solving Bilevel Multi-Objective Optimization Problems Using Evolutionary Algorithms. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 110–124. Springer, Heidelberg (2009)
17. Sinha, A.: Bilevel Multi-objective Optimization Problem Solving Using Progressively Interactive EMO. In: Takahashi, R.H.C., Deb, K., Wanner, E.F., Greco, S. (eds.) EMO 2011. LNCS, vol. 6576, pp. 269–284. Springer, Heidelberg (2011)

18. Kuo, R.J., Huang, C.C.: Application of particle swarm optimization algorithm for solving bi-level linear programming problem. Computers & Mathematics with Applications 58(4), 678–685 (2009)
19. Gao, Y., Zhang, G., Lu, J., Wee, H.M.: Particle swarm optimization for bi-level pricing problems in supply chains. Journal of Global Optimization 51, 245–254 (2011)
20. Zhang, T., Hu, T., Zheng, Y., Guo, X.: An Improved Particle Swarm Optimiza-tion for Solving Bilevel Multiobjective Programming Problem. Journal of Applied Mathematics 2012, Article ID 626717 (2012), doi:10.1155/2012/626717
21. Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent System. Henry Ling Limited, Harlow (2005)
22. Faisca, N.P., Rustem, B., Dua, V.: Bilevel and multilevel programming, Multi-Parametric Programming. WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim (2007)
23. Molla, A.: A multiparametric programming approach for multilevel optimization, A thesis submitted to the department of mathematics, Addis Ababa University (2011)