# Bi-level Optimisation using Genetic Algorithm

V. Oduguwa and R. Roy
*Department of Enterprise Integration, School of Industrial and Manufacturing Science, Cranfield University, Cranfield, Bedford, MK43 0AL, UK*
*Email: v.oduguwa and r.roy@cranfield.ac.uk*

## Abstract

*In most real-life problems such as rolling system design decision-making can be hierarchical and the search space is unknown. Bilevel optimisation problem (BLP) is an operation research technique for solving real life hierarchical decision-making problems. There are a number of different algorithms developed based on classical optimisation methods to solve different classes of the BLP problems where the search space is known. There also exist a number of problems in the BLP which current algorithms are not sufficiently robust to solve. In this paper, Bi-level Genetic Algorithm (BiGA) is a new proposed to solve different classes of the BLP problems within a single framework. BiGA is an elitist optimisation algorithm developed to encourage limited asymmetric cooperation between the two players. The performance of the algorithm is illustrated using test functions. The results suggest that BiGA algorithm is robust to solve different classes of the BLP problems and demonstrates potential for real life problems.*

## 1. Introduction

Decision-making in most real life problems such as rolling system design can be made in hierarchical order and in most cases the search space is unknown. The bi-level optimisation problem (BLP) is a nested optimisation problem with two levels in a hierarchy, the higher and lower level decision-makers.

The BLP problems are generally difficult to solve due to the non-convex nature of the search space resulting from the complex interaction of the leader and follower problem. A number of various algorithms have been developed based on classical optimisation approaches such as variable elimination method based on Kuhn Tucker approach [1-3] and algorithms based on penalty function approach [4-6]. Literature suggests that most of the traditional solutions are problem dependent relying on knowledge of the search space and are not sufficiently robust methods to solve real life problems [7]. Heuristic approaches are now generating interest in the research community as an alternative for solving the BLP problems. Anadalingam et al [8] developed a Simulated Annealing Based Bi-level programming Algorithm (SABBA), Mathieu et al [7] developed a Genetic Algorithm Based Bi-level programming Algorithm (GABBA), Gendreau et al [9] proposed a hybrid Tabu Search ascent algorithm and recently Yin [10] proposed an approach (GAB) based on genetic algorithm. Most of the computational results of the heuristic approaches reported in the literature are still not as satisfying when compared to exact methods such as the Kuhn Tucker approach or the penalty method and also they are reported to be computationally expensive [11].

This study is carried out as part of an overall research interest in developing robust solutions for real life industrial problems using evolutionary computing techniques[12, 13]. In this paper, a bi-level genetic algorithm (BiGA) is proposed for solving BLP problems within a single framework. The motivation is to develop a search algorithm that identifies the global optimal region in a robust manner for different classes of BLP.

This paper is organised as follows: Section 2 discusses the characteristics and formulation of the BLP, and section 3 states the various classifications of the problem. Challenges in solving these problems are explored in section 4 and details of the BiGA algorithm given in section 5 with the results and performance analysis given in section 6. Section 7 discusses the limitations and future work and the conclusion is given in section 8.

## 2. Bi-level optimisation problem

The bi-level programming problems are a nested optimisations problem characterised by the decision-maker at one level influencing the behaviour of a decision - maker at another level. Thus an important feature of the bi-level programming problem is that the objective functions of each unit may be partially determined by variables controlled by other units operating at other levels. While the decisions made at lower level are not dictated by their superiors, their reactions have influence at the upper levels and have the tendency to improve their own objectives. The common characteristics of the bi-level programming are stated as follows:

- The decision-making units are interactive and exist within a predominantly hierarchical structure.
- Decision-making is sequential from higher to lower level. The lower level decision-maker executes its policies after decisions are made at the upper level.
- Each unit independently optimises its own objective functions but is influenced by actions taken by other units.

The general form of the BLP is defined as follows [14]: Let the vector of decision variables $(x, y) \in \Re^n$ be partitioned among the two planners. Where the higher level decision maker has control over the vector $y \in \Re^{n_1}$ ,and the lower level decision maker vector has control over the $x \in \Re^{n_2}$ , where $n_1 + n_2 = n$.

$$L : \max_{x} F_L(x, y) = ax + by, \text{ where } y \text{ solves}:$$

$$\text{F} : \max_{y} \text{F}_\text{F}(x, y) = cx + dy, \quad s.t: \quad \text{A}x + \text{B}y \le \text{r},$$

where $\text{a, c} \in \Re^{\text{n}_1}$, $b$, $d \in \Re^{n_2}, r \in \Re^m$, A is an $\text{m x n}_1$ *matrix*, B is an $\text{m x n}_2$ matrix. The problem constraint region is denoted by S. The higher level decison maker (L) who control over x, makes his decision first which fixes $x$ before the lower level decision maker (F) selects y. For a given $x$, the set $\text{Y}(x)$ of optimal solutions to the lower problem (F) can be stated as : $\max_{y \in \text{Q(x)}} f(y) = dy$,

where $\text{Q}(x) = \{y \mid \text{B}y \le \text{r - A}x\}$.

**Table 1: Classification of the BLP problems**

| **T1: Linear leader and linear follower** | **T2: Linear leader and non-linear follower** |
|---|---|
| $\text{L} : \min_{x} f_\text{L} = \text{a}x + \text{b}y$<br>$\text{F} : \min_{y} f_\text{F} = \text{c}x + \text{d}y$ | $\text{L} : \min_{x} f_\text{L} = \text{a}x + \text{b}y$<br>$\text{F} : \min_{y} f_\text{F} = \text{p}x^j + \text{q}y^k + rx + sy$ |
| **T3: Non-linear leader and linear follower** | **T4: Non-linear leader and non-linear follower** |
| $\text{L} : \min_{x} f_\text{L} = \text{a}x^n + \text{b}y^m + cx + dy$<br>$\text{F} : \min_{y} f_\text{F} = \text{p}x + \text{q}y$ | $\text{L} : \min_{x} f_\text{L} = \text{a}x^n + \text{b}y^m + cx + dy$<br>$\text{F} : \min_{y} f_\text{F} = \text{p}x^j + \text{q}y^k + rx + sy$ |
| **T5: Parametric BLP problem**<br>This is a special case of the BLP problem where the follower problem is implicitly contained in the leader problem.<br>$\text{L} : \min_{x} f_\text{L} = \text{a}x + \text{b}y + cx + dy$<br>$\text{F} : \min_{y} f_\text{F} = \text{p}x + \text{q}y \quad \textit{where } p = \text{c}, \text{and q} = \text{d}$ | |
| L: Leader, F: Follower | |

## 3. Classification of the BLP Problems

In this section, the classification of the BLP problem is developed. Since the BLP problem involves two players, and the nature of each player can either be linear or non-linear. The complexity of the overall problem depends on the combination of the nature of the each player. Non-linearity is characterised by the quadratic or the high order polynomial nature of the problem. Table 2 shows a brief classification of the BLP problem. In most of the cases the resulting search space is complex, non-convex and multi-modal. This study focuses on types T1, T4 and T5.

## 4. Challenges in Solving the BLP Problem

Two main challenges facing the BLP problems discussed in this paper are: limitations due to classical solution approaches and the follower multi-modality problem.

Some of the limitations due to classical solution approaches are due to inherent non-convexity and non-differentiability of the complex search space. Although algorithms exist that addresses these problems, most find local rather than global optimal. Approaches such as Kuhn-Tucker method handles convexity in different ways but successes have been limited [11].

The follower multi-modality problem occurs when the follower's hyper-plane is multi-modal and is indifferent to any point in its search space. The leader on the other hand might not enjoy this luxury, and the follower may be required to make a particular move for the leader to achieve its optimal solution but is not induced to do so. This problem often results to sub-optimal solutions in the leader problem. The follower multi-modality problem has received little interest in the multilevel programming research community [7]. This could be due to the inherent limitations of current techniques in dealing with this problem. Since the follower plays an integral part in influencing the optimality condition of the leader problem, it is imperative to develop algorithms to deal with this problem.

## 5. BiGA: A Proposed Solution Algorithm

A Bilevel genetic algorithm is proposed as an alternative to conventional approaches to find global solutions. In the sections a brief overview of the GA is given and the proposed BiGA described with some of the unique features discussed.

### 5.1. Genetic Algorithms

Genetic Algorithms (GA's) are adaptive methods used to solve search and optimisation problems, based on genetic processes of biological organisms [15]. They simulate the genetic state of a population of individuals through operators such as natural selection, mutation, and crossover. GA is different from the classical optimisation methods in the following ways:

- GA's search by using a population of points and not single point search by the classical methods. As a result of the parallel search GA's are effective in finding the global optimum.
- GA's allows coding of the parameter set enabling GA to be used for different problems.
- GA use randomness to guide its search towards the optimal region.

Literature reveals that GA based optimisation techniques are effective for finding the global optimum region due to its inherent parallelism. They do not require the objective function to be differentiable and are quite robust in dealing with non-convex problems. As a result of these well-established properties the GA was adopted as the basic solution approach.

### 5.2. The BiGA algorithm

BiGA is a sequential nested optimisation algorithm developed to encourage limited asymmetric cooperation between the two players. The algorithm solves two optimisation problems iteratively, one for the leader in all the $\boldsymbol{x}$ variables and a subset of the $\boldsymbol{y}$ variables associated with the optimal basis follower's problem, and the other for the follower problem with all the $\boldsymbol{x}$ variables fixed. The optimal basis of the follower problem is explored with $\boldsymbol{x}$ fixed and then returned to the leader problem with the corresponding $\boldsymbol{y}$ variables. The algorithm uses the full operators in the basic genetic algorithm in a dual population environment to solve both leader and the follower problem. Tournament selection is used to select fit individuals. The basic flowchart of the BiGA algorithm is shown in figure 1. The subscript 'L' refers to leader, 'F' the follower and t refers to generation at time t.

### 5.3. Description of BiGA

The solution strategy is encoded in C++ using the proposed algorithm 'Bilevel Genetic Algorithm (BiGA)' shown in figure 1. The pseudo code in figure 2 outlines the steps involved in the BiGA algorithm and a brief description of the key features is given.

After the initialisation of the leader population ($P^L$), the follower members represented in the leader problem are copied from the follower population ($P^F$) into the leader population. Selection, crossover and mutation operations are similar to conventional simple GA technique. The sort operator is designed to give a larger selective preference to the fitter members of the population when the co-evolutionary operator is invoked. It sorts the population in descending order of fitness with the fittest member towards the front end of the population and the less fitter members towards the back end of the population.

The co-evolutionary operator is used to preserve the interactive nature of the decision-making units intrinsic in the BLP problem in the search process. This property is imperative for maintaining the evolutionary process given the structure of the BLP problem. The co-evolutionary operator synchronizes each player's population with members of the other player's population.

An external elite population is maintained to identify the elite members of both populations after the co-evolutionary operator for every generation. Members in the elite population are replaced with the best members from current generation if best members are fitter than the elites.

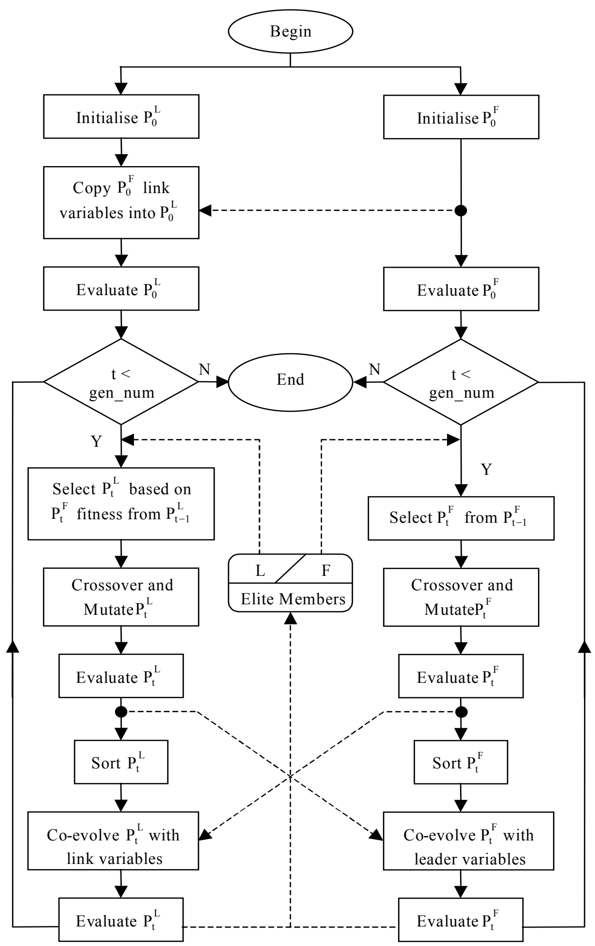


**Figure 1: Bi-level Genetic Algorithm (BiGA)**

| procedure $GA^L$; | procedure $GA^F$; |
|---|---|
| begin;<br>randomly initialise $P_0^L$;<br>copy link members to $P_0^L$;<br>evaluate leader population;<br>t=1;<br>repeat<br>  add best leader member;<br>  select $P_t^L$ from $P_{t-1}^L$;<br>  crossover and mutate $P_t^L$;<br>  evaluate leader population;<br>  sort $P_t^L$;<br>  co-evolute $P_t^L$ with link members;<br>  evaluate leader population;<br>  update global best member;<br>  t=t+1;<br>until (termination condition);<br>end; | begin;<br>randomly initialise $P_0^F$;<br>evaluate leader population;<br>t=1;<br>repeat<br>  add best follower member;<br>  select $P_t^F$ from $P_{t-1}^F$;<br>  crossover and mutate $P_t^F$;<br>  evaluate leader population;<br>  sort $P_t^F$;<br>  co-evolute $P_t^F$ with leader members;<br>  evaluate leader population;<br>  update global best member;<br>  t=t+1;<br>until (termination condition);<br>end; |

**Figure 2: Pseudo code for the BiGA algorithm**

### 5.4. Search strategy

The selection strategy is developed to encourage limited asymmetric cooperation between the two players. Limited cooperation in the sense that optimality conditions are not determined on the basis of non-domination criteria as is common with multi-objective problems. Asymmetric cooperation is implied by a mono-directional cooperation. The follower cooperates with the leader but not otherwise. Cooperation is only allowed amongst the fitter individuals (evaluated using the objective function space) who are able to satisfy their own local objectives. This serves as the incentive to guide the search towards the optimal region. (see section 5.5 discussion on optimality criteria).

**Table 2: Test problem adopted in the study**

| Problem | Variable Bounds | Objective Functions | Related Features |
|---|---|---|---|
| F1 | $x \in [0,1]$<br>$y \in [0,1]$<br>$= 1,2$ | $\max_x \; f_L(x, y) = 100x + 1000y_1$<br>$\max_{y_1 y_2} \; f_F(x, y) = y_1 + y_2$<br>*s.t.*<br>$x + y_1 - y_2 \le 1, \; y_1 + y_2 \le 1$<br>$x, y_1, y_2 \ge 0.$ | • Linear upper problem<br>• Linear lower problem<br>• Linear multiple constraints<br>• Multi-modal lower problem |
| F2 | | $\min_x \; f_L(x, y) = (x-1)^2 + (y-1)^2$<br>$\min_y \; f_F(x, y) = 0.5y^2 + 500y - 50xy$ | • Non-linear upper problem<br>• Polynomial lower problem |
| F3 | $x \in [0,15]$<br>$y \in [0,20]$ | $\min_x \; f_L(x, y) = x^2 + (y-10)^2$<br>$-x + y \le 0$<br>$\min_y \; f_F(x, y) = (x + 2y - 30)^2$<br>$x + y \le 20$ | • Non-linear upper problem<br>• Non-linear lower problem<br>• Single constraints at each level |
| F4 | | $\min_x \; f_L(x, y) = (x-1)^2 + 2y - 2x$<br>$\min_{y_1 y_2} f_F(x, y) = (2y_1 - 4)^2 + (2y_2 - 1)^2 + xy_1$<br><br>$4x + 5y_1 + 4y_2 \le 12, \; -4x - 5y_1 + 4\;_2 \le -4$<br>$4x - 4y_1 + 5y_2 \le 4, \; -4x + 4y_1 + 5y_2 \le 4$<br>$x \ge \;, y_i \ge 0, \; = 1,2$ | • Non-linear upper problem<br>• Non-linear lower problem<br>• Multiple constraints level |


0-7695-1733-1/02 $17.00 © 2002 IEEE

### 5.5. Optimality criteria for the BLP problem

The optimality criteria in this algorithm is based on the directional search of the follower problem $\boldsymbol{y}(\boldsymbol{x})$.

Optimality is obtained by using optimal value of the lower-level problem in the upper level and by solving the resulting upper-level problem. The lower-level problem is viewed as a parameter program where the optimal solution $\boldsymbol{y}(\boldsymbol{x})$ depends on the parameter $\boldsymbol{x}$. This approach was adopted from the work of Savard and Gauvin [16, 17]. Their work relates to the specific case in which the optimal value of the lower-level problem is used in the upper-level objective function.

### 5.6. Representation of the decision variables

In BiGA, the vector representation of the decision variables $(x, y) \in \Re^n$ are represented in each individual members of the population as points in the $(n_1+n_2)$ dimensional space. This representation is illustrated below.

| $y_1$ | $y_2$ | $y_3$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|---|---|

**Figure 3: Parameter coding of Individual**

Figure 3 shows the representation of the individual members of the population. The decision variables $(x_i, y_i)$ where $i$= 1….n, takes the value (0,1). The variables $y_1$, $y_2$ and $y_3$ referred to as the *link variables* are the follower variables represented in the leader problem.

## 6. Discussion and Results

The BiGA algorithm was compared with conventional approaches using 4 test problems obtained from the literature[11]. These problems represent a number of features that create difficulties for BLP optimisation algorithms. Analysis of the existing test problem is shown in Table 1.

For all test problems and with the BiGA algorithm a population size of 50, a crossover probability of 0.6 and a mutation probability of 0.08 were used. BiGA was run for 250 generation. The BiGA algorithm was run ten times with different random seeds. Seven out of ten runs obtained similar results. Table 3 shows the results obtained from the best convergence of the optimal solution for each test problem. The methods identified in the second column are methods used to derive results listed in the traditional results column (columns 3 and 4). These results were obtained from literature. The BiGA results are shown in columns 5 and 6. The first column of each of the results column presents the best optimal value for the leader problem ($F_L$) and the second column presents the best optimal value of 5 runs for the follower problem ($F_F$). Table 4 shows the variable values for the global solution indicating the optimal region of the BLP problem.

**Table 3: Optimal solution of BLP problem**

| Problem | Traditional method | Traditional Results | | BiGA Results | |
|---|---|---|---|---|---|
| | | $F_L$ | $F_F$ | $F_L$ | $F_F$ |
| F1 | | 1000 | 1 | 1000 | 1 |
| F2 | Penalty Function | 81.33 | - | 82.44 | 0.271 |
| F3 | Double Penalty function | 99.6 | 0.034 | 100.58 | 0.001 |
| F4 | Steepest Descent | -1.708 | 8.27 | 3.57 | 2.4 |

**Table 4: Optimal Region of BLP problem**

| Problem | Results | | BiGA Results | |
|---|---|---|---|---|
| | $x$ | $y$ | $x$ | $x$ |
| F2 | 10.02 | 0.82 | 10.04 | 0.1429 |
| F3 | 9.98 | 9.917 | 10.03 | 9.969 |

### 6.1. Performance analysis

It was observed from Table 3, that the BiGA algorithm gives comparable results to the traditional methods for problems F1, F2 and F3. By this it is implied that BiGA is able to find solutions in the optimal region of the search space similar to traditional methods. Table 4 shows the variable values for F2 and F3 problems indicating the optimal region of the BLP problem. However F4 obtained a sub-optimal. Further investigation with 5000 generations did not improve the solution, this suggest that the algorithm was getting stuck in a local optimum. This can be due to the complex nature of the search space and limitations in the constraints handling method adopted in this study. The test performed with the BiGA lead to the following conclusions.

- The proposed algorithm is able to find solutions close to the optimal region similar to traditional approaches and it can deal with non-differentiable complex the BLP problems.
- The BiGA is robust to address different classes of the BLP problem and can be used to solve problems independent of the nature of the search space. This is an important requirement for solving real-life problems where the nature search space is not always known.


0-7695-1733-1/02 $17.00 © 2002 **IEEE**

## 7. Future Work

The BiGA algorithm will be further developed and tested with more difficult test cases, and real world problems. Current limitations of BiGA and the corresponding future research work are listed below.

- Further work is required to investigate why the algorithm is getting stuck in the local optimum.
- The algorithm in its present form cannot handle the multi-modality problem. Existing multi-modal techniques in multi-objective optimisation research community can be adapted to solve this problem.
- More sophisticated constraints handling method will be investigated to improve the performance of the search.

## 8. Conclusion

This paper has proposed BiGA algorithm based on a genetic algorithm approach for solving bi-level optimisation problems. The proposed solution is illustrated using 4 test functions and the results shows that the BiGA approach is robust for solving different classes of the BLP problem with reasonable performance and therefore it demonstrates potential for solving real life problems.


## Acknowledgements

The authors wish to acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC) and Corus UK.


## 9. References


1. Hansen, P., B. Jaumard, and G. Savard, *New Branch-and-Bound Rules for Linear Bilevel Programming.* SIAM Journal of Scientific and Statistical Computing, 1992. **13**(5): p. 1194-1217.

2. Judice, J.J. and A.M. Faustino, *A Sequential LCP Method for Bilevel Linear Programming.* Annals of Operations Research, 1992. **34**(1-4): p. 89-106.

3. Bard, J.F. and J.T. Moore, *A Branch and Bound Algorithm for the Bilevel Programming Problem.* SIAM Journal of Scientific and Statistical Computing, 1990. **11**(2): p. 281-292.

4. Wei, Z., *Penalty function method for bi-level multiobjective programming.* Acta Automatica Sinica, 1998. **24**(3): p. 331-337.

5. Shimizu, K. and M. Lu, *A Global Optimization Method fro the Stakelberg Problem with Convex Functions via Problem Transformations and Concave Programming.* IEEE Transactions System, Man and Cybernatics, 1995. **SMC-25**(12): p. 1635-1640.

6. White, D.J. and G. Anandalingam, *A Penalty Function for Solving Bi-Level Linear Programs.* Journal of Global Optimization, 1993. **3**: p. 397-419.

7. Mathieu, R., L. Pittard, and G. Anandalingam, *Genetic Algorithm Based Approach to Bi-level Linear Programming.* Operations Research, 1994. **28**(1): p. 1-21.

8. Anandalingam, G., et al., *Artificial Intelligence Based Approaches for Hierarchical Optimization*, in *Impact of Recent Computer Advances in Operations Research*, R. Sharda, Editor, North-Holland: New York,1989.

9. Gendreau, M., P. Marcotte, and G. Savard, *A Hybrid Tabu-Ascent Algorithm for the Linear Bilevel Programming Problem.* Journal of Global Optimization, 1996. **8**(3): p. 217-233.

10. Yin, Y., *Genetic Algorithm based approach for bilevel programming models.* Journal of Transportation Engineering, 2000. **126**(2): p. 115-120.

11. Bard, J., *Practical Bilevel Optimization Algorithm and Applications*. Nonconvex Optmization and Its Applications. Vol. 30. Kluwer Academics, 1998.

12. Roy, R., *Adaptive Search and the Preliminary Design of Gas Turbine Blade Cooling System*., PhD Thesis, University of Plymouth: Plymouth, 1997.

13. Tiwari, A., *Evolutionary computing techniques for handling variable interaction in engineering design optimisation*, in *School of Industrial and Manufacturing Science*., PhD Thesis, Cranfield University: Cranfield,UK, 2001.

14. Wen, P.N. and S.T. Hsu, *Linear Bi-level programming Problems - A Review.* Journal of Operational Research Society, 1991. **42**(2): p. 125-133.

15. Goldberg D. E, *Genetic Algorithm in Search, Optimization and Machine Learning* ,Massachusetts: Addison Wesley,1989.

16. Savard, G. and J. Gauvin, *The Steepest Descent Direction for the Nonlinear Bilevel Programming Problem.* Operations Research letters, 1994. **15**: p. 265-272.

17. Tanino, T. and T. Ogawa, *An Algorithm for Solving Two-Level Convex Optimization Problems.* International Journal of System Sciences, 1984. **15**(2): p. 163-174.