UNIVERSITY OF LEEDS

This is a repository copy of *Solving transportation bi-level programs with differential evolution* .

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/3628/

Proceedings Paper:

Koh, Andrew (2007) Solving transportation bi-level programs with differential evolution. In: IEEE Congress on Evolutionary Computing. Proceedings of the 2007 IEEE Congress on Evolutionary Computation, 25-28 Sep 2007, Singapore. IEEE , pp. 2243-2250. ISBN 978-1-4244-1340-9

https://doi.org/10.1109/CEC.2007.4424750

Reuse
See Attached

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Solving Transportation Bi-Level Programs with Differential Evolution

Andrew Koh, Member, IEEE

Abstract— **Bi-level programming problems arise in situations when the decision maker has to take into account the responses of the users to his decisions. These problems are recognized as one of the most difficult and challenging problems in transportation systems management. Several problems within the transportation literature can be cast in the bi-level programming framework. At the same time, significant advances have been made in the deployment of stochastic heuristics for function optimization. This paper reports on the use of Differential Evolution (DE) for solving bi-level programming problems with applications in the field of transportation planning. After illustrating our solution algorithm with some mathematical functions, we then apply this method to two control problems facing the transportation network manager. DE is integrated with conventional traffic assignment techniques to solve the resulting bi-level program. Numerical computations of this DE based algorithm (known as DEBLP) are presented and compared with existing results. Our numerical results augment the view that DE is a suitable contender for solving these types of problems.**

I. INTRODUCTION

WE present an application of the Differential Evolution heuristic to a class of transportation decision making problems. Stochastic optimization techniques are recognized as useful tools for solving problems where objective functions do not necessarily satisfy classical optimization assumptions such as continuity, convexity and differentiability. Techniques include simulated annealing (SA) [1] and genetic algorithms (GA) [2], Ant Colony Optimization (ACO) [3], Particle Swam Optimization (PSO) [4] and Differential Evolution (DE) [5]. These have been applied in various ways in solving difficult problems within transportation (and elsewhere) with a high degree of success.

In the highway transportation context, the network design problem was tackled using SA [6], ACO [7] and PSO [8]. GAs have been used in [9] for toll and reserve capacity optimization while [10, 11, 12] report on the use of GAs for designing toll pricing cordons.

In this paper, two transportation problems with particular relevance to policy makers are formulated in a bi-level programming framework and the Differential Evolution Heuristic is applied to solve them. DE is a simple algorithm that utilizes perturbation and recombination to optimize a multi-modal function and has already been reported to perform effectively when applied to practical engineering problems.

This paper has a further 6 sections. Section 2 outlines the generic bi-level problem. Section 3 reviews DE and shows how it can be used to solve bi-level programming problems. Section 4 applies this method to several mathematical test problems. Section 5 and 6 applies the DE Heuristic to 2 problems of relevance to transportation regulators. Finally Section 7 provides some conclusions and directions for further research.

II. BI-LEVEL PROGRAMMING FRAMEWORK

Bi-level programming has applications in robot motion planning, chemical engineering, production planning, as well as in the field of transportation [13-14]. In game theory, a bi-level programming problem is known as a Stackleberg or leader-follower game [15] in which the leader chooses his variables so as to optimize his objective function, taking into account the response of the follower(s) who separately optimize their own objectives, treating the leader's decisions as exogenous. Denoting x as the leader's decision variables, y as the vector of the follower's decision variables, the generic bi-level program can be written as

$\min_x U(x, y)$ where y is obtained by solving the lower level optimization problem (Program L)

$$\min_y L(x, y)$$

In this framework, the evaluation of the upper-level objective function requires solving the lower-level problem. In other words, the leader cannot optimize her objective without regards to the reactions of the followers. Even when both the upper level and the lower level consist of convex programming problems, the resulting bi-level problem itself may be non-convex [16]. The lower-level problem is an implicit non-linear constraint on the upper level problem [17]. Non convexity suggests the possibility of multiple local optima. This framework has been applied within transportation to the Network Design Problems [6], Optimization of signal timings [18], Toll Pricing Problems [9-12],[19] among others. In this paper, we illustrate, through numerical examples, that the Differential Evolution heuristic can be applied in this framework to generate good solutions to support informed transportation systems

Manuscript received June 15, 2007. This work was supported in part by the Engineering and Physical Sciences Research Council of the UK.

Andrew Koh is a Research Officer with the Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, United Kingdom (corresponding author phone: 00-44-(0)-113-343-5325; fax: 00-44-(0)-113-343-5334; e-mail: a.koh@its.leeds.ac.uk).

1-4244-1340-0/07$25.00 ©2007 IEEE

management. We develop and demonstrate a meta-heuristic DE algorithm to solve this class of problems, first with mathematical functions and then specifically with transportation planning problems.

III. Differential Evolution Based Algorithm

In general, the test problems in the DE literature have focused on the optimization of multimodal single-level programs e.g. Rastrigin's or Ackley's function etc. The novelty of our DE application is that we eliminate the bi-level aspect of the problem by treating the program as a single level problem *BUT* simultaneously taking into account the follower's program (Program L) in the process of optimizing the leader's objective (Program U). Our resulting meta-heuristic is called: DEBLP, Differential Evolution for Bi-Level Programming. In essence, it combines Differential Evolution manipulation of the Leader's variables with classical gradient based algorithms for optimization for the lower level problem.

Differential Evolution (DE) was devised by Storn and Price [5] as a direct search algorithm. DE has been applied to a variety of real-world engineering problems [20-22]. As far as we are aware, DE has not yet been tested on bi-level problems.

The pseudo code for DEBLP is given in Table I. Our solution method uses DE to generate and modify x , the leader's decision variables. At each iteration ("generation" in DE parlance), DEBLP uses classical gradient based optimization algorithms to solve Program L to obtain y . This enables evaluation of the upper level objective $U(x, y)$ and determination of the "fitness" of a particular x vector. DE operations (described herein) are then performed to generate a new trial population of the leader's variables and the process is repeated for a number of user specified iterations ($MaxG$).

TABLE I
PSEUDOCODE FOR DEBLP

Input MaxG, Q,CR,NP

```
Begin G = 1
Initialization of Leader's Variables
Evaluation
-solve Follower's Program for each vector of the Leader's
Variables using gradient based non-linear optimization
algorithm
-obtain leader's objective and fitness of Leader's vector
Do until G= MaxG
     Perform Mutation
     Perform Crossover
     Perform Evaluation
-solve Follower's Program for each vector of the Leader's
Variables using gradient based non-linear optimization
algorithm
-obtain leader's objective and fitness of Leader's trial
vector
     Perform Selection
     G = G+1
Repeat
Output: U(x*, L(x*, y*))
```

In what follows, we discuss the steps of DEBLP with reference to Table I and assume minimization of the leader's problem. Let the problem dimension be denoted Dim .

A. Initialization

An initial population of size (NP) of the leader's decision variables (x), known as the parent population in DE parlance, is randomly generated using (1) as follows:

$$x_{i,j,G} = rnd(UB_j - LB_j) + LB_j \qquad (1)$$

$$\forall i \in \{1,2,...,NP\}, \forall j \quad \{1,2,...,Dim\}, rnd \quad [0,1]$$

B. Evaluation

For each of the leader's decision variables (x), we can obtain the corresponding follower's variables (y) by using traditional gradient based optimization methods. Hence we obtain the value of Program U for each member of this population; the member that results in the lowest objective function value for $U(x, L(x, y))$ is denoted the "best member" of the population ($x^{Best}_{j,G}$) at generation G .

C. Mutation

The mutation process combines different elements of the parent population heuristically to generate a mutant vector ($m_{i,a,G}$) in accordance with (2):

$$m_{i,j,G} = x^{Best}_{j,G} + Q(x_{r1,j,G} \quad x_{r2,j,G}) \qquad (2)$$

$$\forall i \in \{1,2,...,NP\}, \forall j \quad \{1,2,...,Dim\}$$

$r1, r2 \in \{1,2,...,NP\}$ are random integer and mutually different indices and also different from the current running index i . $Q \in [0,2]$ is a mutation factor that scales the impact of the differential variation. The mutation strategy shown in (2) is one of several variants proposed in [5].

D. Crossover

On this mutant vector ($m_{i,j,G}$) crossover is probabilistically performed to produce a child vector ($w_{i,j,G}$) according to (3) as follows:

$$w_{i,j,G} = \begin{cases} m_{i,j,G} \text{ if } rnd \in [0,1] < CR \vee \text{ j} \quad h \\ x_{i,j,G} \quad \text{otherwise} \end{cases} \qquad (3)$$

$$\forall i \in \{1,2,...,NP\}, \forall j \quad \{1,2,...,Dim\}$$

$h \in \{1,2,...,Dim\}$: a random integer parameter index chosen to ensure that the child vector will differ from its parent by at least one parameter. CR is the probability of crossover.

Crossover can produce child vectors that lie out of the bounds of the original problem specification. There are several ways to ensure satisfaction of these constraints [23-24]. One could set the parameter equal to the limit it exceeded. Alternatively, as outlined in [23], out of bound values can be reset to a point half way between its pre-mutation value and the bound violated using (4).

$$w_{i,j,G}=\begin{cases}\dfrac{x_{i,j,G}+LB_j}{2} & \text{if } u_{i,j,G}<LB_j\\ \dfrac{x_{i,j,G}+UB_j}{2} & \text{if } u_{i,j,G}>UB_j\\ w_{i,j,G} & \text{otherwise}\end{cases} \quad (4)$$

$\forall i \in \{1,2,...,NP\}, \quad j \quad \{1,2,...,Dim\}$

E. Selection

The fitness of each child vector is compared against that of the parent. This means that comparison is against the same i^{th} $(\forall i \in \{1,2,...,NP\})$ vector parent on the basis of whichever of the two gives a lower value (assuming minimization) for Program U. The one that produces a lower value survives to become a parent in the next generation as shown in (5).

$$x_{i,G+1}=\begin{cases} w_{i,G} & \text{if } U(w_{i,G},L(w_{i,G},y))<U(x_{i,G},L(x_{i,G},y))\\ x_{i,G} & \text{otherwise}\end{cases} \quad (5)$$

$\forall i \in \{1,2,...,NP\}$

These steps are repeated for a maximum number of user defined iterations ($MaxG$). Our description of DE is based on the variant known as "DE/best/1/bin" scheme. Though at least many as 10 different variants have been proposed [5, 23], we utilized this variant in the work reported here.

In the following section, we test DEBLP on several mathematical bi-level functions from the literature. We subsequently apply DEBLP to two specific bi-level problems from the transportation literature.

IV. Bi-Level Function Optimization

In this section, we apply DEBLP to test problems taken from the literature [25-30]. The 7 test problems used are shown in Table II. Table III compares the optima reported with the solutions found by DEBLP.

TABLE II
TEST PROBLEMS FOR BI-LEVEL PROGRAMMING

#	Problem
P1	$\min_x U(x,L(x,y)) = (x-1)^2 \quad (y \quad 1)^2$ $\min_y L(x,y) = 0.5y^2 + 500y \quad 50xy$
P2	$\min_x U(x,L(x,y)) = x_1^2 - 2x_1 \quad x_2^2 \quad 2x_2 \quad y_1^2 \quad y_2^2$ st. $0 \le x_1, x_2 \le 2$ $\min_y L(x,y) = (y_1 - x_1)^2 \quad (y_2 \quad x_2)^2$ st. $0.25 - (y_i - 1)^2 \quad 0, i \quad 1,2$

TABLE II CONT'D
TEST PROBLEMS FOR BI-LEVEL PROGRAMMING

#	Problem
P3	$\min_x U(x,L(x,y)) = x^2 + (y \quad 10)^2$ st. $x + y \quad 0, 0 \quad x \quad 15$ $\min_y L(x,y) = (x + 2y \quad 30)^2$ st. $x + y \le 20, 0 \quad y \quad 20$
P4	$\min_x U(x,L(x,y)) = (x-5)^2 + (2y \quad 1)^2$ st. $x \ge 0$ $\min_y L(x,y) = (y-1)^2 \quad 1.5xy$ st. $-3x + y \quad 3,$ $x - 0.5y \le 4,$ $x + y \le 7, y \quad 0$
P5	$\min_x U(x,L(x,y)) = (x-3)^2 \quad (y \quad 2)^2$ st. $2x - y \ge \quad 1,$ $x + 2y \ge 2,$ $x - 2y \quad 14, 0 \quad x \quad 8$ $\min_y L(x,y) = (y-5)^2$
P6	$\min_x U(x,L(x,y)) = (x-3)^2 \quad (y \quad 2)^2$ st. $0 \le x \le 8$ $\min_y L(x,y) = (y-5)^2$ st. $2x - y \ge \quad 1,$ $x + 2y \ge 2,$ $x - 2y \quad 14$
P7	$\min_x U(x,L(x,y)) = 2x_1 + 2x_1 - 3y_1 \quad 3y_2 \quad 60$ st. $x_1 + x_2 + y_1 \quad 2y_2 \quad 40 \quad 0$ $0 \le x_1, x_2 \le 50$ $\min_y L(x,y) = (y_1 - x_1 \quad 20)^2 \quad (y_2 \quad x_2 \quad 20)^2$ st. $2y_1 \quad x_1 + 10 \quad 0,\ 2y_2 \quad x_2 \quad 10 \quad 0,$ $10 \le y_1, y_2 \quad 20$

TABLE III
REPORTED GLOBAL OPTIMA SOLUTIONS VS SOLUTIONS FOUND BY DEBLP FOR TEST PROBLEMS

#	Reported Optimal Solution/ [Source]	Solution with DEBLP	Percentage Optima Obtained in 30 runs
P1	U* =81.33 ; L* = 0.34; /[25]	As per literature	100%
P2	U* = -1 ; L* = 0; /[26-27]	As per literature	100%
P3	U* = 100; L* =0; /[13,28]	As per literature	100%
P4	U* =17; L* =1; /[28-29]	As per literature	100%
P5	U* = 9; L * = 0; /[29-30]	As per literature	100%

TABLE III CONT'D
REPORTED GLOBAL OPTIMA SOLUTIONS VS SOLUTIONS FOUND BY DEBLP FOR TEST PROBLEMS

#	Reported Optimal Solution/ [Source]	Solution with DEBLP	Percentage Optima Obtained in 30 runs
P6	U* = 5; L* = 4; /[29–30]	U* = 2; L* = 4 (x = 4,y =3)	100%
P7	U* = 0; L* = 200; /[28]	U* = 0; L* = 100 (x1 = 0, x2 =30; y1 = 10, y2 =-10).	100%

With $NP = 20$, $Q = 0.8$, $CR = 0.8$ and $MaxG = 40$, DEBLP found the optimal solution to each of these problems which were each run 30 times. The lower level program was solved using the *fmincon* routine (a gradient based sequential quadratic programming algorithm) from MATLAB's optimization toolbox, integrating that routine into DEBLP at the evaluation phase. Table III also shows that in all 30 runs, DEBLP obtained/bettered the best reported optimal solution. For all 7 problems, DEBLP consistently obtained to the global optimum reported/discovered by the 30th generation.

Keeping in mind that all 7 test problems are minimization problems, Table III shows that we found better solutions for two problems: P6 and P7. For P6, the best reported optima was U* = 5; L* = 4 but DEBLP found a lower objective U* = 2 with the same L* = 4.

For P7, the best reported optima was U* = 0; L* = 200 but DEBLP found U* = 0 as in [28] but L* = 100 which is lower than the best solution found for the lower level program.

P3, P5 and P7 had linear inequalities as constraints in the upper level problem. These were handled by degrading the fitness of the trial values, in the evaluation phase, if the trial values produced did not satisfy the constraints, by using a fixed penalty method. Specifically, we added a random number between 5,000 and 10,000 to degrade the fitness produced by trial vectors that did not satisfy these constraints.

Note that it is also possible to apply DEBLP to linear bi-level programs (i.e. bi-level problems that are linear in both the upper and lower hierarchy). In that case, the well known simplex algorithm for linear programming [13] can be used for solving the lower level program.

One obvious limitation of these examples is that the leader's vector consists of at most 2 dimensions. This is because the known global optima for such problems even with such small dimensions are difficult to obtain with traditional algorithms and the task will be made more difficult with increased dimensions. However, in further examples that focus on transportation management problems, we will demonstrate that DEBLP can be applied to problems with larger dimensions with encouraging success.

In the next two sections we consider two practical problems encountered in the transportation literature that can be cast in the bi-level programming framework. We then apply DEBLP to these problems and compare with the results reported in the literature.

V. CONTINUOUS OPTIMAL TOLL PROBLEM (COTP)

The continuous optimal toll problem involves selecting an optimal toll level for each predefined tolled link in the network [11]. Many transportation authorities around the world are interested in setting tolls to control congestion (e.g. Singapore, London). In view of this, the COTP provides a practical application of DEBLP.

A. Model Formulation

Consider a transportation network with N nodes and A links, let:

R : the set of all routes in the network

H : the set of all Origin Destination (OD) pairs in the network

R_h : the set of routes between OD pair $h\,(h \in H)$

D_h : the demand between each OD pair $h\,(h \in H)$

f_r : the flow on route $r\,(r \in R)$

v : the vector of link flows, $v = [v_a]$ $(a \in A)$

$t_a(v_a)$: the travel time on the link a, as a function of link flow v_a on that link only.

δ_{ar} : 1 if the route $r\,(r \in R)$ uses link $a\,(a \in A)$, 0 otherwise

T : the set of links that are tolled ($T \subseteq A$)

τ : the vector of tolls $\tau = [\tau_a]$, $(a \in T)$

$\tau_a^{max}, \tau_a^{min}$: the upper and lower bounds of toll charge $(a \in T)$

System cost, conventionally measured as the sum product of the travel times and traffic flows on all links in the network, may be interpreted as the social cost of the transport sector and proxies the resource cost to the economy of the highway system. The objective of the decision maker in the COTP is to minimize this given by (6) by charging tolls:

Program U :

$$\min_{\tau} U(v) = \sum_{\forall a \in A} v_a(\tau) t_a(v_a(\tau)) \tag{6}$$

Subject to:

$$\tau_a^{min} \le \tau_a \le \tau_a^{max} \quad \forall a \in T \tag{7}$$

$$\tau_a = 0 \quad \forall a \notin T$$

where v is obtained by solving the lower level program (Program L)

$$\min_{v} L = \sum_{\forall a} \int_0^{v_a} (t_a(z), \tau_a) dz \tag{8}$$

Subject to:

$$\sum_{r \in R_h} f_r = D_h \ , h \in H \tag{9}$$

$$v_a = \sum_{r \in R} f_r \delta_{ar} \ , \forall a \quad A \tag{10}$$

$$\in f_r \geq 0 \ , \forall r \quad R \ . \tag{11}$$

B. *The Lower Level Program in Transportation*

In the transportation literature, Program L has a special interpretation in that it is the mathematical formulation for representing the follower's (road user's) route choice [31], often referred to as the Traffic Assignment Problem (TAP). The basic behavioral premise employed in Traffic Assignment is that the route choice is governed by Wardrop's user equilibrium principle [32] where user equilibrium is attained when no user can decrease his travel costs (with or without tolls) by unilaterally changing routes. The solution to the TAP is the equilibrium link flow vector (v). It is known that a traffic assignment algorithm (e.g. those in [33]) can be used to be used to solve Program L .

In the COTP as formulated, the leaders variables "x" is analogous to the toll vector τ . Within DEBLP, solving the lower level program is equivalent therefore to solving a TAP for each toll vector generated by DE.

C. *Previous Work on the COTP*

Various solution algorithms have been proposed for the COTP. Yang and Lam proposed a linearization based method that uses derivative information to form approximations to the upper level objective [19]. Another derivative-based method was derived from constraint accumulation [34]. Preliminary tests by the author and colleagues on this latter method have shown that it is very sensitive to the starting point [35]. A comprehensive review of these and other algorithms for the COTP are found in [11].

D. *Example*

We illustrate the use of DEBLP to solve the COTP with a small example from [19]. Fig. 1 shows the network which has 6 nodes and 7 links. Link numbers are written above the links and node numbers are indicated accordingly. There are two OD pairs between nodes 1 and 3 and between 2 and 4 of 30 trips each. The rest of the nodes represent junction/intersections of the road network and travel is in the direction indicated by the arrows.

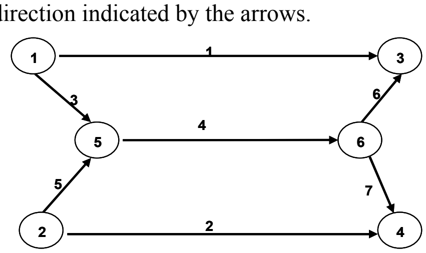

Fig. 1. Network for COTP Example

The link travel times take the explicit function forms as given by:

$$t_a(v_a) = t_a^0 (1 + 0.15 (\frac{v_a}{C_a})^4) \tag{12}$$

where t_a^0 is as the free flow travel time on the link and C_a is the link capacity. The parameter details for the network and the upper bound on tolls are given in Table IV. Note that $\tau_a^{\min} = 0, \forall a \in T$.

TABLE IV
NETWORK PARAMETERS FOR COTP EXAMPLE

Link	t_a^0	C_a	$\tau_a^{\max}$
1	8	20	5
2	9	20	5
3	2	20	2
4	6	40	2
5	3	20	2
6	3	25	2
7	4	25	2

For this example, NP = 20, = 7, Q = 0.8, CR = 0.9 and $MaxG$ = 50. Table V compares the results of DEBLP with that of two deterministic algorithms (direct from [19] and our implementation of the algorithm of [34]) together with a Genetic Algorithm based method from [9]. UPO refers to the upper level objective in (6). It can be seen from Table V that the four different algorithms provided different tolls. However the upper level objective function values are the same in all cases.

TABLE V
COMPARISON OF EXISTING AND DE BASED RESULTS FOR COTP EXAMPLE

Tolls	[19]	[34]	[9]	DEBLP
Link 1	3.820	2.667	4.324	3.824
Link 2	4.265	3.548	4.976	3.920
Link 3	0.472	0.038	0.035	0.564
Link 4	0.476	0.154	1.759	0.462
Link 5	0.294	0.116	0.016	0.145
Link 6	0.472	0.038	0.127	0.396
Link 7	0.294	0.116	0.013	0.111
UPO	628.60	628.60	628.60	628.60

VI. CONTINUOUS NETWORK DESIGN PROBLEM (CNDP)

The continuous network design problem (CNDP) involves determination of capacity enhancements of existing facilities of a network in such a way that the decision is regarded as optimal [6]. Care has to be taken when solving the CNDP because additional capacity can unproductively increase the total network travel time and this is a phenomenon known as Braess's paradox. Braess's paradox has been known to occur in transportation [36] and telecommunication networks [37].

A. Model Formulation

To proceed with this example, we introduce additional notation as follows (others as previously defined):

K : the set of links that have their individual capacities enhanced ($K \subseteq A$).

β : the vector of capacity enhancements $\beta = [\beta_a],\ (a \in K)$.

$\beta_a^{\max}, \beta_a^{\min}$: the upper and lower bounds of capacity enhancements $(a \in K)$.

d_a : the monetary cost of capacity increments per unit of enhancement $(a \in K)$.

C_a^0 : existing capacity of link a $(\forall a \in A)$

θ : conversion factor from monetary investment costs to travel times.

The CNDP seeks a K dimension vector of capacity enhancements optimal to the following bi-level program:

The Upper level problem (Program U) is described as:

$$\min_{\beta} U(v,\beta) = \sum_{\forall a \in A} v_a(\beta) t_a(v_a(\beta), \beta_a) + \sum_{\forall a \in K} \theta d_a \beta_a \quad (13)$$

subject to:

$$\beta_a^{\min} \le \beta_a \le \beta_a^{\max} \quad \forall a \in K; \quad (14)$$
$$\beta_a = 0 \quad \forall a \notin K$$

where v is obtained by solving a lower level problem (Program L) similar to that previously defined in the equation set (8-11).

In the CNDP, the regulator aims to minimize the sum of the system cost and the investment cost given by (13) with constraints on the amount of capacity additions (14) while Program L determines the user's route choice, for a given β , once again based on Wardrop's principle of route choice as mentioned previously.

The leaders variables " x " is analogous to the vector of capacity enhancements, β . Within DEBLP, solving the lower level program is equivalent therefore to solving a TAP for each capacity enhancement vector generated by DE.

B. Previous Work on CNDP

The CNDP has been investigated by many researchers and various solution algorithms have so far been proposed. Meng *et al* transformed the bi-level program into a single level continuously differentiable problem using the marginal function method and solved the resulting problem with Augmented Lagragian method [38]. Chiou investigated several variants of the descent based Karush-Khun-Tucker approaches [39]. Stochastic optimization techniques have also been used; GAs were applied in [40] and the use of SA has been reported in [6].

C. Example 1: Hypothetical Network

The network for the first example is taken from [40] and reproduced in Fig. 2. This network has 6 nodes and 2 OD pairs; the first between 1 and 6 of 10 trips and the second, between nodes 6 and 1 of 20 trips.

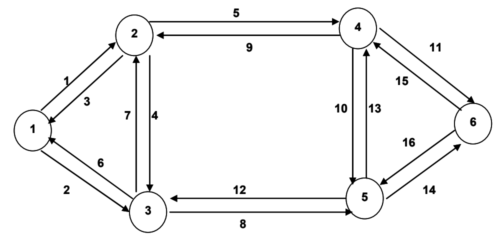

Fig. 2. Network for CNDP- Example 1

The travel times adopt the following form as given by

$$t_a(v_a) = t_a^0 (1 + 0.15(\frac{v_a}{C_a^0 + \beta_a})^4) . \quad (15)$$

TABLE VI
NETWORK PARAMETERS FOR CNDP EXAMPLE 1

Link	t_a^0	C_a^0	d_a
1	1	3	2
2	2	10	3
3	3	9	5
4	4	4	4
5	5	3	9
6	2	2	1
7	1	1	4
8	1	10	3
9	2	45	2
10	3	3	5
11	9	2	6
12	4	6	8
13	4	44	5
14	2	20	3
15	5	1	6
16	6	4.5	1

Table VI gives details regarding the free flow travel times, initial link capacities and the cost of capacity expansion. Note that $\beta_a^{\min} = 0$ and $\beta_a^{\max} =$ 20, $\forall a \in K, K \subseteq A$ as in [40].

For this example, we assumed NP = 20, Q = 0.9, CR = 0.99 and $MaxG$ =150. Table VII summarizes the results that have been reported previously and compares it with the results reported in our paper. UPO is the upper level objective in (13), NFE is the number of function evaluations (number of lower level programs solved equal to ($NP * MaxG$)) and SD is the standard deviation over 30 runs. Our results from DEBLP are based on the mean of these 30 runs.

Though the standard deviation of the GA method used in [40] is much lower, the authors also reported using a local search method to aid the search process.

TABLE VII
RESULTS FOR CNDP EXAMPLE 1

Method	DETERMINISTIC			STOCHASTIC	
Source	[39]	[38]	[6]	[40]	DEBLP
UPO	534.0	532.71	528.49	519.03	522.71
NFE	29	4,000	24,300	10,000	3,000
SD	-------Not Reported------------			0.403	1.34

D. *Example 2: Sioux Falls Test Network*

The second example is the CNDP for the Sioux Falls network with 24 nodes, 76 links and 552 OD pairs. This is the network of a real city of Sioux Falls, South Dakota in the USA. Due to space constraints, please refer to [38] for the network parameters and OD details. Only 10 links out of the 76 are subject to improvements.

While this network is clearly larger and arguably more realistic, the problem dimension (number of variables simultaneously optimized) is smaller than in the previous network used, since 10 links are subject to improvement compared to 16 links previously. This could explain why the number of function evaluations reported in all studies compared is less than for the first example. Furthermore, our literature review does not indicate that GA has been used for this particular problem. The results are compared in Table VIII. Our results from DEBLP are again based on the mean of 30 runs.

TABLE VIII
RESULTS FOR CNDP EXAMPLE 2

Method	DETERMINISTIC			STOCHASTIC
Source	[39]	[38]	[6]	DEBLP
UPO	82.57	81.75	80.87	80.74
NFE	10	2,000	3,900	1,600
SD	-----------Not Reported---------------------			0.002

From Table VIII, DEBLP is able to locate the global optimum; again with a lesser number of iterations than the SA method in [6] or the deterministic method in [38]. The standard deviation is also very low which suggests that this heuristic is reasonably robust as well.

VII. CONCLUSIONS

The purpose of this paper was to assess the ability of a simple meta-heuristic, DEBLP which combined both DE manipulation with traditional optimization algorithms to solve bi-level programming problems. In particular, bi-level programming problems are important for transportation analysts intending to use policy variables (such as capacity and tolls) to optimize their network.

Through simple numerical bi-level function optimization examples as well as specific applications in transportation, we have demonstrated that DE can be integrated within the bi-level programming framework to provide good solutions. In the numerical function optimization examples, DEBLP not only found all the global optima that have been reported for 5 test functions but also provided better solutions to two other examples than previously reported in the literature.

Our specific examples from transportation network analysis subsequently demonstrated that DEBLP can outperform some deterministic (locally optimal) convergent algorithms with an approximately equivalent number of function evaluations and can perform as well, if not outperform established stochastic optimization techniques.

One concern with DE, as with most evolutionary algorithms, is that it potentially suffers from the "curse of dimensionality" [41]. This is illustrated by the fact that the number of lower level program evaluations required for CNDP Example 1 and its standard deviation are both higher than for CNDP Example 2 partially due to the larger dimensions of the former problem. More investigation into this is clearly required.

Our simple examples also ignored complementary or competing objectives and trade off decisions. In our formulation of the bi-level program, we assumed that the objective of decision maker was only singly defined e.g. minimize system cost *only*. However, she might be envisaged to have other objectives e.g. maximizing revenue in the case of toll pricing while simultaneously minimizing system cost. A possible extension of this work would be to extend the bi-level formulation within a multi-objective optimization framework such as those proposed in [42].

It is interesting that DE's simplistic perturbation and recombination scheme can result in a reasonably efficient optimization heuristic. The performance of the DE heuristic and its comparison to other techniques, particularly pertaining to the optimization of bi-level programs, would constitute a challenging and fruitful area for further research.

ACKNOWLEDGMENT

The author thanks Professor Hillel Bar Gera (Ben Gurion University of the Negev, Israel) for providing the use of his origin based traffic assignment software within DEBLP and Professor Mike Maher for comments on an earlier draft of this paper. The author expresses his appreciation to the anonymous referees who suggested illustrating the DEBLP meta-heuristic with mathematical functions from the literature. All errors remain the author's sole responsibility.

REFERENCES

[1] E Aarts. and J. Korst, *Simulated Annealing and Boltzman Machines,* Chichester: John Wiley,1988.
[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning,* Reading, Massachusetts: Addison-Wesley, 1989.
[3] M. Dorigo and T. Stützle, *Ant Colony Optimization,* Cambridge, Massachusetts: MIT Press,2004.

[4] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", *Proc IEEE Conf Neural Networks*, Piscataway, NJ: IEEE Press, 1995, pp. 1942-1948.
[5] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.
[6] T. Friesz, H. Cho, N. Mehta, R. Tobin and G. Anandalingam, "A Simulated Annealing Approach to the Network Design Problem with Variational Inequality Constraints", *Transportation Science,* vol. 26, no. 1, pp. 18-26, 1992.
[7] H. Poorzahedy and F. Abulghasemi, (2005), "Application of Ant System to network design problem", *Transportation* vol. 32, no.3, pp. 251-273, 2005.
[8] H. Xiao and C. Huang, "A new model for equilibrium network design problem and algorithm" *Journal of Wuhan University (Natural Science Edition),* vol. 52, no**.** 3, pp. 301-304, 2006, (in Mandarin).
[9] Y. Yin, "Genetic Algorithms Based Approach for BiLevel Programming Models", *ASCE Journal of Transportation Engineering,* vol. 126, no. 2, pp. 115-120,2000.
[10] A. Sumalee, "Optimal Road User Charging Cordon Design: a Heuristic Optimization Approach", *Computer Aided Civil and Infrastructure Engineering,* vol. 19, no**.5**, pp. 377-392, 2004.
[11] A. Sumalee, "Optimal Road Pricing Scheme Design", PhD Thesis, Institute for Transport Studies, University of Leeds, UK, 2004.
[12] X. N. Zhang and H. Yang, "The optimal cordon-based network congestion pricing problem", *Transportation Research Part B,* vol. 38, no**.** 5, pp. 517–537, 2004.
[13] J.F. Bard, *Practical Bilevel Optimisation: Algorithms and Applications*. Dordrecht: Kluwer, 1988.
[14] Z.Q. Luo, J.S. Pang and D. Ralph, *Mathematical Programs with Equilibrium Constraints*. Cambridge: Cambridge University Press, 1996.
[15] C. Fisk, "Game theory and transportation systems modelling", *Transportation Research Part B,* vol. 18, no. 4/5, pp. 301-313,1984.
[16] M. G. H. Bell and Y. Iida, *Transportation Network Analysis,* Chichester: John Wiley, 1997.
[17] X. Y. Zhang and M. Maher, "An algorithm for the solution of bi-level programming problems in transport network analysis" in *Mathematics in Transport Planning and Control* , J. D. Griffiths, (ed), Amsterdam: Pergamon, 1998, pp. 177-186.
[18] S. Chiou, (1998), "Bi-level Formulation for Equilibrium Traffic Flow and Signal Settings" in *Mathematics in Transport Planning and Control*, J. D. Griffiths, (ed), Amsterdam: Pergamon, 1998, pp. 59-68.
[19] H. Yang and H. K. Lam, ''Optimal road tolls under conditions of queuing and congestion'', *Transportation Research Part A,* vol. 30, no. 5 pp. 319–332, 1996.
[20] N. Karaboga, "Digitial IIR Filter Design Using Differential Evolution Algorithm" *Eurasip Journal of Applied Signal Processing*, vol. 2005,no.8, pp. 1269-1276, 2005.
[21] J. Ilonen, J. Kamarainen, and J. Lampinen, "Differential Evolution Training Algorithm for Feed-Forward Neural Networks", *Neural Processing Letters* vol. 7, no.1, pp. 93-105, 2003.
[22] A. Qing, "Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems" *IEEE Trans. Geosci. Remote Sens*., vol. 44, no.1, 116-125, 2006.
[23] K. Price, "An Introduction to Differential Evolution" in *New Techniques in Optimization* D. Corne, M. Dorigo and F. Glover (eds),, London: McGraw Hill, 1999, pp. 79-108.
[24] K. Zeilinski and R. Laur, "Constrained Single Objective Optimization Using Differential Evolution" *Proc. IEEE Conf. Evolutionary Computation*, Piscataway , NJ: IEEE Press, pp. 927-934, 2006.
[25] V. Oduguwa and R. Roy, "Bi-level Optimisation using Genetic Algorithms" in *Proc. IEEE Conf. Artificial Intelligence Systems,* Piscataway, NJ: IEEE Press, pp. 322-327, 2002.
[26] F. Facchinei, H. Jiang and L. Qi, "A smoothing method for mathematical programs with equilibrium constraints" *Mathematical Programming Part B*, vol. 85, no. 1, pp. 107-134, 1999.
[27] A. H. DeSilva "Sensitivity formulae for nonlinear factorable programming and their application to the solution of an implicitly defined optimization model of US crude oil production" PhD Thesis, George Washington University, Washington,1978.
[28] V. Visweswaran, C. Floudas, M. Ierapetritou and E. Pistikopolous "A decomposition based global optimization approach for solving bilevel linear and non-linear quadratic programs" in *State of the Art in Global Optimization: Computational Methods and Applications.* P. Pardalos and C. Floudas ,(eds), Kluwer: Dordrecht, pp.139-162,1996.
[29] Rajesh J., K. Gupta, H. S. Kusumar, V. Jayaraman and B. Kulkarni "A tabu search based approach for solving a class of bilevel programming problems in chemical engineering", *Journal of Heuristics*, vol. 9, no.4, pp. 307-319, 2003.
[30] P. A. Clark and A. Westerberg, "Bilevel programming for steady stead chemical process design- 1 Fundamentals and Algorithms." *Computers and Chemical Engineering,* vol. 14, no.1, pp. 87-97,1990.
[31] M. Beckmann, C.B. McGuire, and C.B. Winsten, *Studies in the Economics of Transportation*, New Haven, Connecticut: Yale University Press, 1956.
[32] J G. Wardrop "Some theoretical aspects of road traffic research", *Proc. Institute of Civil Engineers Part II*, vol. 1, no. 2, 325-378, 1952.
[33] Y. Sheffi, *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods* Englewood Cliffs, New Jersey: Prentice Hall, 1985.
[34] S. Lawphongpanich. and D. W. Hearn, An MPEC Approach to Second-Best Toll Pricing, *Mathematical Programming Part B*, vol. 101, no. 1, pp. 33-55, 2004.
[35] A. Koh, S. Shepherd and A. Sumalee, "Second Best Toll and Capacity Optimisation in Networks" *Proc 39th University Transport Studies Group Annual Conference*. Harrogate, England, January 3-5,2007.
[36] D. Braess, "Uber ein paradoxon aus der verkehrsplanung," *Unternehmenforschun*g, vol. 12**,** pp. 258-268, 1968.
[37] Y. Korilis, A. Lazar and A. Orda "Avoiding the Braess paradox in non-cooperative networks", *Journal of Applied Probability*, vol. 36, no.1, 1999.
[38] Q. Meng, H. Yang and M.G.H. Bell, (2000), "An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem" *Transportation Research Part B*, vol. 35, no. 1, pp. 83-105, 2000.
[39] S. Chiou, "Bilevel programming formulation for the continuous network design problem", *Transportation Research Part B*, vol. 39, no. 4, pp.361-383, 2005.
[40] N.D. Cree, M.J. Maher and B. Paechter, "The continuous equilibrium optimal network design problem: a genetic approach", in *Transportation Networks: Recent Methodological Advances* Bell, M. G. H., (ed), 1996, London: Pergamon, pp. 163-174.
[41] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton, New Jersey: Princeton University Press, 1961.
[42] V. L. Huang, P. N. Suganthan and S. Baskar, "Multiobjective Differential Evolution with External Archive", *Technical Report*, Nanyang Technological University, Singapore, 2005.