

promoting access to White Rose research papers



Universities of Leeds, Sheffield and York
<http://eprints.whiterose.ac.uk/>

This is an author produced version of a paper published in **Studies in Computational Intelligence**.

White Rose Research Online URL for this paper:

<http://eprints.whiterose.ac.uk/75425/>

Published paper:

Koh, ATM (2013) *A Metaheuristic Framework for Bi-level Programming Problems with Multi-disciplinary Applications*. In: *Metaheuristics for Bi-level Optimization*. *Studies in Computational Intelligence*, 482 . Springer, Berlin, Heidelberg , 153 - 187 .

http://dx.doi.org/10.1007/978-3-642-37838-6_6

A Meta-heuristic Framework for Bi-Level Programming Problems with MultiDisciplinary Applications

Andrew Koh

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. Several problems arising in engineering and economics can be cast within the bi-level programming framework. The bi-level programming model is also known as a Stackleberg or leader-follower game in which the leader chooses his variables so as to optimise his objective function, taking into account the response of the follower(s) who separately optimise their own objectives, treating the leader's decisions as exogenous. In this chapter, we present a unified framework fully consistent with the Stackleberg paradigm of bi-level programming that allows for the integration of meta-heuristic algorithms with traditional gradient based optimisation algorithms for the solution of bi-level programming problems. In particular we employ Differential Evolution as the main meta-heuristic in our proposal. We subsequently apply the proposed method (DEBLP) to a range of problems from many fields such as transportation systems management, parameter estimation and game theory. It is demonstrated that DEBLP is a robust and powerful search heuristic for this class of problems characterised by non smoothness and non convexity.

1 Introduction

This paper introduces a meta-heuristic framework for solving the Bi-level programming Problem (BLPP) with a multitude of applications [26, 74]. As a historical footnote, the term “bi-level programming” was first coined in a technical report by Candler and Norton in [20] who were concerned with general multilevel programming problems. The BLPP is a special case of a multilevel programming problem restricted to two levels. Prior to that time, the BLPP was known simply as a math-

Andrew Koh
Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, United Kingdom e-mail:
a.koh@its.leeds.ac.uk

ematical program with an optimisation problem in the constraints [18] but had already found military applications [19]. In economics and game theory, a BLPP is a Stackleberg [106] or “leader-follower” game (see Fig. 1) in which the leader chooses his variables so as to optimise his objective but continues to take into account the response of the follower(s) who when independently optimising their separate objectives, treat the leader’s decisions as an exogenous input [72].

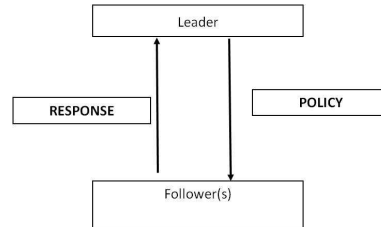


Fig. 1 Pictorial Representation of a BLPP

BLPPs possess in common the following three characteristics [86, 117]:

- The decision-making units are interactive and exist within a 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 BLPP has been a subject of intense research and several notable volumes have been published to date [7, 32, 72, 87]. At the same time applications of BLPP can be found in fields as diverse as chemical engineering [49], robot motion planning and control [72], production planning [8] occurring in a multitude of disciplines [7]. In tandem, there has been much work on the development of solution methodologies (see [26, 32] for a review of these).

This chapter is structured as follows. In Section 2 we formally introduce the BLPP and provide a brief and by not means exhaustive review of solution methodologies for the BLPP. In Section 3 we discuss the Differential Evolution for Bi-Level Programming (DEBLP) meta-heuristic introduced in [61] and apply it to problems in transportation systems management in Section 4. To emphasise the multidisciplinary applications to which DEBLP is applicable, Section 5 provides examples of BLPPs arising from parameter estimation problems. Section 6 introduces a method for handling constraints integrating recent developments in evolutionary algorithms to propose a method to handle constraints in the leader’s problem. Having set up this framework, we are ready in Section 7 to apply DEBLP to solving Generalised Nash Equilibrium Problems (GNPE) which when formulated as BLPPs are in fact characterised by constraints in the leader’s objective. Section 8 summarises and provides extensive directions for further research on problems related to the topic of BLPPs.

2 The Bi-Level Programming Problem

2.1 A General BLPP

We can write a generic BLPP as the system of equations in Eq. 1. The unique feature of Eq. 1 is that the constraint region is implicitly determined by yet another optimisation problem¹. This constraint is always active. The upper level problem denoted as Program U, is given in Eq. 1a,

$$\text{Program } U \left\{ \begin{array}{l} \min_{\mathbf{x} \in X} U(\mathbf{x}, \mathbf{y}) \\ \text{subject to} \\ G(\mathbf{x}, \mathbf{y}) \leq 0 \\ E(\mathbf{x}, \mathbf{y}) = 0 \end{array} \right. \quad (1a)$$

where for given \mathbf{x} , \mathbf{y} is the solution to the lower level program (Program L) in 1b:

$$\text{Program } L \left\{ \begin{array}{l} \min_{\mathbf{y} \in Y} L(\mathbf{x}, \mathbf{y}) \\ \text{subject to} \\ g(\mathbf{x}, \mathbf{y}) \leq 0 \\ e(\mathbf{x}, \mathbf{y}) = 0 \end{array} \right. \quad (1b)$$

In the formulation in Eq. 1 we define the following mappings: $U, L : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^1$, $G : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{q_1}$, $g : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{q_2}$, $E : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{r_1}$, $e : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{r_2}$. In the general case the objectives and constraints at both levels are non-linear. The sets X and Y representing the search domains for 1a and 1b respectively are defined as follows: $X = \{(x_1, x_2, \dots, x_{n_1})^\top \in \mathbb{R}^{n_1} \mid x_i^l \leq x_i \leq x_i^u, i = 1, \dots, n_1\}$ and $Y = \{(y_1, y_2, \dots, y_{n_2})^\top \in \mathbb{R}^{n_2} \mid y_j^l \leq y_j \leq y_j^u, j = 1, \dots, n_2\}$ with \top denoting the transpose. Arising from the ‘‘leader-follower’’ analogy of BLPPs, we use the terms leader’s variables and upper level variables interchangeably when referring to \mathbf{x} .

2.2 Mathematical Programs with Equilibrium Constraints

We also define a class of BLPPs known as the Mathematical Programs with Equilibrium Constraints (MPECs). MPECs are BLPPs where the lower level problem consists of a variational inequality (VI) [26].

¹ Hence the original name of mathematical programs with optimisation problems in the constraints chosen by Bracken and McGill in [18].

$$\text{Program } U \left\{ \begin{array}{l} \min_{\mathbf{x} \in X} U(\mathbf{x}, \mathbf{y}) \\ \text{subject to} \\ G(\mathbf{x}, \mathbf{y}) \leq 0 \\ E(\mathbf{x}, \mathbf{y}) = 0 \end{array} \right. \quad (2a)$$

where for given \mathbf{x} , \mathbf{y} is the solution of the VI in Program L 2b:

$$L(\mathbf{x}, \mathbf{y})^\top (\mathbf{y} - \mathbf{y}^*) \geq 0, \forall \mathbf{y} \in Y(x) \quad (2b)$$

Another class of problems closely related to MPECs are Mathematical Programs with *Complementarity Constraints* [67] which feature in place of a VI, a Complementarity Problem instead in Program L . However since the VI is a generalization of the Complementarity Problem [56, 82], we will treat these two categories as synonymous for the purposes of this chapter and neglect the theoretical distinctions. We return in Section 4 to give an example of MPECs that arise naturally in transportation systems management.

2.3 Solution Algorithms for the BLPP

When all functions (both objectives and constraints) at both levels are linear and affine, this class of problems is known as the linear-BLPP. However even in this deceptively “simple” case the problem is still nondeterministic polynomial time hard [11]. Even when both the upper level and the lower level are convex programming problems, the resulting BLPP itself can be non-convex [12]. Non convexity suggests the possibility of multiple local optima. Ben-Ayed and Blair [11] demonstrated the failure of both the Parametric Complementarity Pivot Algorithm [13] and the Grid Search Algorithm [5] to locate the optimal solution. Since then, progress has been made in solving the linear-BLPP and techniques including implicit enumeration [21], penalty based methods [2] and methods based on Karush-Kuhn-Tucker (KKT) conditions [41] have been developed. (See [117] for a detailed review of the algorithms available for the linear-BLPP).

Turning to solution algorithms for the general BLPP, several intriguing attempts have been proposed to solve it. One early proposal was the Iterative Optimisation Algorithm (IOA) [3, 107]. This method involved solving the Program U for fixed \mathbf{y} and using the solution thus obtained to solve the lower level problem, Program L , and repeatedly iterating between the two programs until some convergence criteria is met. However the IOA was shown to be an exact method for solving a Cournot Nash game [40, 42] rather than the Stackleberg game that the BLPP reflects. The IOA implicitly assumes that the leader is myopic as he does not take into account the follower’s reaction to his policy [42]. To be consistent with the Stackleberg model, the leader must be modelled as endowed with knowledge of the follower’s reaction function which the leader knows the follower will obey.

The primary difficulty with solving MPECs is that they fail to satisfy certain technical conditions (known as constraint qualifications) at any feasible point [23, 102].

The penalty interior point algorithm (PIPA) was proposed in [72]. Unfortunately a counterexample in [67] demonstrates that PIPA can converge to a nonstationary point. Subsequent research has led to the development of many other techniques to solve the MPEC such as the piecewise sequential quadratic programming in [72], branch-and-bound [6], nonsmooth approaches [32, 87] and smoothing methods [38].

Wrapping up this section, we summarise briefly the use of methods based on meta-heuristics. Meta-heuristics including stochastic optimisation techniques are recognised as useful tools for solving problems such as the BLPPs which do not necessarily satisfy the classical optimisation assumptions of continuity, convexity and differentiability. Techniques include Simulated Annealing (SA) [1], Tabu Search (TS) [48], Genetic Algorithms (GA) [47], Ant Colony Optimisation (ACO) [34], Particle Swarm Optimisation (PSO) [57] and Differential Evolution (DE) [94, 95, 108].

SA was used to optimise a chemical process plant layout design problem formulated as a BLPP in [101] and a Network Design Problem formulated as an MPEC [43]. ACO techniques for BLPPs are found in [93]. GAs have been used to solve BLPPs in *inter alia* [73, 86, 111, 114, 122]. PSO was applied to BLPPs in e.g. [126]. DE was used for BLPPs in [61] where an example demonstrated the inability of the TS method implemented in [96] to locate the global optima of a test function. Despite their reported successes in tackling very difficult problems, it must be emphasised that heuristics provide no guarantee of convergence to even a local optimum. Despite this heuristics have been successfully used to solve a variety of difficult problems such as the BLPP.

3 Differential Evolution for Bi-Level Programming (DEBLP)

Differential Evolution for Bi-Level Programming (DEBLP) was initially proposed in [61] to tackle BLPPs arising in transportation systems management. It is developed from the GA Based Approach proposed in [111, 122] but substitutes the use of binary coded GA strings with real coded DE [95] as the meta-heuristic instead.

DE is a simple algorithm that utilises perturbation and recombination to optimise multi-modal functions and has already shown remarkable success when applied to the optimisation of numerous practical engineering problems [94, 95, 108]. On the other hand, many years of research have resulted in the development of a plethora of robust gradient based algorithms for tackling many operations research questions posed as non-linear programming problems (NLP) [9, 70, 85]. If we momentarily ignore the upper level problem, then for fixed \mathbf{x} , Eq. 1b is effectively an NLP² which can be tackled by dedicated NLP tools such as sequential quadratic programming [9, 70, 85]. Such considerations motivated the development of the DEBLP meta-heuristic which sought to synergise DE's well-documented global search capability to optimise the upper level problem with the dedicated NLP tools focused on solving

² Note that for fixed \mathbf{x} , the lower level problem in the MPEC in Eq. 2b can also be solved using deterministic methods. See e.g. [82] for a review of deterministic solution algorithms for VIs.

the lower level problem. More importantly, as we shall emphasise later, DEBLP continues to maintain the crucial “leader-follower” paradigm upon which the BLPP is founded.

In the rest of this section, we provide an overview of the operation of the DEBLP algorithm and discuss some of its limitations. However we temporarily neglect consideration of the $q_1 + r_1$ upper level constraints in Program U . Our discussion of the procedure used to ensure satisfaction of the upper level inequality and/or equality constraints is postponed till later (see Section 6).

3.1 Differential Evolution

Conventional deterministic optimisation methods generally operate on a single trial point, transforming it using search directions computed based on first (and possibly, second) order conditions until some criteria measuring convergence to a stationary point is satisfied [9, 70, 85]. On the other hand, population based meta-heuristics such as DE operate with a population of trial points instead. The idea here is that of improving each member throughout the operation of the algorithm by way of an analogy with Darwin’s theory of evolution³.

Let there be π members in such a population of trial points. Specifically we denote the population at iteration it as \mathcal{P}^{it} . An illustration of such a population is given in Eq. 3. Each member of \mathcal{P}^{it} representing a single trial point $\mathbf{x}_k^{it} = (x_{k,1}^{it}, \dots, x_{k,n_1}^{it})$, $k = \{1, \dots, \pi\}$, also known as an individual, is a n_1 dimensional vector that represents the upper level variables (see Eq. 1a). To avoid notational clutter, we drop the it superscript as long as it does not lead to confusion. Without loss of generality, we will assume minimization. The DEBLP algorithm is outlined in Algorithm 1 which we elaborate upon in the ensuing paragraphs of this section.

$$\mathcal{P}^{it} = \begin{pmatrix} \mathbf{x}_1^{it} \\ \vdots \\ \mathbf{x}_k^{it} \\ \vdots \\ \mathbf{x}_\pi^{it} \end{pmatrix} = \begin{pmatrix} x_{1,1}^{it} & x_{1,2}^{it} & \cdots & x_{1,n_1}^{it} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k,1}^{it} & x_{k,2}^{it} & \cdots & x_{k,n_1}^{it} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\pi,1}^{it} & x_{\pi,2}^{it} & \cdots & x_{\pi,n_1}^{it} \end{pmatrix} \quad (3)$$

³ Hence some of these methods are sometimes referred to as evolutionary algorithms in the literature.

Algorithm 1 Differential Evolution for Bi-Level Programming (DEBLP)

-
1. Randomly generate parent population \mathcal{P} of π individuals.
 2. Evaluate \mathcal{P}
 - set iteration counter $it = 1$
 3. While stopping criterion not met, do:
 - For each individual in \mathcal{P}^{it} , do:
 - (a) Mutation and Crossover to create a single child from individual.
 - (b) Evaluate the child using a hierarchical strategy.
 - (c) Selection: If the child is fitter than the individual, the child replaces the parent. Otherwise, the child is discarded.
 - End For
 - $it = it + 1$
-

3.1.1 Generate Parent Population

When the algorithm begins, real parameters in each dimension i of each member k of \mathcal{P} , that comprise the parent population, are randomly generated within the lower and upper bounds of the domain of the BLPP as in Eq. 4.

$$x_{k,i} = rand(0, 1)(x_i^u - x_i^l) + x_i^l, k \in \{1, \dots, \pi\}, i \in \{1, \dots, n_1\}. \quad (4)$$

In Eq. 4 $rand(0, 1)$ is a pseudo random number generated from an uniform distribution between 0 and 1.

3.1.2 Evaluation

The evaluation process to determine the fitness⁴ of a trial point in the population has to be developed within the Stackleberg model [106] since we have to specifically model the leaders taking into account the response (reaction) of the followers to his strategy \mathbf{x} . One way to accomplish this is via a “two stage” or hierarchical strategy which is achieved as follows.

In the first stage, for each individual k vector of the leader’s decision variables \mathbf{x}_k , we solve Program L i.e. Eq. 1b to obtain \mathbf{y} by using deterministic methods such as linear programming or sequential quadratic programming [9, 70, 85]. With \mathbf{y} so obtained, we are then able to carry out the second stage which involves computing the value of the upper level objective U , corresponding to each individual vector of the leader’s decision variables input in the first stage.

It is worth highlighting that this procedure is *different* from the IOA described earlier in Section 2 as DEBLP obviates any iteration between the two levels. Instead, entirely consistent with the “leader-follower” paradigm, the leader’s vector \mathbf{x}_k being

⁴ The term “fitness” used in such evolutionary meta-heuristics is borrowed from its analogy with evolution where Darwin’s concept of survival of the fittest is a cornerstone. In minimization problems, when comparing two individuals, the fitter individual is the one that evaluates to a *lower* upper level objective.

manipulated by DE is offered as an exogenous input to the lower level program to be solved in the first stage. One obvious drawback of doing this is the resulting increase in computational burden which has been significantly reduced by advances in computing power.

3.1.3 Mutation and Crossover

The objective of mutation and crossover is to produce a child vector \mathbf{w}_k from the parent. This is accomplished by stochastically adding to the parent vector the factored difference of two other randomly chosen vectors from the population as shown in Eq. 5.

$$w_{k,i} = \begin{cases} x_{s1,i} + \lambda(x_{s2,i} - x_{s3,i}) & \text{if } \text{rand}(0,1) < \chi \text{ or } i = \text{intr}(1, n_1) \\ x_{k,i} & \text{otherwise} \end{cases} \quad (5)$$

In Eq. 5, $s1, s2$ and $s3 \in \{1, 2, \dots, \pi\}$ are randomly chosen population indices distinct from each other and also distinct from the current population member index k . $\text{rand}(0, 1)$ is a pseudo random real number between 0 and 1 and $\text{intr}(1, n_1)$ is a pseudo random integer between 1 and n_1 . The mutation factor $\lambda \in (0, 2)$ is a parameter which controls the magnitude of the perturbation and $\chi \in [0, 1]$ is a probability that controls the ratio of new components in the offspring. The or condition in Eq. 5 ensures that the child vector \mathbf{w}_k will differ from its parent \mathbf{x}_k in at least one dimension.

We stress that the mutation and crossover strategy shown in Eq. 5 is not the only possible strategy available though this is the one used in this work. Other strategies are found in [94, 95, 108]. Nevertheless all the strategies of DE reflect a common theme: the creation of the child vector \mathbf{w}_k via the arithmetic recombination of randomly chosen vectors along with addition of difference vector(s) typified in Eq. 5.

3.1.4 Enforce Bound Constraints

Mutation and crossover can however produce child vectors that lie outside the bounds of the original problem specification. There are several ways to ensure satisfaction of these constraints. One could set the parameter equal to the limit exceeded or regenerate it within the bounds. Alternatively, following [94], we reset out of bound values in each dimension i half way between its pre-mutation value and the bound violated as shown in Eq. 6.

$$w_{k,i} = \begin{cases} \frac{x_{k,i} + x_i^l}{2} & \text{if } w_{k,i} < x_i^l \\ \frac{x_{k,i} + x_i^u}{2} & \text{if } w_{k,i} > x_i^u \\ w_{k,i} & \text{otherwise} \end{cases} \quad (6)$$

3.1.5 Selection

Once the hierarchical evaluation process is carried out on the child vector \mathbf{w}_k produced, we can compare the fitness obtained with that of its parent \mathbf{x}_k . This means that comparison is against the same k parent vector⁵ on the basis of whichever of the two gives a lower value for Program U . Assuming minimization the one that produces a lower value survives to become a parent in the following generation as shown in Eq. 7.

$$\mathbf{x}_k^{it+1} = \begin{cases} \mathbf{w}_k^{it} & \text{if } U(\mathbf{w}_k^{it}, L(\bullet)) \leq U(\mathbf{x}_k^{it}, L(\bullet)) \\ \mathbf{x}_k^{it} & \text{otherwise} \end{cases} \quad (7)$$

These steps are repeated until some user specified termination criteria is met, and this is usually when it reaches the maximum number of iterations, although other criteria are possible [95].

3.2 Control Parameters of DE

Unless otherwise stated, for all experiments reported throughout this chapter we used a Mutation Factor, λ , of 0.9 and a Probability of Crossover, χ , of 0.9. The population size, π , and the maximum number of iterations allowed varied for each of the BLPPs we investigated and these will be clearly stated in the relevant sections. Because DEBLP is a stochastic meta-heuristic, we always carry out 30 independent runs with different random seeds. All numerical experiments were conducted using MATLABTM 7.8 running on a 32 bit WindowsTM XP machine with 4 GB of RAM.

3.3 Implicit Assumptions of DEBLP

Through the rest of this paper we will demonstrate in examples from various disciplines that DEBLP is a powerful and robust solution methodology for handling a variety of problems formulated as BLPPs. However we are cognizant at the outset two key limitations of our approach:

1. DEBLP is a heuristic: with its strength arising from it avoiding reliance on the objective functions being differentiable and/or satisfying convexity properties and hence able to handle a large class of intrinsically non smooth problems. However it should be recognised that for this very reason, it is not generally possible to establish convergence of the algorithm to even a local optimum.
2. DEBLP implicitly assumes that the Program L is convex for fixed \mathbf{x} and can be solved to global optimality by deterministic methods and that failure to solve the

⁵ This is sometimes referred to as “one to one” comparison in [94, 95, 108].

lower level problem to global optimality does not affect the solution of Program U .

This section has focused on defining the motivation for, and outlining, the DEBLP meta-heuristic which sought to synergise the exploratory power of DE with robust deterministic algorithms focused on solving the lower level problem. Recognizing its limitations, in the next section, we apply DEBLP to control problems arising from Transportation Systems Management formulated as BLPPs where the lower level program is shown to be convex for a given tuple of the leader's variables.

4 Applications to Transportation Systems Management

In this section, we study two problems in transportation systems management. In applications, the leader in Program U could be thought of as a regulatory authority applying control strategies (policy) that influence the travel choices of the followers who are the highway users on the road network. It will be shown under certain assumptions, the followers problem can be established as a VI thus the problems under consideration are MPECs.

4.1 The Lower Level Program in Transportation

In the transportation systems management literature, Program L has an interpretation in that it is the mathematical formulation representing the follower's (road user's) route choice [15] on a highway network. This is often referred to as the Traffic Assignment Problem (TAP). Traffic assignment aims to determine the number of vehicles and the travel time on different road sections of a traffic network, given the travel demand between different pairs of origins and destinations [60].

Definition 1 [115] *The journey times on all the routes actually used are equal, and not greater than those which would be experienced by a single vehicle on any unused route.*

The TAP is founded on the behavioral premise of Wardrop's User Equilibrium as given in Definition 1. In effect this states that user equilibrium is attained when no user can decrease his travel costs by unilaterally changing routes. The TAP provides the link flow vector (v) when user equilibrium is attained.

To facilitate exposition of Program L , consider a transportation network represented as a graph with N nodes and A links/arcs, and let:

- P : the set of all paths/routes in the network,
- H : the set of all Origin Destination (OD) pairs in the network,
- P_h : the set of paths connecting an OD pair $h, h \in H$,
- F_p : the flow on route/path $p, p \in P$,

v_a : the link flow on link a $\mathbf{v} = [v_a], a \in A$,
 $c_a(v_a)$: the travel cost of utilising the link a , as a function of link flow v_a on that link only, $\mathbf{c}(\mathbf{v}) = [c_a(v_a)], a \in A$
 c_p : the travel cost of path $p, p \in P$,
 δ_{ap} : a dummy variable that is 1 if the path $p, p \in P$ uses link $a, a \in A$, 0 otherwise
 and
 Ω : the set of feasible flows and demands.

On the demand side, we assume that there is an amount of demand $d_h, h \in H$ ($d_h \geq 0$) wishing to travel between OD pair h and μ_h is the minimum travel cost that OD pair $h, h \in H$.

4.1.1 TAP as a Variational Inequality

Lemma 1. *Wardrop's Equilibrium Condition of route choice implies that at equilibrium the following conditions are simultaneously satisfied:*

$$\begin{aligned}
 F_{p \in P_h} \geq 0 &\Leftrightarrow c_{p \in P_h} = \mu_h \quad \forall h \in H, \forall p \in P; \\
 F_{p \in P_h} = 0 &\Leftrightarrow c_{p \in P_h} \geq \mu_h \quad \forall h \in H, \forall p \in P; \\
 d_h &= \sum_{p \in P_h} F_p \quad \forall h \in H, \forall p \in P;
 \end{aligned}$$

Lemma 1 states that path p connecting OD pair h will be used by the travellers if and only if the cost of travelling on this route is the minimum travel cost between that OD pair. The Variational Inequality (VI) in Eq. 8 restates Wardrop's Equilibrium Condition.

$$\text{Find } \mathbf{v}^* \in \Omega \text{ such that } \mathbf{c}(\mathbf{v}^*)^T (\mathbf{v} - \mathbf{v}^*) \geq 0, \forall \mathbf{v} \in \Omega \quad (8)$$

Proposition 1. *The solution of the Variational Inequality defined in Eq. 8 results in a vector of link flows demands ($\mathbf{v}^* \in \Omega$) that satisfies Wardrop's Equilibrium Condition of route choice given by Lemma 1.*

Proof. For a proof of Proposition 1, see [28, 105]. \square

4.1.2 Convex Optimisation Reformulation

In the particular instance (and in the cases considered in this chapter) when the travel cost of using a link is dependent only on its own flow⁶, there exists an equivalent convex optimisation program for the VI (Eq. 8) as shown in Eq. 9.

$$\min_{\mathbf{v}} L = \sum_{\forall a} \int_0^{v_a} c_a(z) dz \quad (9a)$$

⁶ This is known as the separability assumption.

Subject to:

$$\sum_{p \in P_h} F_p = d_h, h \in H \quad (9b)$$

$$v_a = \sum_{p \in P} F_p \delta_{ap}, a \in A \quad (9c)$$

$$F_p \geq 0, p \in P. \quad (9d)$$

The objective of the program in Eq. 9 is a mathematical construct, with no behavioral interpretation, employed to solve for the equilibrium link flows that satisfies Wardrop's Equilibrium Condition [103]. In this program, the first constraint states that the flow on each route used by each OD pair is equal to the total demand for that OD pair. The second constraint is a definitional constraint which stipulates that the flow on a link comprises flow on all routes that use that link. The last constraint restricts the equilibrium flows and demands to be non negative. These linear constraints define Ω . Since Ω is closed and convex, the equilibrium link flows $\mathbf{v}^* \in \Omega$ are unique [15]. In practice, it is usually the case that traffic assignment algorithms (see examples in texts such as [89, 103]) are used to solve Program L .

4.2 Continuous Optimal Toll Pricing Problem (COTP)

The continuous optimal toll pricing problem involves selecting an optimal toll level for each predefined tolled link in the network [11]. With a view to controlling congestion, there has been renewed interests by transportation authorities globally to study this "road pricing" problem (e.g. Singapore, London, Stockholm).

4.2.1 Model Formulation

In addition to the notation defined at the start of this section, we introduce the following notation to describe the COTP. Let:

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

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 on link a , $a \in T$

Total travel 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 acts as a proxy for the resource cost to the economy of the highway system. The objective of the upper level decision maker in the COTP is to minimise this by encouraging more efficient routing of traffic by levying tolls on the road users in the network. The upper level program is the system in Eq. 10.

$$\min_{\tau} U = \sum_{a \in A} v_a t_a(v_a) \quad (10a)$$

Subject to:

$$\begin{aligned} \tau_a^{\min} &\leq \tau_a \leq \tau_a^{\max}, & a \in T \\ \tau_a &= 0, & a \notin T \end{aligned} \quad (10b)$$

Note however that \mathbf{v} can only be obtained by solving Program L in Eq. 9. Thus in terms of Figure 1, the policy variables \mathbf{x} is the toll vector τ and the follower's response is the traffic routing that manifests in the vector of link flows on the road network \mathbf{v} that in turns affect the leader's objective.

Recall that in defining the lower level program in Eq. 9, the road user was assumed to consider the travel cost of utilising an arc a , $a \in A$. Eq. 11 maps the travel time $t_a(v_a)$ on an arc a , into the equivalent travel costs $c_a(v_a)$.

$$c_a(v_a) = \begin{cases} t_a(v_a) + \tau_a & \text{if } a \in T \\ t_a(v_a) & \text{otherwise} \end{cases} \quad (11)$$

4.2.2 Previous Work on the COTP

Various solution algorithms have been proposed for the COTP. Yang and Lam proposed a linearisation based method that uses derivative information to form approximations to the upper level objective [118] known as a sensitivity based analysis algorithm (SAB). However it has been pointed out [122] the global optimality of the SAB algorithm is not assured and that obtaining a local optimum is indeed possible. Another derivative-based method was derived from constraint accumulation [66]. A review of algorithms for the COTP is found in [111].

4.2.3 Example

We illustrate the use of DEBLP to solve the COTP with an example from [118]. Fig. 2 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. The link travel times $t_a(v_a)$ take the explicit function forms as given in Eq. 12.

$$t_a(v_a) = t_a^0 \left(1 + 0.15 \left(\frac{v_a}{Cap_a} \right)^4 \right) \quad (12)$$

In Eq. 12, t_a^0 is the free flow travel time of the link a and Cap_a is the capacity of link a . The parameter details for the network and the upper bound on tolls τ_a^{\max} are found in [118] and given in Table 1. Note that $\tau_a^{\min} = 0, \forall a \in T$.

For this example, we use a population size, π , of 20 and allowed a maximum of 50 iterations in each of 30 runs. Table 2 compares the results of DEBLP with that of two deterministic algorithms (direct from [118] and our implementation of the algorithm of [66] together with a GA based method from [122]). UPO refers to the value of (Upper level) Objective in Eq. 10. It can be seen from Table 2 that the four different algorithms provided different tolls underlying the multimodal nature of this problem. However the upper level objective function values are the same in all cases. This bears testimony to the multimodal nature of the COTP where many different toll vector tuples could potentially result in attaining the same upper level objective function value.

Table 1 Network Parameters for COTP Example

Link a	τ_a^0	Cap a	τ_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

Table 2 Comparison of existing against DEBLP results for COTP Example

Method	Deterministic		Stochastic	
	Tolls [118]	Method of [66]	[122]	DEBLP
Link 1	3.82	2.667	4.324	3.824
Link 2	4.265	3.548	4.976	3.92
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	628.6	628.6	628.6	628.6

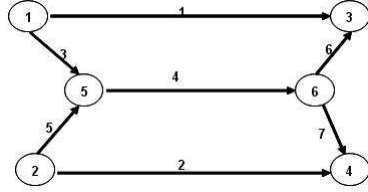


Fig. 2 Network for COTP Example [118]

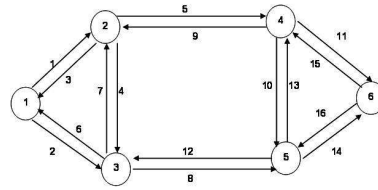


Fig. 3 Network for CNDP Example 1 [27]

4.3 Continuous Network Design Problem

The continuous network design problem (CNDP) aims to determine the optimal capacity enhancements of existing facilities of a traffic network [43]. Care has to be taken when solving the CNDP because additional capacity can counter productively increase the total network travel time and this is a phenomenon known as Braess's paradox [17]. Braess's paradox has been known to occur in transportation [17] and telecommunication networks [63].

4.3.1 Model Formulation

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

- κ : the set of links that have their individual capacities enhanced, $\kappa \subseteq A$.
- β : the vector of capacity enhancements, $\beta = [\beta_a], a \in \kappa$
- $\beta_a^{\max}, \beta_a^{\min}$: the upper and lower bounds of capacity enhancements, $a \in \kappa$.
- d_a : the monetary cost of capacity increments per unit of enhancement, $a \in \kappa$.
- Cap_a^0 : existing capacity of link a , $a \in A$.
- θ : conversion factor from monetary investment costs to travel cost units.

In the CNDP, the regulator aims to minimise the sum of the total travel times and investment costs with constraints on the amount of capacity additions 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. Hence the CNDP seeks a $|\kappa|$ dimension vector of capacity enhancements optimal to the following BLPP in Eq. 13:

$$\min_{\tau} U = \sum_{a \in A} v_a t_a(v_a) + \sum_{\forall a \in \kappa} \theta d_a \beta_a \quad (13a)$$

subject to:

$$\begin{aligned} \beta_a^{\min} &\leq \beta_a \leq \beta_a^{\max} & a \in \kappa; \\ \beta_a &= 0 & a \notin \kappa \end{aligned} \quad (13b)$$

where v is the solution of a lower level TAP (Program L) Eq. 9, parameterised in the vector of capacity enhancements for the fixed demand case. We map the travel times to the travel costs by means of Eq. 14.

$$c_a(v_a) = \begin{cases} t_a^0 (1 + 0.15 (\frac{v_a}{Cap_a^0 + \beta_a})^4) & \text{if } a \in \kappa \\ t_a^0 (1 + 0.15 (\frac{v_a}{Cap_a^0})^4) & \text{if } a \notin \kappa \end{cases} \quad (14)$$

4.3.2 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 CNDP into a single level continuously differentiable problem using a marginal function and solved the resulting formulation with the Augmented Lagrangian method [75]. Chiou investigated several variants of the descent based Karush-Khun-Tucker (KKT) approaches [24]. Stochastic meta-heuristics have also been used; GAs were applied in [27] and the use of SA has been reported in [43].

4.3.3 Example 1: Hypothetical Network

The network for the first example is taken from [27] and reproduced in Fig. 3. This network has 6 nodes and 2 OD pairs; the first between nodes 1 and 6 of 10 trips and the second, between nodes 6 and 1 of 20 trips. Please refer to [27] for the link parameter details. Note that $\beta_a^{\min} = 0$ and $\beta_a^{\max} = 20$, $\forall a \in \kappa, \kappa \subseteq A$ as in [27]. We assumed a population size, π , of 20 and allowed a maximum of 150 iterations. Table 3 summarises the results that have been reported previously and compares it with the results reported in our paper. UPO refers to the value of (upper level) objective in Eq. 13. NFE is the number of function evaluations. Note that the number of lower level programs solved equal to population size multiplied by the maximum number of iterations allowed. SD is the standard deviation over 30 runs. Our results are based on the mean of these 30 runs. Though the SD of the GA method is much lower, [27] also reported using local search method to aid the search process which accounts for the higher NFE as well.

4.3.4 Example 2: Sioux Falls Network

The second example is the CNDP for the Sioux Falls (South Dakota, USA) network with 24 nodes, 76 links and 552 OD pairs. The network parameters and OD details are found in [75]. Only 10 links out of the 76 are subject to improvements.

While this network is clearly larger and arguably more realistic, the problem dimension (i.e. leader's variables simultaneously optimised) is smaller than in Example 1, since 10 links are subject to improvement rather than the 16 links in the former. This offers an explanation as to why the number of function evaluations (NFE) reported in all studies compared is less than for the first example. The results are compared in Table 4. Our results show the mean of 30 runs with different random seeds.

It can be deduced from Table 4 that DEBLP is able to locate the global optimum; again with a lesser number of iterations than the SA method in [43]. More interestingly, DEBLP required less iterations than the deterministic method of [75]. The standard deviation is also very low which suggests that this heuristic is reasonably robust as well.

Table 3 Comparison of existing against DEBLP results for CNDP Example 1

Method:	Deterministic				Stochastic	
Source	[24]	[75]	[43]	[27]	DEBLP	
UPO	534	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

Table 4 Comparison of existing against DEBLP results for CNDP Example 2

Method:	Deterministic			Stochastic		
Source	[24]	[75]	[43]	DEBLP		
UPO	82.57	81.75	80.87	80.74		
NFE	10	2,000	3,900	1,600		
SD	—Not Reported—				0.002	

5 Applications to Parameter Estimation Problems

In this section we derive the Error-In-Variables model and show that it can be formulated as a BLPP and apply it to 2 examples from [49]. Parameter estimation is an important step in the verification and utilization of mathematical models in many fields of science and engineering [37, 49, 59]. In the classical least-squares approach to parameter estimation, it is implicitly assumed that the set of independent variables is not subject to measurement errors [46]. On the other hand, the error-in-variables (EIV) approach assumes that there are measurement errors in *all* variables [16, 98].

5.1 Formulation of EIV Model

We consider models of the implicit form as in Eq. 15.

$$f(x, y) = 0 \quad (15)$$

In Eq. 15, x is the vector of n_1 unknown parameters, y is the vector of n_2 measurement variables and f is the system of algebraic functions. The measured variables are the sum of the true values ζ_m which are unknown and the additive error term ε_m at the data point m as shown in Eq. 16.

$$y_m = \zeta_m + \varepsilon_m \quad (16)$$

We assume that the error is normally distributed with zero mean and possessing a known covariance matrix. The vector of unknown parameters x can be estimated from the solution of the constrained optimisation problem in Eq. 17.

$$\begin{aligned} & \min_{\hat{x}, \hat{y}} \sum_{m=1}^M (\hat{\mathbf{y}}_m - \mathbf{y}_m)^\top \Lambda^{-1} (\hat{\mathbf{y}}_m - \mathbf{y}_m) \\ & \text{subject to} \\ & f(\hat{\mathbf{y}}_m, \hat{x}) = 0, m = 1, \dots, M \end{aligned} \quad (17)$$

As mentioned, we do not know the true values of ζ_m . However they are approximated from the optimisation as the fitted variables $\hat{\mathbf{y}}_m$. Assuming that the covariance matrix Λ is the same in each experiment and diagonal, we write Eq. 17 as Eq. 18.

$$\begin{aligned} & \min_{\hat{x}, \hat{y}} \sum_{m=1}^M \sum_{i=1}^{n_2} \frac{(\hat{y}_{m,i} - y_{m,i})^2}{\sigma_i^2} \\ & \text{subject to} \\ & f(\hat{\mathbf{y}}_m, \hat{x}) = 0, m = 1, \dots, M \end{aligned} \quad (18)$$

In Eq. 18, σ_i is the standard deviation of variable i in all the experiments. Following [49], we can write the EIV model as a BLPP of the form of Eq. 19.

$$\text{Program } U \left\{ \min_{\hat{\mathbf{x}}} \sum_{m=1}^M \sum_{i=1}^{n_2} \frac{(\hat{y}_{m,i} - y_{m,i})^2}{\sigma_i^2} \right. \quad (19a)$$

where for given \mathbf{x} , \mathbf{y} is the solution to the lower level program (Program L):

$$\text{Program } L \left\{ \begin{array}{l} \min_{\hat{\mathbf{x}}, \hat{\mathbf{y}}} \sum_{m=1}^M \sum_{i=1}^{n_2} \frac{(\hat{y}_{m,i} - y_{m,i})}{\sigma_i^2} \\ \text{subject to} \\ \mathbf{f}(\hat{y}_{m,i}, \hat{\mathbf{x}}) = 0, \quad m = 1, \dots, M, i = 1, \dots, n_2 \end{array} \right. \quad (19b)$$

For a survey of the alternative optimisation based formulations of the EIV model, the reader is referred to [59].

5.2 Examples

We present 2 examples of the EIV model that were solved using deterministic methods in the cited references. Note that we consider only a single common variance term for all variables and we can eliminate it from further consideration. In all our experiments of DEBLP we assumed a population size, π , of 20 and allowed a maximum of 100 iterations.

5.2.1 Example 1: “Kowalik Problem”

Consider the model due to Moore et al in [76] known as the “Kowalik Problem” where we estimate the equation of the form in Eq. 20.

$$\hat{y}_{m,1} = \frac{x_1 y_{m,2}^2 + x_1 x_2 y_{m,2}}{y_{m,2}^2 + y_{m,2} x_3 + x_4} \quad (20)$$

We have 11 data points for this model (see [49] for the data set). It is assumed that $y_{m,1}$ contains errors, and $y_{m,2}$ is error-free. The resulting BLPP is shown in Eq. 21. Notice that the lower level equality constraint in Eq. 21 is the model formulation hypothesised in Eq. 20.

$$\begin{array}{l} \min_{\hat{\mathbf{x}}} \sum_{m=1}^{11} (\hat{y}_{m,1} - y_{m,1})^2 \\ \text{subject to} \\ \min_{\mathbf{y}} \sum_{m=1}^{11} (\hat{y}_{m,1} - y_{m,1})^2 \\ \hat{y}_{m,1} (y_{m,2}^2 + y_{m,2} x_3 + x_4) - x_1 y_{m,2}^2 - x_1 x_2 y_{m,2} = 0 \end{array} \quad (21)$$

30 runs of DEBLP were performed for this problem with a maximum of 100 iterations allowed per run and a population size, π of 20. Following [49], the parameter bounds are assumed to be between -0.2892 and 0.2893 for each of the 4 upper level

variables x_1, x_2, x_3 and x_4 . Table 5 shows the results which clearly agrees with that reported in [49]. In this table UPO refers to the objective of the upper level in Eq. 21. Note that the standard deviation of the UPO over the 30 independent DEBLP runs conducted was less than 1×10^{-5} .

5.2.2 Example 2: “Linear Fit”

The model we intend to estimate is a linear equation of the form in Eq. 22. The 10 data points are from [49]. Compared to Example 1, here we assume that measurement errors are present in *both* $y_{m,1}$ and $y_{m,2}$, $m = \{1, \dots, 10\}$.

$$\hat{y}_{m,2} = x_1 + x_2 \hat{y}_{m,1} \quad (22)$$

Assuming a common variance for each data tuple $\{y_{m,1}, y_{m,2}\}$, we can estimate the vector of unknown x parameters via the BLPP in Eq. 23.

$$\begin{aligned} & \min_{\hat{x}} \sum_{m=1}^{10} \sum_{i=1}^2 (\hat{y}_{m,i} - y_{m,i})^2 \\ & \text{subject to} \\ & \min_{\hat{y}} \sum_{m=1}^{10} \sum_{i=1}^2 (\hat{y}_{m,i} - y_{m,i})^2 \\ & \hat{y}_{m,2} - x_1 - x_2 \hat{y}_{m,1} = 0 \end{aligned} \quad (23)$$

The results of 30 runs of DEBLP (with a maximum of 100 iterations allowed per run and a population size π of 20) for this problem are shown in Table 6. Again the standard deviation over the 30 runs was less than 1×10^{-5} . As with Example 1, the results obtained by DEBLP agrees with those reported in [49].

Table 5 Parameter Estimation Example 1 (“Kowalik Problem”)

Variable	DEBLP	[49]
x_1	0.1928	0.1928
x_2	0.1909	0.1909
x_3	0.1231	0.1231
x_4	0.1358	0.1358
UPO	0.000307	0.000307

Table 6 Parameter Estimation Example 2 (“Linear Fit”)

Variable	DEBLP	[49]
x_1	5.7840	5.784
x_2	-0.544556	-0.54556
UPO	0.61857	0.61857

6 Handling Upper Level Constraints

The keen reader would notice that up to this point our discussions and our numerical examples have neglected mention of constraints in the upper level problems (cf. Eqn. 1a). We have in fact thus far only assumed the presence of bound constraints

and described a technique to ensure that the population remains within the search domain which was sufficient for the problem examples investigated. Before proceeding to our next application area for BLPPs, we outline in this section, necessary modifications to DEBLP to enable it to handle them effectively.

6.1 Overview of Constraint Handling Techniques with Meta-Heuristics

In their most basic form, meta-heuristics do not have the capability to handle general constraints aside from bound constraints. However since real world problems generally have linear and nonlinear constraints, a large amount of research effort has been expended on the topic of constraint handling with such algorithms. In the past few years many techniques have been proposed. Among others these include penalty methods [121], adaptive techniques [104], techniques based on multiobjective optimisation [25, 65] etc.

The penalty method transforms the constrained problem into an unconstrained one. However one of the drawbacks of this method when applied with meta-heuristics is that the solution quality is sensitive to the penalty parameter used. The penalty parameter itself is problem dependent [99]. This method also encounters difficulties when solutions lie at the boundary of the feasible and infeasible space.

Recall the selection criteria of the DEBLP in Algorithm 1. In the presence of constraints, when we are deciding whether to accept or reject the child, \mathbf{w}_k , it is no longer a case of comparing the values of objective U attained. The key consideration is how one would say, decide between a infeasible individual with low U and a feasible individual but higher U .

Intuitively one could conclude that a feasible individual is better than the infeasible individual because the aim is to ultimately seek solutions that minimise the objective function and satisfy all the constraints. This viewpoint however ignores the fact that the meta-heuristics are generally stochastic by design. There exists the possibility that the infeasible individuals could in fact be better than the feasible one at some iterations during the algorithm [124]. The question then is how to strike the right balance between objective and constraints.

6.2 Stochastic Ranking

Runarsson and Yao [99, 100] proposed an alternative constraint handling method known as stochastic ranking (SR)⁷ to aid in answering this question. To use SR, the first step is to obtain a measure of the constraint violation, $v(\mathbf{x}_k)$, of vector \mathbf{x}_k using

⁷ The source code of SR is available at <http://notendur.hi.is/tpr/index.php?page=software/sres/sres>, accessed Oct 2011.

Eqn. 24. The first term on the RHS of Eq. 24 sums the maximum of either 0 or the value of the inequality constraint $G_j(\mathbf{x}_k), j \in \{1, \dots, q_1\}$ ⁸. The second term sums the absolute value of each of the equality constraints $E_j(\mathbf{x}_k), j \in \{1, \dots, r_1\}$.

$$v(\mathbf{x}_k) = \sum_{j=1}^{q_1} \max\{0, G_j(\mathbf{x}_k)\} + \sum_{j=1}^{r_1} |E_j(\mathbf{x}_k)| \quad (24)$$

The key operation of SR involves counting how many comparisons of adjacent pairs of solutions are dominated by the objective function and constraint violations. This is accomplished in SR through a stochastic bubble sort like procedure that is used to rank⁹ the population. This comparison is illustrated in Algorithm 2 where $rand(0, 1)$ is a pseudo random real number between 0 and 1. The method requires a probability factor η which should be less than 0.5 to create a bias against infeasible solutions [99].

Suppose we have two individuals \mathbf{x}_{k1} and $\mathbf{x}_{k2}, k1 \neq k2$. If both do not violate constraints *or* if a pseudo random real number is less than or equal to η , we swap their rank order based on the objective function obtained, with the lower one being assigned a higher rank. Otherwise we swap their ranks based on the constraint violations, again with the lower constraint violation being assigned a higher rank. Working our way through the population to be ranked, we continue comparing adjacent members according to Algorithm 2 and swapping ranks. When no change in rank order occurs, SR terminates.

Algorithm 2 Stochastic Ranking

```

if  $v(\mathbf{x}_{k1}) = 0$  and  $v(\mathbf{x}_{k2}) = 0$  or  $rand(0, 1) \leq \eta$  then
  rank based on objective function value only
else
  rank based on constraint violation only
end if

```

6.3 Revised DEBLP with Stochastic Ranking

DEBLP-SR, as presented in Algorithm 3, is the result of incorporating SR in DEBLP. *Italics* highlight the changes between DEBLP in Algorithm 1 and DEBLP-SR in Algorithm 3. These are summarised as follows:

1. Evaluation of both the upper level objective and constraint violation for each member of the parent and child population.

⁸ In Eq. 1a, all the upper level inequality constraints are in the form “ ≤ 0 ”.

⁹ With π population members, ranking results in the best ranked 1 (highest rank) and the worst ranked π (lowest rank).

2. Instead of the one to one selection criteria discussed in Section 3, we propose to pool the parent and child population (along with the corresponding objective values and constraint violations) together as an input into SR.
3. Combining parents and children will lead to a population size of 2π . Hence the selection process will only retain the top π ranked individuals output by SR to constitute the population at the next iteration. The remainder are discarded.

Algorithm 3 DEBLP with Stochastic Ranking (DEBLP-SR)

1. Randomly generate parent population \mathcal{P} of π individuals.
 2. Evaluate \mathcal{P} and obtain constraint violations using Eq. 24
set iteration counter $it = 1$
 3. While stopping criterion not met, do:
 - For each individual in \mathcal{P}^{it} , do:
 - a) Apply Mutation and Crossover to create a single child from individual.
 - b) Evaluate child and obtain constraint violations using Eq. 24
 - End For
 4. Combine parents and children violations and objectives.
 5. Apply stochastic ranking
 6. Selection: retain the top π ranked individuals to form new population \mathcal{P}^{it+1}
 $it = it + 1$
End While
-

In the next section, we apply DEBLP-SR to a examples of BLPPs that are in fact characterised by the presence of upper level constraints. It will be shown that DEBLP-SR continues to be a robust meta-heuristic in such applications.

7 Applications to Generalised Nash Equilibrium Problems

Game theory [116] is a branch of social science that provides methodologies to study behaviour when rational agents seek to maximise personal gains in the presence of others symmetrically doing the same simultaneously. The solution concept of such games was devised by Nash in [83, 84]. The game attains a Nash Equilibrium (NE) if no one player can unilaterally improve her payoff given the strategic decisions of all other players. While establishing that an outcome is not a NE (by establishing that a player can profitably deviate) is usually not difficult, locating the NE itself is more challenging. In this section we show how the process of determining NE in some games can be formulated as a BLPP and illustrate the performance of DEBLP on some example problems from the literature.

7.1 The Generalised Nash Equilibrium Problem

We are concerned with a specific Nash Game known as the Generalised Nash Equilibrium Problem (GNEP). In the GNEP, the players' payoffs and their strategies are continuous (and subsets of the real line) but most critically the GNEP embodies the distinctive feature that players face constraints depending on the strategies their opponents choose. This distinctive feature is in contrast to a standard Nash Equilibrium Problem (NEP) where the utility/payoff/reward the players obtain depend solely on the decisions they make and their actions are not restricted as a result of the strategies chosen by others. The ensuing constrained action space in GNEPs makes them more difficult to resolve than standard NEPs discussed in monographs such as [116]. As will be demonstrated in this section, the technique here can nevertheless be applied to standard NEPs.

The GNEP under consideration is a single shot¹⁰ game with a set Γ of players indexed by $i \in \{1, 2, \dots, \rho\}$ and each player can play a strategy $x_i \in X_i$ which all players are assumed to announce simultaneously. X is the collective action space for all players. In a standard NEP, $X = \prod_{i=1}^{\rho} X_i$, i.e. X is the Cartesian product.

In contrast, in a GNEP, the feasible strategies for player $i, i \in \Gamma$ depend on the strategies of all other players [4, 39, 53, 112]. We denote the feasible strategy space of each player by the point to set mapping: $\mathcal{C}^i : X^{-i} \rightarrow X^i, i \in \Gamma$ that emphasises the ability of other players to influence the strategies available to player i [39, 51, 112]. The distinction between a conventional Nash game and a GNEP can be viewed as analogous to the distinction between unconstrained and constrained optimisation.

To give stress to the variables chosen by player i , we sometimes write $\mathbf{x} = (x_i, x_{-i})$ where x_{-i} is the combined strategies of all players in the game *excluding* that of player i i.e. $x_{-i} = (x_1, \dots, x_{(i-1)}, x_{(i+1)}, \dots, x_\rho)$. Note that the notation (x_i, x_{-i}) *does not* mean that the components of \mathbf{x} are somehow reordered such that x_i becomes the first block. In addition, let $\phi_i(\mathbf{x})$ be the payoff/reward to player $i, i \in \Gamma$ if \mathbf{x} is played.

Definition 2 [112] *A combined strategy profile $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_\rho^*) \in X$ is a Generalised Nash Equilibrium for the game if:*

$$\begin{aligned} \phi_i(x_i^*, x_{-i}^*) &\geq \phi_i(x_i, x_{-i}^*), \\ \forall x_i &\in \mathcal{C}(x_{-i}^*), i \in \Gamma \end{aligned} \tag{25}$$

At a Nash Equilibrium no player can benefit (increase individual payoffs) by unilaterally deviating from her current chosen strategy. Players are also assumed not to cooperate and in this situation each is doing the best she can given what her competitors are doing [45, 62, 116]. For a GNEP, the strategy profile \mathbf{x}^* is a Generalised Nash Equilibrium (GNE) if it is *both* feasible with respect to the mapping \mathcal{C}^i and if it is a maximizer of each player's utility over the constrained feasible set [51].

¹⁰ It is one-off and not played repeatedly in a dynamic sense.

7.2 Nikaido Isoda Function

The Nikaido Isoda (NI) function in Eq. 26 is an important construct much used in the study of Nash Equilibrium problems [39, 52, 53]. Its interpretation is that each summand shows the increase in payoff a player will receive by unilaterally deviating and playing a strategy $y_i \in \mathcal{C}(x_{-i})$ while all other players play according to x .

$$\Psi(\mathbf{x}, \mathbf{y}) = \sum_1^{\rho} [\phi_i(y_i, x_{-i}) - \phi_i(x_i, x_{-i})] \quad (26)$$

The NI function is always non-negative for any combination of \mathbf{x} and \mathbf{y} . Furthermore, this function is everywhere non-positive when either \mathbf{x} or \mathbf{y} is a NE by virtue of Definition 2 since at a NE no one player should be able to increase their payoff by unilaterally deviating. This result is summarised in Definition 3.

Definition 3 [53] *A vector $\mathbf{x}^* \in X$ is called a Generalised Nash Equilibrium (GNE) if $\Psi(\mathbf{x}, \mathbf{y}) = 0$.*

7.3 Solution of the GNEP

Proposition 2 establishes the key result that the GNEP can be formulated as a BLPP.

Proposition 2. *The Generalised Nash Equilibrium is the solution to the BLPP in Eq. 27.*

$$\min_{(\mathbf{x}, \mathbf{y})} f(\mathbf{x}, \mathbf{y}) = (\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) \quad (27a)$$

$$\text{subject to } x^i \in \mathcal{C}^i(x^{-i}), \forall i \in \Gamma. \quad (27b)$$

where \mathbf{y} solves

$$\max_{(\mathbf{x}, \mathbf{y})} (\phi_1(y^1, x^{-1}) + \dots + \phi_{\rho}(y^{\rho}, x^{-\rho})) = \quad (28a)$$

$$\max_{(\mathbf{x}, \mathbf{y})} \sum_{i=1}^n [\phi_i(y_i, x_{-i}) - \phi_i(x_i, x_{-i})]$$

$$\text{subject to } y^i \in \mathcal{C}^i(x^{-i}), \forall i \in \Gamma. \quad (28b)$$

Proof. For a proof of Proposition 2, see [112]. \square

The upper level problem (Eq. 27a) is a norm minimization problem subject to strategic variable constraints (Eq. 27b). The objective function of the lower level problem (Eq. 28) is exactly the Nikaido Isoda function (Eq. 26).

Proposition 3. *The optimal value of the upper level objective in Eq. 27a, $f(\mathbf{x}, \mathbf{y})$, is 0 at the Generalised Nash Equilibrium.*

Proof. For a proof of Proposition 3, see [14, 112]. \square

Proposition 3 serves the critical role of being the termination criteria of the DEBLP. Although DEBLP and DEBLP-SR are heuristic in nature, Proposition 3 enables us to detect that we have found the solution to the GNEP.

7.4 Examples

In this section, we present four numerical examples of GNEPs sourced from the literature. The first case study is in fact a standard NEP and it serves to demonstrate that the BLPP formulation proposed here can also be applied in this situation. We then impose a constraint which transforms the standard NEP into a GNEP which serves as the second example. The third example has origins in pollution abatement modeling while the last example is an internet switching model from [58].

7.4.1 Example 1

Example 1 is a non-linear Cournot-Nash Game with 5 players from [81]. As mentioned, this is a standard NEP i.e. where the feasible strategies of each player is unconstrained. The profit function for player $i, i \in \{1, \dots, 5\}$, comprising the difference between revenues and production costs, is given by: $\phi_i(\mathbf{x}) = (5000)^{\frac{1}{11}} \left(\sum_{i=1}^5 x_i \right)^{-\left(\frac{1}{11}\right)} x_i - \omega_i x_i + \left(\frac{\alpha_i}{\alpha_i + 1} \right) \gamma_i^{\frac{-1}{\alpha_i}} x_i^{\frac{\alpha_i + 1}{\alpha_i}}$. The player dependent parameters (ω_i , γ_i and α_i) are found in [81, 87].

The feasible space for this problem is the positive axis since production cannot be negative. The solution of the NEP is $\mathbf{x}^* = [36.9318, 41.8175, 43.7060, 42.6588, 39.1786]^T$ [50, 81].

7.4.2 Example 2

Using the same parameters as in Example 1, and introducing a production constraint on total output of all players ¹¹ as in [87], Example 1 is transformed into a GNEP.

The feasible space for the resulting GNEP is defined by [87]:

$$X = \{ \mathbf{x} \in \mathbb{R}^5 | x_i \geq 0 \forall i \in \{1, \dots, 5\}, \sum_{i=1}^5 x_i \leq 100 \}$$

¹¹ One can think of this as simulating a cartel limiting production to keep prices high.

\mathbf{x}^* is $[14.050, 17.798, 20.907, 23.111, 24.133]^T$ [54].

7.4.3 Example 3

This problem describes an Environmental Pollution Control Problem known as the ‘‘River Basin Pollution Game’’ studied by Krawczyk and co-workers [52, 64]. There are 3 players with a single decision variable each. Each player $i, i \in \{1, 2, 3\}$ attempts to maximise his profits, while others are doing the same simultaneously. Player i ’s payoff function is given as $\phi_i(\mathbf{x}) = (3 - 0.01(\sum_{i=1}^3 x_i))x_i - (c_{1i} + c_{2i})x_i$. The first term in the payoff function is the revenues from the sale of the product. The second term is the production costs. The cost values c_{1i} and c_{2i} are given in [52, 64]. The feasible space reflecting mandatory limits on allowed effluent discharges into a river is defined according to:

$$\begin{aligned} 3.25x_1 + 1.25x_2 + 4.125x_3 &\leq 100 \\ 2.2915x_1 + 1.5625x_2 + 2.8125x_3 &\leq 100 \\ x_i &\geq 0, i \in \{1, 2, 3\} \end{aligned}$$

The last constraint reflects the fact that production cannot be negative. The GNE is $x_1^* = 21.14, x_2^* = 16.03, x_3^* = 2.927$ [52, 64, 54].

7.4.4 Example 4

This problem describes an internet switching model with 10 players originally proposed in [58] and also studied in [54]. The cost function for player $i, i \in \{1, \dots, 10\}$ is given by $\phi_i(\mathbf{x}) = -(\frac{x_i}{(x_1 + \dots + x_{10})})(1 - \frac{(x_1 + \dots + x_{10})}{1})$. The feasible space is $X = \{\mathbf{x} \in \mathbb{R}^{10} | x_i \geq 0.01, i \in \{1, \dots, 10\}, \sum_{i=1}^{10} x_i \leq 1\}$. The NNE is $x_i^* = 0.09, i = \{1, \dots, 10\}$ [53].

7.5 Discussion

As highlighted earlier, Proposition 3 states that when the upper level objective (UPO) (cf. Eqn. 27a), $f(x, y)$, in Program U reaches 0, we have successfully solved the GNEP. Hence this allows us to provide a termination criteria of the DEBLP-SR algorithm. In all other examples, we have always stopped the DEBLP after a user specified number of maximum iterations. In practice, we terminate each run when the UPO attains the value of 1×10^{-8} or less, which we judge to be sufficiently close to 0.

In all these examples, we used DEBLP-SR i.e. Algorithm 3 with a population size, $\pi = 50$ and allowed a maximum of 250 iterations. Following [99, 100], the

probability factor, η used in SR was set to 0.45. Table 7 reports the mean, median and standard deviations (SD) of the number of function evaluations (NFE) over the 30 independent runs of DEBLP-SR to meet the convergence criteria (i.e. UPO attains the value of at least 1×10^{-8}).

Table 7 Summary of Performance of DEBLP-SR on GNEP Examples

Example	1	2	3	4
mean NFE	3993	7378	2825	10098
median NFE	4075	7325	2850	10025
SD	410	2758	273	1659
Constraint Violation	NA	0	0	0

While it is clear that all the examples are easily solved using DEBLP-SR, three observations are pertinent from Table 7. Firstly, comparing Problem 3 and 4 for example, we can see that as the dimensions increase, the NFE required to meet the convergence criteria also increase significantly. This is a manifestation of the so called ‘‘curse of dimensionality’’ [10] which plagues optimisation algorithms in general and meta-heuristics in particular. Secondly the mean and median NFE required to solve the GNEP (Example 2) is almost twice that required to solve the NEP (Example 1). This should not come as a surprise because constrained problems are known [121] to be harder to solve than unconstrained ones. Finally, the constraint violation of all examples at termination is 0 as shown in the last row of Table 7. Thus we can conclude that the SR method for handling constraints is effective for the examples given.

8 Summary and Conclusions

8.1 Summary

In this chapter, we have outlined a meta-heuristic algorithm DEBLP to solve bi-level programming problems. These hierarchical optimisation problems are typically characterised by non convexity and non smoothness. DEBLP is designed to synergise the well-documented global search capability of Differential Evolution with the application of robust deterministic optimisation techniques to the lower level problem. Most importantly, DEBLP is fully consistent with the Stackleberg framework upon which the BLPP is founded where the leader takes into account the follower’s decision variables when optimizing his objective and where the follower treats the leader’s variables as exogenous when solving his problem.

DEBLP was subsequently demonstrated on a number of BLPPs arising from several disciplines. These include control problems in Transportation Systems where we studied the Continuous Optimal Toll Pricing and the Continuous Network Design Problem. In these situations we postulated that the leader/upper level player

was the regulatory agency and the followers were users of the highway network. The BLPPs from this field were shown to be MPECs as the lower level problem arises naturally as a Variational Inequality. We also examined examples from Parameter Estimation Problems, a key step in the development of models in science and engineering applications, which could also be formulated as BLPPs. In order to enable DEBLP to solve BLPPs where the upper level problem was also subject to general constraints, we integrated the stochastic ranking algorithm from [99] into DEBLP to produce DEBLP-SR. Stochastic ranking is a constraint handling technique that seeks to balance the dominance of the objective and constraint violations in the search process of meta-heuristic algorithms. We demonstrated the operation of DEBLP-SR on a series of Generalised Nash Equilibrium Problems which could be formulated as BLPPs characterised by upper level constraints. Developments in the literature of GNEPs also enabled us to even specify a specific termination criteria for the proposed BLPP and hence provides additional justification for the use of a meta-heuristic for these problems.

Due to space constraints, we could not illustrate BLPPs where the leader's decision variables and/or the follower's variables were restricted to be discrete or binary. However there exists a large body of literature of DE being used for such problems, albeit single level ones [91, 92]. Thus we conjecture the techniques proposed therein could be integrated into DEBLP to solve such problems as well. Additionally discrete and mixed integer lower level problems can already be solved using established techniques available in the deterministic optimisation literature [9, 70, 85].

8.2 Further Research

In this chapter, we have demonstrated that DEBLP is an effective meta-heuristic for a variety of BLPPs. Nevertheless there are several topics that still require additional research before robust methodologies can be developed. The study of some of these problems is still in its infancy but we argue that meta-heuristic paradigms such as Differential Evolution can provide a viable alternative solution framework for these.

8.2.1 Multiple Optimisation Problems at Lower Level

The BLPP we have formulated assumes the existence of a single optimisation problem at the lower level. Both DEBLP and DEBLP-SR are unable to handle the situation of *multiple followers* i.e. presence of multiple optimisation problems at the lower level. See e.g. [69, 114] for examples of these. However we have neglected consideration of such problems in this paper but should be the subject of further research.

8.2.2 Bi-Level Multiobjective Problems

Recall that in our formulation of the BLPP in Eq. 1 we assumed the function mappings: $U, L : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^1$. In other words, the objectives in both the upper and lower levels are restricted to be scalar. However there are also problems where the objectives are vectors. Such problems are known as multiobjective (MO) problems i.e. where the decision maker has multiple, usually conflicting, objectives. In such problems the Pareto Optimality criteria is used to identify optimum solutions [30, 90]. One of the major advantages of using population based meta-heuristic algorithms for MO Problems is that because of their population based structure, they are able to identify multiple Pareto Optimal solutions in a single run [29].

Two categories of these problems have been discussed in the literature. Firstly there is the case where only the upper level objective is vector based or secondly where *both* the upper and lower level objectives are vector based. For problems occurring in the first category, advances in meta-heuristics to solve MO problems (e.g. [30]) could be easily integrated into DEBLP to transform it into an algorithm able to handle MO-BLPPs of the type described in e.g. [44, 110, 123]. Problems of the second category are relatively novel in the literature and have only recently been investigated [31]. Further research should introduce new methodologies to enable DEBLP to solve problems in this latter category.

8.2.3 Multiple Leader Follower Games

In Section 4 we provided an example of the COTP which models a highway regulatory agency optimising the total travel time on the highway system by levying toll charges. With the trend in recent years towards privatization together with constrained governmental budgets, it is quite possible that instead of a welfare maximizing authority setting the tolls in future, this task could potentially be consigned to private profit maximising entities. The latter obtain concessions to collect tolls from users on these private toll roads [36, 119] in return for providing the capital layout of investments in new road infrastructure. When setting such tolls, these private firms could also be in competition with others doing the same on other roads in the network.

The problem just described is in fact an example of a class of Equilibrium problems with Equilibrium Constraints (EPEC). In EPECs, the decision variables of the private firms are constrained by a variational inequality describing equilibrium in some parametric system [62]. For example in the case of competition between the private toll road operators just highlighted, the equilibrium constraint is just Wardrop's User Equilibrium condition. The study of EPECs has recently been given greater emphasis by researchers in many disciplines [55, 68, 77, 80, 119, 125]. Though it is still in a period of infancy it has emerged as major area of research [22, 35, 109] in applied mathematics.

Formally an EPEC is a mathematical program to find an equilibrium point that simultaneously solves a set of MPECs where each MPEC is parameterised by de-

cision variables of other MPECs [125]. Compared to the MPEC, the focus in the EPEC is shifted away from finding minimum points to finding *equilibrium* points [78, 79]. Figure 4 gives a multi-leader generalization of the BLPP that constitutes a Multi-leader-follower game [68] where there are now $\rho, \rho > 1$ leaders instead.

In this multi-leader generalization of the Stackelberg game researchers have conjectured that there could be two possible behaviours of the leaders at the upper level [78, 88]. At one end, leaders could cooperate which results in a multiobjective problem subject to an equilibrium constraint at the lower level [120]. At the other end, the leaders could act non-cooperatively and play a Nash game amongst themselves resulting in a Non Cooperative EPEC (NCEPEC). EPECs are extremely difficult to solve and the current emphasis has been on the use of nonsmooth methods and nondifferentiable optimisation techniques [78, 79]. We believe that meta-heuristic algorithms offer a powerful alternative solution methodology for EPECs in both cases. In the case when leaders are assumed to cooperate, we have pointed out that because they operate with populations, population based meta-heuristics are able to identify multiple Pareto Optimal solutions in a single simulation run. This is key to solving multiobjective problems. For the NCEPECs, a DE based algorithm exploiting a concept from [71] was proposed and demonstrated on a range of EPECs occurring in transportation and electricity markets in [62].

Most importantly, whatever solution algorithms are proposed in future, when searching for an equilibrium amongst the players at the upper level they must continue to take the reaction of the followers at the lower level into account. This serves to ensure that proposals are entirely consistent with the Stackelberg paradigm which remains applicable in EPECs.

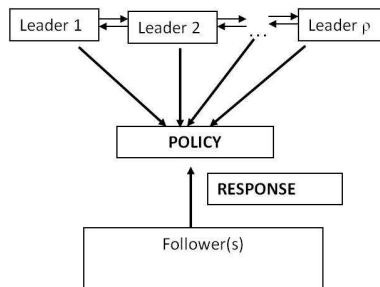


Fig. 4 Pictorial Representation of an EPEC

Acknowledgements The research reported here is funded by the Engineering and Physical Sciences Research Council of the UK under Grant EP/H021345/1 (July 2010 to July 2013).

References

1. Aarts E, Korst J (1988) Simulated Annealing and Boltzmann Machines. John Wiley,Chichester
2. Aiyoshi, E, Shimizu K (1981) Hierarchical decentralized systems and its new solution by a barrier method. IEEE Transactions on Systems, Man, and Cybernetics 11(6):444-449
3. Allsop RE (1974) Some possibilities for using traffic control to influence trip distribution and route choice. In: Buckley DJ (ed) Transportation and Traffic Theory: Proceedings of the Sixth International Symposium on Transportation and Traffic Theory. Elsevier,New York:345-375
4. Aussel D, Dutta J (2008) Generalized Nash equilibrium problem, variational inequality and quasiconvexity. Operations Research Letters 36(4):461-464
5. Bard JF (1983) An efficient point algorithm for a linear two-stage optimization problem. Operations Research 31(4):670-684
6. Bard, JF(1988) Convex two-level optimization. Mathematical Programming 40(1):15-27
7. Bard JF (1998) Practical Bilevel Optimisation: Algorithms and Applications. Kluwer,Dordrecht
8. Bard JF, Plummer J, Sourie JC (2000) A bilevel programming approach to determining tax credits for biofuel production. European Journal of Operational Research 120(1):30-46
9. Bazaraa MS, Sherali HD, Shetty, CM (2006) Nonlinear Programming: Theory and Algorithms (3rd Edition) Wiley,Hoboken,New Jersey
10. Bellman RE (1961) Adaptive control processes: a guided tour.Princeton University Press,Princeton New Jersey
11. Ben-Ayed O, Blair CE (1990) Computational difficulties of bilevel linear programming. Operations Research 38(3):556-560
12. Ben-Ayed O (1993) Bilevel linear programming. Computers & Operations Research 20(5):485-501
13. Bialas WF, Karwan MH (1984) Two-Level Linear Programming. Management Science 30(8):1004-1020
14. Bouza Allende GB (2008) On the calculation of Nash Equilibrium points with the aid of the smoothing approach. Revista Investigación Operacional 29(1):71-76
15. Beckmann M, McGuire C, Winsten C (1956) Studies in the Economics of Transportation. Yale University Press,New Haven,Connecticut
16. Britt H, Luecke R (1973) The Estimation of Parameters in Nonlinear Implicit Models. Technometrics 15(2): 233-247
17. Braess D (1968) Über ein paradoxon aus der verkehrsplanung. Unternehmensforschung 12:258-268
18. Bracken J, McGill J (1973) Mathematical programs with optimization problems in the constraints. Operations Research 21(1):37-44
19. Bracken J, McGill J (1978) Defense applications of mathematical programs with optimization problems in the constraints. Operations Research 22(5):1086-1096
20. Candler W, Norton R (1977) Multi-level programming. Technical Report DRD-20, World Bank, January
21. Candler W, Townsley R (1982) A linear two-level programming problem. Computers & Operations Research 9(1):59-76
22. Červinka M: Hierarchical structures in equilibrium problems. PhD Thesis, Charles University, Prague, Czech Republic (2008)
23. Chen Y, Florian M (1995) The nonlinear bilevel programming problem: Formulations, regularity and optimality conditions. Optimization 32(3):193-209
24. Chiou S (2005) Bilevel programming formulation for the continuous network design problem. Transportation Research 39B(4):361-383
25. Coello-Coello, CA (1999) Treating constraints as objectives for single-objective evolutionary optimization. Engineering Optimization 32(13):275-308
26. Colson B, Marcotte P, Savard G (2007) An overview of bilevel optimization. Annals of Operations Research 153(1):235-256

27. Cree ND, Maher MJ, Paechter B (1996) The continuous equilibrium optimal network design problem: a genetic approach. In: Bell MGH (ed) *Transportation Networks: Recent Methodological Advances*, Pergamon, London: 163–174
28. Dafermos SC (1980) Traffic equilibrium and variational inequalities. *Transportation Science* 14(1):42-54
29. Deb K (1999) Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation* 7(3):205–230
30. Deb K(2001) *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley, Chichester
31. Deb K, Sinha A (2010) An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary Computation* 18(3):403-449
32. Dempe S (2002) *Foundations Of Bilevel Programming*. Kluwer,Dordrecht
33. Dempe S (2003) Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization* 52(3):333-359
34. Dorigo M. Stützle T (2004), *Ant Colony Optimization*, MIT Press,Cambridge,Massachusetts
35. Ehrenmann A (2004) *Equilibrium Problems with Equilibrium Constraints and their Application to Electricity Markets*. PhD Thesis, Fitzwilliam College, University of Cambridge, UK
36. Engel E, Fischer R, Galetovic A(2002) A new approach to private roads. *Regulation* 25(3):18-22
37. Esposito WR, Floudas CA, (1998) Global Optimization in Parameter Estimation of Nonlinear Algebraic Models via the Error-in-Variables Approach. *Industrial Engineering and Chemistry Research* 37(5):1841–1858
38. Facchinei F, Jiang H, Qi L (1999) A smoothing method for mathematical programs with equilibrium constraints. *Mathematical Programming* 85B(1):107–134
39. Facchinei F, Kanzow C (2010) Generalized Nash equilibrium problems. *Annals of Operations Research* 175(1):177–211
40. Fisk CS (1984) Game theory and transportation systems modelling. *Transportation Research* 18B(4-5):301–313
41. Fortuny-Amat J, McCarl B (1981) A representation and economic interpretation of a two-Level programming problem. *Journal of the Operational Research Society* 32(9):783–792
42. Friesz T, Harker PT(1985) Properties of the iterative optimization-equilibrium algorithm, *Civil Engineering Systems* 2(3):142–154
43. Friesz T, Cho H, Mehta N, Tobin R, Anandalingam G (1992) A simulated annealing approach to the network design problem with variational inequality Constraints. *Transportation Science* 26(1):18–26
44. Friesz T, Anandalingam G, Mehta NJ, Nam K, Shah, S, Tobin R (1993) The multiobjective equilibrium network design problem revisited: A simulated annealing approach, *European Journal of Operational Research* 65(1): 44-57
45. Gabay D, Moulin H (1980) On the uniqueness and stability of Nash-equilibria in non cooperative games. In: Bensoussan A, Dorfer PK, Tapiero C(eds) *Applied Stochastic Control in Econometrics and Management Science*. North Holland, Amsterdam: 271–293
46. Gau CY, Stadtherr MA (2002) Deterministic global optimization for error-in-variables parameter estimation. *AIChE Journal* 48(6):1192–1197
47. Goldberg DE (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley,Reading, Massachusetts
48. Glover F, Laguna M (1997) *Tabu Search*. Kluwer,Norwell, Massachusetts
49. Gümüş ZH, Floudas CA (2001) Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization* 20(1): 1–31
50. Harker PT (1984) A variational inequality approach for the determination of oligopolistic market equilibrium. *Mathematical Programming* 30(1):105–111
51. Harker PT (1991) Generalized Nash games and quasi-variational inequalities. *European Journal of Operations Research* 54(1):81–94
52. Haurie A, Krawczyk J (1997) Optimal charges on river effluent from lumped and distributed sources. *Environment Modelling and Assessment* 2(3):177-189

53. von Heusinger A, Kanzow C (2009) Relaxation methods for generalized Nash equilibrium problems with inexact line search. *Journal of Optimization Theory and Applications* 143(1):159–183
54. von Heusinger A (2009) Numerical Methods for the Solution of the Generalized Nash Equilibrium Problem. PhD Thesis, Institute of Mathematics, University of Würzburg, Würzburg
55. Hu X, Ralph D (2007) Using EPECs to model bilevel games in restructured electricity markets with locational prices. *Operations Research* 55(5):809–827
56. Karamardian S (1971), Generalized complementarity problems. *Journal of Optimization Theory and Applications* 8(3):161–168
57. Kennedy J, Eberhart RC (1995) Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*:1942–1948
58. Kesselman A, Leonardi S, Bonifaci V (2005) Game-Theoretic Analysis of Internet Switching with Selfish Users. *Internet and Network Economics*, LNCS 3828, Springer, Berlin:236–245
59. Kim IW, Liebman MJ, Edgar TF (1990) Robust nonlinear error-in-variables estimation method using nonlinear programming techniques. *AIChE Journal* 36(7):985–993
60. Koh A, Watling D (2010) Traffic Assignment Modelling. In: Button K, Vega H, Nijkamp P (eds) *A Dictionary of Transport Analysis*, Edward Elgar, Cheltenham:418–420
61. Koh A (2007) Solving transportation bi-level programs with Differential Evolution. *Proceedings of the IEEE Congress on Evolutionary Computation*:2243–2250
62. Koh A (2011) An evolutionary algorithm based on Nash dominance for equilibrium problems with equilibrium constraints. *Applied Soft Computing* <http://dx.doi.org/10.1016/j.asoc.2011.08.056> (in press)
63. Korilis Y, Lazar A, Orda A (1999) Avoiding the Braess paradox in non-cooperative networks. *Journal of Applied Probability* 36(1): 211–222
64. Krawczyk J, Uryasev S (2000) Relaxation algorithms to find Nash equilibria with economic applications. *Environmental Modelling and Assessment* 5(1):63–73
65. Lampinen J (2002) A constraint handling approach for the differential evolution algorithm. *Proceedings of the IEEE Congress on Evolutionary Computation*:1468–1473
66. Lawphongpanich S, Hearn DW (2004) An MPEC approach to second-best toll pricing. *Mathematical Programming* 101B(1):33–55
67. Leyffer S (2005) The penalty interior-point method fails to converge. *Optimization Methods and Software* 20(4-5):559–568
68. Leyffer S, Munson T (2010) Solving multi-leader-common-follower games. *Optimization Methods and Software* 25(4):601–623
69. Liu B (1998) Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms, *Computers & Mathematics with Applications* 36(7):79–89
70. Luenberger DG, Ye Y (2008) *Linear and Nonlinear Programming*. Springer, Berlin
71. Lung R I, Dumitrescu D (2008) Computing Nash equilibria by means of evolutionary computation. *International Journal of Computers, Communications and Control* III:364–368
72. Luo ZQ, Pang JS, Ralph D (1996) *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge
73. Matthieu R, Pittard L, Anandalingam G (1994) Genetic algorithm based approach bi-level linear programming. *RAIRO-Operations Research* 28(1):1–21.
74. Marcotte P, Savard G (2009) Bilevel programming: applications. In: Floudas CA, Pardalos PM (ed) *Encyclopedia of Optimization*. (Second Edition), Springer, New York:241–242
75. Meng Q, Yang H, Bell MGH (2000) An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transportation Research* 35B(1) 83–105, (2000)
76. Moore RE, Hansen E, Leclerc A (1992), Rigorous methods for global optimization. In: Floudas CA, Pardalos PM (eds) *Recent advances in global optimization*. Princeton University Press, Princeton, New Jersey:321–342
77. Mordukhovich BS, Outrata JV, Červinka M (2007) Equilibrium problems with complementarity constraints: Case study with applications to oligopolistic markets. *Optimization* 56(4):479–494

78. Mordukhovich BS (2005) Optimization and equilibrium problems with equilibrium constraints. *Omega* 33(5):379–384
79. Mordukhovich BS (2006) *Variational Analysis and Generalized Differentiation, I: Basic Theory*. Springer, Berlin
80. Mordukhovich BS (2006) *Variational Analysis and Generalized Differentiation, II: Applications*. Springer, Berlin
81. Murphy FH, Sherali HD, Soyster AL (1982) A mathematical programming approach for determining oligopolistic market equilibrium. *Mathematical Programming* 24(1): 92–106
82. Nagurney A (1999) *Network Economics: A Variational Inequality Approach*. Kluwer, Boston
83. Nash J (1950) Equilibrium points in N-person games. *Proceedings of the National Academy of Science of the USA* 36(1):48–49
84. Nash J (1951) Non-Cooperative games. *Annals of Mathematics Second Series* 54(2):286–295
85. Nocedal J, Wright SJ (1999) *Numerical Optimization*. Springer, New York
86. Oduguwa V, Roy R (2002) Bi-level optimisation using genetic algorithms. *Proceedings of the IEEE Conference on Artificial Intelligence Systems*:322–327
87. Outrata JV, Kočvara M, Zowe J (1998) *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*. Kluwer, Dordrecht
88. Outrata JV (2004) A note on a class of equilibrium problems with equilibrium constraints. *Kybernetika* 40(5):585–594
89. Ortuzar JD, Willumsen L (1998) *Modelling Transport*. Wiley, Chichester
90. Osyczka A (1998) *Multicriterion optimization in engineering with FORTRAN programs*. Ellis Horwood, Wiley, Chichester
91. Pampara G, Engelbrecht AP, Franken N (2006) Binary differential evolution. *Proceedings of the IEEE Congress on Evolutionary Computation*:1873–1879
92. Pan Q, Tasgetiren MF, Liang YC (2008) A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Computers & Industrial Engineering* 55(4):795–816
93. Poorzahedy H, Abulghasemi F (2005) Application of ant system to network design problem. *Transportation* 32(3):251–273
94. Price K (1999), An Introduction to Differential Evolution In Corne D, Dorigo M, Glover F (eds) *New Techniques in Optimization*. McGraw Hill, London:79–108
95. Price K, Storn R, Lampinen J (2005) *Differential evolution: a practical approach to global optimization*. Springer, Berlin
96. Rajesh J, Gupta K, Kusumakar HS, Jayaraman VK, Kulkarni B (2003) A tabu search based approach for solving a class of bilevel programming problems in chemical engineering. *Journal of Heuristics* 9(4):307–319
97. Rosen JB (1965) Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica* 33(3):520–534
98. Rod V, Hančil V (1980) Iterative estimation of model parameters when measurements of all variables are subject to error. *Computers & Chemical Engineering* 4(2):33–38
99. Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4(3):284–294
100. Runarsson TP, Yao X (2005) Search biases in constrained evolutionary optimization. *IEEE Transactions on Systems, Man and Cybernetics Part C* 35(2):233–243
101. Sahin KH, Ciric AR (1998) A dual temperature simulated annealing approach for solving bilevel programming problems. *Computers & Chemical Engineering* 23(1): 11–25
102. Scheel H, Scholtes S (2000) *Mathematical Programs with Complementarity Constraints: Stationarity, Optimality, and Sensitivity*. *Mathematics of Operations Research* 25(1):1–22
103. Sheffi Y (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice Hall, Englewood Cliffs, New Jersey
104. Smith A, Tate D (1993) Genetic optimization using a penalty function, *Proceedings of the 5th International Conference on Genetic Algorithms*:499–503
105. Smith MJ (1979) The existence, uniqueness and stability of traffic equilibria. *Transportation Research* 13B(4):295–304

106. von Stackelberg, HH (1952) The theory of the market economy. William Hodge, London
107. Steenbrink PA (1974) Optimization of Transportation Networks. John Wiley, London
108. Storn R, Price K (1997) Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4):341–359
109. Su C (2005) Equilibrium problems with equilibrium constraints: stationarities, algorithms and applications. PhD Thesis, Stanford University, California, USA
110. Sumalee A, May A, Shepherd S (2009) Road user charging design: dealing with multi-objectives and constraints. *Transportation* 36(2): 167–186
111. Sumalee A (2004) Optimal Road Pricing Scheme Design. PhD Thesis, Institute for Transport Studies, University of Leeds, UK
112. Sun L (2010) Equivalent Bilevel Programming Form for the Generalized Nash Equilibrium Problem. *Journal of Mathematics Research* 2(1), 8–13
113. Tjoa T, Biegler L (1992) A reduced successive quadratic programming strategy for errors-in-variables estimation. *Computers & Chemical Engineering* 16(6):523–533
114. Wang Y, Jiao Y, Li H (2005) 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.
115. Wardrop JG (1952) Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers Part II* 1(36): 325–378
116. Webb JN (2007) Game theory: Decisions, Interaction and Evolution. Springer, London
117. Wen U, Hsu S (1991) Linear bi-level programming problems – a review. *Journal of the Operational Research Society* 42(2):125–133
118. Yang H, Lam HK (1996) Optimal road tolls under conditions of queuing and congestion. *Transportation Research* 30A(5):319–332
119. Yang H, Feng X, Huang H (2009) Private road competition and equilibrium with traffic equilibrium constraints. *Journal of Advanced Transportation* 43(1): 21–45
120. Ye J J, Zhu Q J (2003) Multiobjective optimization problem with variational inequality constraints. *Mathematical Programming* 96A(1): 139–160
121. Yeniy Ö (2005) Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications* 10(1):45–56
122. Yin Y (2000) Genetic Algorithm based approach for bilevel programming models. *ASCE Journal of Transportation Engineering* 126(2):115–120
123. Yin Y (2002) Multiobjective bilevel optimization for transportation planning and management problems. *Journal of Advanced Transportation* 36(1): 93–105
124. Yuchi M, Kim JH (2004) Grouping-based evolutionary algorithm: seeking balance between feasible and infeasible individuals of constrained optimization problems. *Proceedings of the IEEE Congress on Evolutionary Computation*:280–287
125. Zhang XP (2010) Overview of electricity market equilibrium problems and market power analysis. In Zhang XP (ed) *Restructured Electric Power Systems: analysis of electricity markets with equilibrium models*. John Wiley, Hoboken, New Jersey:99–137
126. Zhao Z, Gu X (2006) Particle swarm optimization based algorithm for bilevel programming problems. *Proceedings of the Sixth IEEE International Conference on Intelligent Systems Design and Applications*:951-956