Conference paper

Koh, ATM (2010) *Differential evolution based bi-level programming algorithm for computing normalized nash equilibrium.* In: 15th Online World Conference On Soft Computing In Industrial Applications, 15-27 November 2010, World Wide Web.

(Published proceedings to follow.)

Differential Evolution Based Bi-Level Programming Algorithm for Computing Normalized Nash Equilibrium

Andrew Koh

Abstract The Generalised Nash Equilibrium Problem (GNEP) is a Nash game with the distinct feature that the feasible strategy set of a player depends on the strategies chosen by all her opponents in the game. This characteristic distinguishes the GNEP from a conventional Nash Game. These shared constraints on each player's decision space, being dependent on decisions of others in the game, increases its computational difficulty. A special solution of the GNEP is the Nash Normalized Equilibrium which can be obtained by transforming the GNEP into a bi-level program with an optimal value of zero in the upper level. In this paper, we propose a Differential Evolution based Bi-Level Programming algorithm embodying Stochastic Ranking to handle constraints (DEBLP-SR) to solve the resulting bi-level programming formulation. Numerical examples of GNEPs drawn from the literature are used to illustrate the performance of the proposed algorithm.

1 Introduction

In a game when a rational agent optimizes her welfare in the presence of others symmetrically doing the same simultaneously, game theory [23] provides a way to analyze the strategic decision variables of all players. The solution concept of such games was analyzed by Nash in [16]. A game is considered to have attained 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 showing that a player can profitably deviate) is usually not difficult, determining the NE itself is more challenging. A review of some deterministic and stochastic methodologies for determination of NE is found in [13].

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

This paper is concerned with a special class of Nash Games known as the Generalized Nash Equilibrium Problem (GNEP). In the GNEP, the players' payoffs and their strategies are continuous (and subsets of the real line) but more importantly the GNEP possesses the distinctive feature that players face constraints depending on the strategies their opponents choose. This feature is in contrast to more common Nash Game where the utility/payoff/reward the players obtain depend solely on the decisions they make and their actions are not restricted because of the strategies chosen by others. The ensuing constrained action space in GNEPs makes them more difficult to resolve than conventional Nash games discussed in monographs such as [23]. We point out in passing that the solution algorithm proposed in this paper can be easily applied to conventional Nash games (see below).

This paper is structured thus. Following this introduction, we introduce the GNEP formally along with the various game theoretic terminologies. In Sect 3 the key result emphasized is that the GNEP can be formulated as a bi-level program. Sect 4 outlines DEBLP-SR, a Differential Evolution based algorithm integrated with deterministic gradient based solvers and embodying stochastic ranking to resolve the resulting bi-level formulation. Numerical examples from the literature are discussed in Sect 5. Results of runs using the proposed DEBLP-SR are presented in Sect 6 and Sect 7 wraps up with some concluding remarks.

2 Nash Equilibrium and the GNEP

This section introduces the notation used throughout this work. The GNEP is a single shot normal form game with a set N of players indexed by $i \in \{1,2,\ldots,n\}$ and each player can play a strategy $x_i \in X_i$ which all players are assumed to announce simultaneously. $X \subseteq \mathbb{R}^m$ is the collective action space for all players. In a standard Nash Game, $X = \prod_{i=1}^{n} X_i$. X is thus equal to the Cartesian product. In contrast, in a GNEP, the feasible strategies for player $i \in N$ depend on the strategies of all other players [1],[4],[10],[21]. We denote the feasible strategy space of each player by the point to set mapping: $K^i : X^{-i} \to X^i, \forall i \in N$ that emphasizes the ability of other players to influence the strategies available to player i [4],[7],[21]. The distinction between a conventional Nash game and a GNEP is therefore analogous to the distinction between unconstrained and constrained optimization.

To emphasize the variables chosen by player i, we write $\mathbf{x} \equiv (x_i, x_{-i})$ where x_{-i} is the combined strategies of all players in the game excluding that of player i i.e. $x_{-i} \equiv (x_1,\ldots,x_{(i-1)},x_{(i+1)},\ldots,x_n)$. Critically note that the notation (x_i, x_{-i}) *does not* mean that the components of $\mathbf{x}$ are reordered such that x_i becomes the first block. In addition, let $\phi_i(\mathbf{x})$ be the payoff/reward to player $i, i \in N$ if $\mathbf{x}$ is played.

Definition 1 *[21] A combined strategy profile $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*) \in X$ is a Generalized Nash Equilibrium for the game if:*

$$\phi_i(x_i^*, x_{-i}^*) \geq \phi_i(x_i, x_{-i}^*), \ \forall x_i \in K(x_{-i}^*)\ , \forall i \in \{1,2,\ldots,n\}.$$

At a Nash Equilibrium no player can benefit (increase individual payoffs) by unilaterally deviating from her current chosen strategy. Players are assumed not to cooperate and so each is doing the best she can given what her competitors are doing [5],[13],[23]. For a GNEP, the strategy profile $\mathbf{x}^*$ is a Generalized Nash Equilibrium (GNE) if it is feasible with respect to the mapping K^i *and* if it is a maximizer of each player's utility over the constrained feasible set [7].

2.1 Nikaido Isoda Function

The Nikaido Isoda (NI) function in Eq 1 is an useful tool used in the study of Nash Equilibrium problems eg. [3],[4],[8],[10]. Its interpretation is as follows: each summand shows the increase in payoff a player will receive by unilaterally deviating and playing a strategy $y_i \in K(x_{-i})$ while other players play according to x.

$$\Psi(\mathbf{x},\mathbf{y}) = \sum_{1}^{n} \left[\phi_i(y_i, x_{-i}) - \phi_i(x_i, x_{-i})\right], \forall i \in \{1,2,...,n\} \tag{1}$$

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 Nash Equilibrium point by virtue of Definition 1 since at a Nash Equilibrium no player can increase their payoff by unilaterally deviating. This result is summarized in Definition 2.

Definition 2 *[10] A vector $\mathbf{x}^* \in X$ is called a Normalized Nash Equilibrium if $\Psi(\mathbf{x},\mathbf{y}) = 0$.*

2.2 Solution Approaches for the GNEP

A review of solution methods for the GNEP is discussed in the survey [4]. Deterministic (i.e. gradient-based) descent, the subject of detailed study in Von Heusinger's PhD thesis [9], is the primary solution approach for finding Normalized Nash Equilibrium (NNE). Krawczyk et al [3],[8],[14] also proposed another deterministic descent method based on minimization of the Nikaido-Isoda function. In this paper however we exploit the proof that we can find the NNE by formulating the GNEP as a special bi-level program [2],[21] as discussed in the following section.

3 A Bi-Level Programming Approach for GNEPs

The NNE solution to the GNEP can be found by solving a bi-level programming problem given by the system of equations in 2 and 3. For a proof see [2],[21].

$$\max_{(x,y)} \quad f(x,y) = -(y-x)^T(y-x) \tag{2a}$$

$$\text{subject to } x^i \in K^i(x^{-i}) \,, \forall i \in \{1,2,...,n\}. \tag{2b}$$

where y solves

$$\max_{(x,y)}(\phi_1(y^1,x^{-1}) + \ldots + \phi_n(y^n,x^{-n})) \equiv \max_{(x,y)} \sum_{i=1}^{n} [\phi_i(y_i,x_{-i}) - \phi_i(x_i,x_{-i})] \tag{3a}$$

$$\text{subject to } y^i \in K^i(x^{-i}) \,, \forall i \in \{1,2,...,n\}. \tag{3b}$$

The upper level problem (Eq 2) is a norm minimization problem subject to strategic variable constraints (Eq.2b). The objective function of the lower level problem (Eq 3) is exactly the Nikado Isoda function (Eq 1).

Definition 3 *[21] The optimal value of $f(x,y)$ is 0 at the Normalized Nash Equilibrium.*

Definition 3 will perform the critical role of being the termination criteria of the proposed DEBLP-SR Algorithm discussed in Sect 4.

4 Differential Evolution for Bi-Level Programming

Differential Evolution (DE) for Bi-Level Programming (DEBLP) was initially proposed in [12] to solve Bi-Level Programs (posed as leader-follower games) arising in transportation systems management. It follows the Genetic Algorithms Based Approach proposed in [22] but substitutes the use of binary coded Genetic Algorithms strings with real coded DE [18] as the stochastic optimization method instead.

DEBLP integrates DE manipulation of the upper level variables with gradient based optimization of the lower level problem. The characteristic feature of GNEPs is the constraints facing the players i.e. (Eq.2b); and thus it is necessary to employ constraint handling techniques to produce solutions that satisfy the constraints. Constraint handling methods were not required for the class of Nash Games discussed in [13] and so the technique proposed here is considered more generic.

In the original DEBLP, constraints in the upper level problem were handled by degrading fitness values if constraints were not satisfied via rudimentary penalty methods [12]. In this paper, the upper level constraints in Eq 2 are handled using stochastic ranking [20]. Hence this version of DEBLP is termed DEBLP-SR.

The pseudo code of DEBLP-SR is summarized in Algorithm 1. DEBLP-SR operates thus: A population of h chromosomes is randomly initialized between the bounds of the problem and the user provides the control parameters (mutation probability and crossover factors) for the DE algorithm [18] . The evaluation of fitness is carried out in a two stage process: In the first stage (lines 5 and 13), each chromosome, representing x the upper level variable, is used as a input argument into the lower level program (Eq 3) parameterized in y. Thus given x we solve the lower level program for y using conventional gradient based optimization methods. In the second stage (lines 6 and 14), x and y are used to compute Eq 2 ($f(x)$ in line 10). This measures how far the chromosome is from the optimal value of 0 (cf Definition 3) and thus represents the fitness of the chromosome x. In addition, the constraint violation are also output by the evaluation routine (line 7 and 15).

Stochastic ranking (SR), a robust procedure for handling constraints, uses a stochastic bubble sort procedure to rank population members taking into account both the objective function value and constraint violations. (See [20] for more details). In the first iteration (line 9) the best member of the population is the member that is assigned a rank of 1 (one) by the SR algorithm. DE operations are subsequently used to evolve child chromosomes and evaluated following the two stage process described in the foregoing.

1: Input: h, Max_{it}, DE Control Parameters (Mutation Probability, Crossover Factor)
2: $it \leftarrow 0$
3: Randomly initialize a population of h parent chromosomes $\mathscr{P}$
4: **for** $j = 1$ to h **do**
5:     Solve Eq 3 using deterministic optimization given chromosome $j \in \mathscr{P}$
6:     Compute Eq 2 to evaluate fitness of chromosome $j \in \mathscr{P}$
7:     Compute constraint violation of chromosome $j \in \mathscr{P}$
8: **end for**
9: Apply Stochastic Ranking to rank each member of $\mathscr{P}$ (between 1 (best member) and h)
10: **while** $it < Max_{it}$ or $f(x) \neq 0$ **do**
11:     Apply DE/best/1/bin [18] to create a child population $\mathscr{C}$
12:     **for** $j = 1$ to h **do**
13:         Solve Eq 3 using deterministic optimization given chromosome $j \in \mathscr{C}$
14:         Compute Eq 2 to evaluate fitness of chromosome $j \in \mathscr{C}$
15:         Compute constraint violation of chromosome $j \in \mathscr{C}$
16:     **end for**
17:     Pool Parents and Children Chromosomes:
18:     $\mathscr{T} \leftarrow \mathscr{P} \cup \mathscr{C}$
19:     Apply Stochastic Ranking to rank each member of $\mathscr{T}$ (between 1 (best member) and h)
20:     $\mathscr{P} \leftarrow MaxRank(\mathscr{T})$
21:     **if** $f(x) = 0$ **then**
22:         Terminate
23:     **else**
24:         $it \leftarrow it + 1$
25:     **end if**
26: **end while**
27: Output: Normalized Nash Equilibrium

Algorithm 1: :Differential Evolution for Bi-Level Programming with Stochastic Ranking (DEBLP-SR)

To utilize the ranking information generated by SR, we modify the selection procedure used for determining whether parent or child survive into the next generation. Instead of the one to one greedy selection found in canonical DE [18], we pool the entire set of parent and child chromosomes together and then apply SR to identify the top h ranked population members which will survive (this is the set returned by the *MaxRank* procedure in line 20 of Algorithm 1). The rest of the population is discarded and such a selection procedure is reminiscent of that used in e.g. GENITOR [24]. If the best fitness attains the value of 0 *and* constraints are satisfied, then we have found the NNE and the algorithm terminates, else the iteration counter is increased and the process is repeated until Max_{it} generations are exceeded.

5 Numerical Examples

In this section, we give details of the numerical examples to which we apply DEBLP-SR and present the results of numerical experiments in Sect 6.

5.1 Problem 1

Problem 1, from [19] was originally solved using a projected gradient method. This game has 2 players with 1 decision variable each. Player 1's objective is:

$$\phi_1(x_1,x_2) = \frac{1}{2}(x_1)^2 - x_1x_2$$

Player 2's objective is:

$$\phi_2(x_1,x_2) = (x_2)^2 + x_1x_2$$

The feasible space is defined according to:

$$X = \{x \in \Re^2 | x_1 \geq 0, x_2 \geq 0, -x_1 - x_2 \leq -1\}$$

As an example, we give the NI function explicitly as:

$$\Psi(\mathbf{x},\mathbf{y}) = [(\frac{1}{2}{x_1}^2 - x_1x_2) - (\frac{1}{2}{y_1}^2 - y_1x_2)] + [({x_2}^2 + x_1x_2) - ({y_2}^2 + x_1y_2)]$$

The NNE is $x_1^* = 1, x_2^* = 0$ [9],[19].

5.2 Problem 2

Problem 2, again with 2 players and 1 decision variable each, comes from Harker [7]. Player 1's objective is:

$$\phi_1(x_1,x_2) = (x_1)^2 + \frac{8}{3}x_1x_2 - 34x_1$$

Player 2's objective is:

$$\phi_2(x_1, x_2) = (x_2)^2 + \frac{5}{4} x_1 x_2 - 24.25 x_2$$

The feasible space is defined according to:

$$X = \{x \in \Re^2 | x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 15\}$$

The NNE is $x_1^* = 5, x_2^* = 9$ [7], [9].

5.3 Problem 3

This problem describes an Environmental Pollution Control Problem known as the "River Basin Pollution Game" studied by Krawczyk and co-workers [8],[14]. There are 3 players with 1 variable each. The objective function for player j is:

$$\phi_j(\mathbf{x}) = (c_{1j} + c_{2j}) x_j - (3 - 0.01(x_1 + x_2 + x_3)) x_j, \forall j \in \{1, 2, 3\}$$

The feasible space is defined according to:

$$3.25 x_1 + 1.25 x_2 + 4.125 x_3 \leq 100$$

$$2.2915 x_1 + 1.5625 x_2 + 2.8125 x_3 \leq 100$$

$$x_j \geq 0, \forall j \in \{1, 2, 3\}$$

The NNE is $x_1^* = 21.14, x_2^* = 16.03, x_3^* = 2.927$ [9], [8], [14].

5.4 Problem 4

This problem describes an internet switching model with 10 players proposed in [11] and also studied in [9]. The cost function for player j is given by

$$\phi_j(\mathbf{x}) = -(\frac{x_j}{(x_1 + \ldots + x_{10})})(1 - \frac{(x_1 + \ldots + x_{10})}{1}), \quad \forall j \in \{1, \ldots, 10\}$$

The feasible solution space is:

$$X = \{x \in \Re^{10} | x_j \geq 0.01, \forall j \in \{1, \ldots, 10\}, \sum_{j=1}^{10} x_j \leq 1\}$$

The NNE is $x_j^* = 0.09, \quad \forall j = \{1, \ldots, 10\}$ [10].

5.5 Problems 5a and 5b

The last problem studied is a non-linear Cournot-Nash Game with 5 players proposed in [15] which we refer to as Problem 5a. Inclusion of Problem 5a serves to emphasize that the method articulated here can be applied to both standard Nash Games and GNEPs and thus demonstrate that the method in this paper is more general than that proposed in [13]. With the introduction of a production constraint in [17], it is transformed into a GNEP (and referred to as Problem 5b herein).

For both problems, each player's cost function is given as:

$$\phi_j(\mathbf{x}) = (x_j) = c_j x_j + (\frac{\beta_j}{\beta_j + 1}) L_j^{\frac{-1}{\beta_j}} x_j^{\frac{\beta_j+1}{\beta_j}} \quad - P(\mathbf{x}) x_j, \quad \forall j \in \{1, ..., 5\}$$

$$P(\mathbf{x}) = 5000^{\frac{1}{1.1}} (\sum_{j=1}^{5} x_j)^{-(\frac{1}{1.1})}, \forall j \in \{1, ..., 5\}$$

The firm dependent parameters (c_j, $\beta_j$and L_j) are found in [15],[17]. The feasible space for Problem 5a (NEP) is the positive axis as production cannot be negative. The solution of the NEP is $\mathbf{x}^* = [36.9318, 41.8175, 43.7060, 42.6588, 39.1786]^{\mathrm{T}}$ [6],[15].

The feasible space for the GNEP variant includes a joint production constraint (Problem 5b) given as follows: [17]

$$X = \{x \in \Re^5 | x_j \geq 0 \ \forall j \in \{1, ..., 5\}, \sum_{j=1}^{5} x_j \leq M\}$$

For the case where $M = 100$, then the NNE (for GNEP variant 5b) is $\mathbf{x}^* = [14.050, 17.798, 20.907, 23.111, 24.133]^{\mathrm{T}}$ [9].

6 Results

In numerical experiments, we carried out 30 independent runs of DEBLP-SR for each test problem allowing for a maximum of 200 iterations (Max_{it}) each run. Based on Definition 3, we terminate the algorithm when the objective function reaches a value of 0[1]. When this target value is reached *and* the maximum constraint violation is less than 0.000001, we deem a run to be "successful " and the number of such runs are reported in Table 1. All runs also utilize the DE/best/1/bin strategy [18]. The crossover and mutation factor applied to all problems are both set 0.9 without any parameter tuning. Our results illustrate that the algorithm is very useful for simpler problems but robustness (as measured by standard deviation and number of successful runs out of 30) decreases as both non-linearity (c.f. Problem 5) and

[1] In practice we terminate when the best objective reached is less than or equal to 0.000001.

number of players increases (c.f. Problem 4). However, no solution would be valid if it does not satisfy the constraints and it is evident that all constraints are satisfied for all problems since the maximum constraint violation for each run is zero.

For the purposes of comparing DEBLP-SR against others proposed in the literature, we also used PSwarm [25], which is explicitly designed to handle both bound and linear constraints, to solve our test problems. We are unable to include a comparison of DEBLP-SR with PSWARM due to space constraints but instead have made the performance comparison available at `http://goo.gl/bupz0`. For this we used the MATLAB version of PSWARM available on the world wide web at `http://www.norg.uminho.pt/aivaz/pswarm`.

Table 1 Results of Application of DEBLP-SR to Test Problems defined in Section 5

Problem Number	1	2	3	4	5a	5b
Best Objective	1E-08	0	2E-08	0	0	0
Worst Objective	1E-07	0	1E-07	3.4E-05	0	1.2E-07
Mean Objective	5.13E-08	0	5.5E-08	4.2E-06	0	5.8E-08
Median Objective	5E-08	0	5E-08	0	0	8E-08
Standard Deviation	2.57E-08	0	2.84E-08	1.1E-05	0	5.33E-08
Maximum Constraint Violation	0	0	0	0	NA	0
Minimum No. of Function Evaluations	400	420	800	4320	1920	2700
Maximum No. of Function Evaluations	660	640	1940	6000	2790	6000
Mean No. of Function Evaluations	529	535	1081	5618	2388	4935
Median No. of Function Evaluations	520	520	1040	5730	2430	5685
Population Size	20	20	20	30	30	30
No. of Successful Runs	30	30	30	24	30	30

7 Conclusions

The Generalized Nash Equilibrium Problem is a Nash Game with the characteristic that the strategic options open to each player depend on what others have chosen. One particular solution of the GNEP is the Normalized Nash Equilibrium which can be found by solving a specialized bi-level program. We have demonstrated the use of a heuristic method which integrates deterministic optimization with Differential Evolution to solve the resulting bi-level program. DEBLP-SR incorporates stochastic ranking to deal with constraints and tournament selection to select survivors when comparing parent and child chromosomes. We illustrated the performance of DEBLP-SR with numerical examples drawn from the literature and evidence suggests that DEBLP-SR is a viable algorithm for this class of Nash games.

Acknowledgements The research reported here is funded by the Engineering and Physical Sciences Research Council of the UK under the “Competitive Cities” Grant EP/H021345/1. The author thanks Dr Simon Shepherd and colleagues for comments on an earlier draft.

References

1. Aussel D , Dutta J: Generalized Nash equilibrium problem, variational inequality and quasiconvexity, Oper Res Lett 36 (4), 461-464 (2008)
2. Bouza Allende GB: On the calculation of Nash Equilibrium points with the aid of the smoothing approach, Revista Investigación Operacional 29(1), 71–76 (2008)
3. Contreras J, Klusch M, Krawzyck JB: Numerical solutions to Nash-Cournot equilibrium in electricity Markets, IEEE T Power Syst 19(1), 195–206 (2004)
4. Facchinei F, Kanzow C: Generalized Nash equilibrium problems. 4OR 5(3), 173–210 (2007)
5. Gabay D, Moulin H: On the uniqueness and stability of Nash-equilibria in non cooperative games, In: Bensoussan A, et al (eds) Applied Stochastic Control in Econometrics and Management Science. North Holland, Amsterdam, 271–293 (1980)
6. Harker PT: A variational inequality approach for the determination of Oligopolistic Market Equilibrium, Math Program 30(1),105–111 (1984)
7. Harker PT: Generalized Nash games and quasi-variational inequalities, E J Oper Res 54(1),81–94 (1991)
8. Haurie A, Krawczyk J: Optimal charges on river effluent from lumped and distributed sources. Environ Model Assess 2(3), 177-189 (1997
9. von Heusinger A: Numerical Methods for the Solution of the Generalized Nash Equilibrium Problem. PhD Thesis, Institute of Mathematics, University of Wüzburg, Würzburg, (2009)
10. von Heusinger A, Kanzow C: Relaxation Methods for Generalized Nash Equilibrium Problems with Inexact Line Search. J Optimiz Theory App 143(1), 159–183 (2009)
11. Kesselman A, Leonardi S, Bonifaci, V: Game-Theoretic Analysis of Internet Switching with Selfish Users, Internet and Network Economics, 3828, Springer, Berlin, 236–245 (2005)
12. Koh A: Solving transportation bi-level programs with Differential Evolution, Proc IEEE CEC, Singapore, 25 Sept– 28 Sept, 2243–2250 (2007)
13. Koh A: Nash Dominance with Applications to Equilibrium Problems with Equilibrium Constraints, In: Gao X, et al (eds) Soft Computing in Industrial Applications: Algorithms, Integration, and Success Stories. Advances in Soft Computing 75, Springer, Berlin 71–79 (2010)
14. Krawczyk J, Uryasev, S: Relaxation algorithms to find Nash equilibria with economic applications, Environ Model Assess 5(1),63-73 (2000)
15. Murphy FH, Sherali HD, Soyster AL: A mathematical programming approach for determining oligopolistic market equilibrium, Math Program 24(1), 92–106 (1982)
16. Nash J: Equilibrium points in N-person games. Proc Natl Acad Sci USA 36(1), 48–49 (1950)
17. Outrata JV, Kočvara M, Zowe J, Nonsmooth Approach to Optimization Problems with Equilibrium Constraints. Nonconvex Optimization and its Applications 28, Kluwer,Dordrecht (1998)
18. Price K, Storn R, Lampinen J: Differential evolution: a practical approach to global optimization. Springer, Berlin (2005)
19. Rosen JB: Existence and Uniqueness of Equilibrium Points for Concave N-person Games, Econometrics, 33(3), 520–534 (1965)
20. Runarsson TP, Yao X: Search biases in constrained evolutionary optimization. IEEE T Syst Man Cy C 35(2): 233-243 (2005)
21. Sun L: Equivalent Bilevel Programming Form for the Generalized Nash Equilibrium Problem, Journal of Mathematics Research 2(1), 8–13 (2010)
22. Yin Y: Genetic Algorithm based approach for bilevel programming models. J Transp Eng-ASCE 126(2),115–120 (2000)
23. Webb JN: Game theory: decisions, interaction and Evolution. Springer, London (2007)
24. Whitley, D: The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. Proc ICGA3 Fairfax, Virginia, 116–121 (1989)
25. Vaz AIF, Vicente LN: PSwarm: A hybrid solver for linearly constrained global derivative-free optimization, Optim Method Softw, 24 (4-5), 669–685 (2009)