Cui Yanhong

Guo Renkuan
University of Cape Town, Cape Town, South Africa

Nash-lambda algorithm with applications in safety and reliability

Keywords

lambda algorithm, Nash equilibrium, genetic algorithm, safety, reliability, terrorist threat

Abstract

In this paper, a new algorithm, named as Nash-lambda algorithm by merging Nash equilibrium solution and the lambda algorithm, is proposed. The lambda algorithm, a new global optimization algorithm, is created by imitating ancient Chinese human body system model, which has already demonstrated its simplicity in searching scheme, codes and efficiency in computation comparing to the genetic algorithm. The non-corporative game environments determine the optimization problems which are different from those of the traditional safety and reliability optimizations because of the engagement of the Nash equilibrium for seeking the best strategy. The lambda algorithm serves the searching the Nash equilibrium solution efficiently. In other worlds, the Nash-lambda algorithm is just developed to address the optimization problems of the multiple objective functions representing non-corporative players' interests.

1. Introduction

Safety and reliability optimization problems are a fundamental components intrinsically in the sense that the statistical theory underlying them is built up by a pile of relevant mathematical optimal theories and methodologies. However, since the 911 event occurred in New York, 2001, the threat from the terrorist organizations has merged into the western governments' agenda list [7], [15], [16]. Any government or a utility company, say, the electricity power plant, the water supply company, the public transportation network, the international airport, etc. has the responsibility to secure the highest safety and availability to the public, while the terrorist organization wants to destroy or damage the target to the maximum. It is obvious that the players in the game battle are non-corporative. The optimization problem is no longer the traditional one. Nash equilibrium is "a solution concept of a game involving two or more players, in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his own strategy unilaterally", [12]. To obtain the solution set of the Nash equilibrium, it is necessary to search it within the players' strategy sets. There have been many search methodologies, for example, Nash-LQ, Nash-polynomial algorithms etc.

It is noticeable that researchers have try to merge Nash equilibrium solution and the genetic algorithm (abbreviated by Nash-GA) for seeking optimal numerical strategies [21], [24]. The lambda algorithm is created by imitating an ancient human body system [4], [5], [6], also the sister paper in this seminar, "Lambda algorithm and maximum likelihood estimation". In its searching scheme, except the necessary mathematical computations for evaluating the objective function and the creation of the initial "searching population" randomly, the algorithm only involves if-else logical operation and sort procedure. In contrast to existing global optimization algorithms, particularly GA, the lambda algorithm engages the simplest mathematics but reaches the highest searching efficiency. Therefore it is logical to consider in the Nash-GA replacing the genetic algorithm (GA) part by the lambda algorithm for merging Nash equilibrium solution concept with lambda algorithm to achieve the optimal numerical

strategies because of the merits of it comparing to GA.
The remaining structure of the paper is stated as following: Section two serves the explanation of the Nash equilibrium solution and related theory; Section three will analyze the merging of Nash equilibrium solution and lambda algorithm and analyzing an numerical example to illustrate the new Nash-lambda algorithm; Section four will discuss briefly the applications in safety and reliability optimizations; Section five concludes this paper.

2. Nash equilibrium solution concept

The game theory is a applied mathematical branch dealing with the behaviour in strategic situations, in which an individual's gain in making choices depends on the choices of the individual's competitors. Game theory studies theory on the rational side of social science in broad sense, including human as well as non-human players e.g., computers, animals, and etc., [10].
In *n*-player non-corporative games, the Nash equilibrium is a solution state, in which an individual player knows the strategies of the others and also knows that no one can gain anything by altering any individual strategy unilaterally while the others keep their strategies unchanged. Such a set of strategy choices and the corresponding payoffs constitute a Nash equilibrium, [12].
Let (S, f) be a game with n players, in which $S = S_1 \times S_2 \cdots \times S_n$ is the strategy-profile set with the i^{th} player's strategy set S_i, $i = 1, 2, \cdots, n$, and $f = f\left(f_1(x), \cdots, f_n(x)\right)$ is the payoff function. When each individual player decides to choose the strategy x_i, then a strategy profile $x = \left(x_1, \cdots, x_n\right)$ is obtained so that the i^{th} player i obtains payoff $f_i\left(x\right)$. Let x_{-i} be a strategy profile of all players except for the i^{th} player. Note that the payoff depends on the strategy profile chosen, i.e. on the strategy chosen by player i as well as the strategies chosen by all the remaining players.
Definition 1. A strategy profile $x^* \in S$ is Nash equilibrium if no unilateral deviation in strategy by any individual player is profitable for that player, that is

$$\forall i, x_i \in S_i, x_i \neq x_i^* : f_i\left(x_i^*, x_{-i}^*\right) \geq f_i\left(x_i, x_{-i}^*\right). \qquad (1)$$

A game can have either a pure-strategy or a mixed-strategy Nash Equilibrium, (in the latter a pure strategy is chosen stochastically with a fixed frequency). Nash proved that if we allow mixed strategies, then every game with a finite number of players in which each player can choose from finitely many pure strategies has at least one Nash equilibrium solution [11], [12].

2.1. Bi-level program

In the multilevel programming problem, the notation level is actually the sets of variables. For example, a bi-level program, (bi-level programming) has two sets of variables [9].
Definition 2. A bi-level program is the optimization problem within which one optimization problem is embedded in another one.
As a matter of fact, the formulation of a bi-level programming problem can be stated simply as:

$$\min_{x \in X, y \in Y} f^u\left(x, y\right) \qquad (2)$$

Subject to:

$$\begin{aligned} & g^u\left(x, y\right) \leq 0, \\ & y \in \arg\min_{z \in Y} f^l\left(x, z\right) \\ & g^l\left(x, z\right) \leq 0, \end{aligned} \qquad (3)$$

where

$$\begin{aligned} & f^u, f^l : R^{nx} \times R^{ny} \to R \\ & g^u, g^l : R^{nx} \times R^{ny} \to R^{mu} \\ & X \subseteq R^{nx} \\ & Y \subseteq R^{ny}. \end{aligned} \qquad (4)$$

where the variables z are dummy variables.

2.2. Stackelberg model

Decision making problems in decentralized organizations are often modelled as Stackelberg competitions, which are formulated as two-level mathematical programming problems [13], [19], [22], [23]. Conflict and cooperation among individual players are an essential part of the process. In the Stackelberg game model, there are two kinds of players; the player of the first kind chooses a strategy at the start, and thereafter the player of the second kind with knowledge of the player’s strategy of the first kind determines a strategy of the player of the second kind.
In game theory, players are classified as a leader and the remaining ones as the followers. Stackelberg model is a strategic game in which "the leader firm moves first and then the follower firms move sequentially", ..., the constraints for maintaining the Stackelberg equilibrium is that "the leader must know *ex ante* that the follower observes his action. The follower must have no means of committing

to a future non-Stackelberg follower action" [13]. "The Stackelberg model can be solved to find the subgame perfect Nash equilibrium or equilibria (SPNE), i.e. the strategy profile that serves best each player, given the strategies of the other player and that entails every player playing in a Nash equilibrium in every subgame" [13].

Definition 3. Let $x \in \mathrm{R}^N$ be partitioned as $x = \left(x^\alpha, x^\beta\right)$, and a compact set $\mathrm{S} \subset \mathrm{R}^N$. Let $f : \mathrm{R}^N \to \mathrm{R}$ and be continuous on S. The set $\Re_f(\mathrm{S}) \triangleq \left\{ \hat{x} \in \mathrm{S} | f(\hat{x}) = \max_{x \in \mathrm{S} \cap \{x^\beta = \hat{x}^\beta\}} f(x) \right\}$ is the one of rational reactions under function f on the set S.

To formally define the *n*-player Stackelberg game model, let $x \in \mathrm{R}^N$ be the vector of decision variables for all *n* players, and let x be partitioned among *n* players with $x^k \triangleq \left(x_1^k, x_2^k, \cdots, x_{N_k}^k\right) \in \mathrm{R}^{N_k}$, $k = 1, 2, \cdots, n.$ Note that $\sum_{k=1}^n N_k = N$. The game model requires all *n* players take x from S^l, whose shape determines the ability of the leader player to affect the set of feasible choices of the follow players. Let $f_k : \mathrm{S}^k \to \mathrm{R}$, $k = 1, 2, \cdots, n,$ $\left\{f_1(x), f_2(x), \cdots, f_n(x)\right\}$ the set of continuous functions.

Definition 4. Let x be partitioned as $x = \left(x^\alpha, x^\beta\right)$ with $x^\alpha \triangleq \left(x^1, x^2, \cdots, x^{k-1}\right)$ and $x^\beta \triangleq \left(x^k, x^{k+1}, \cdots, x^n\right)$. The level-*k* feasible region $\mathrm{S}^k \triangleq \Re_{f_{k-1}}\left(\mathrm{S}^{k-1}\right)$ recursively for $k = 2, 3, \cdots, n.$

The set S^k collects the feasible outcomes resulting from the rational reactions of players at level-*i*, $i = 1, 2, \cdots, k-1.$. Hence S^k contains all of the information necessary for player *i* to evaluate the behaviour of these players. Given the preemptive decisions $\left(\hat{x}^{k+1}, \hat{x}^{k+2}, \cdots, \hat{x}^n\right)$ of the first *n* − *k* leading players, the optimization problem which must be solved by the player at level *k* is then

$$
\begin{aligned}
&\left(L^k\right): \max\ f_k(x) \\
&\text{s.t.} \\
&x \in \mathrm{S}^k, \\
&x^i = \hat{x}^i,\ i = k+1, \cdots, n
\end{aligned}
\tag{5}
$$

This presents a nested multi-level programming problem.

It is quite obvious that Stackelberg model, a pure strategy optimization may have only one Nash equilibrium, while mixed strategies could have finitely many Nash equilibria (at least one). The lambda algorithm is designed for both pure strategy and mixed strategies optimization for bi-level programming, which is named as Nash-lambda algorithm.

Nash-lambda algorithm allowed program at each loop of optimization evaluate two strategy objective functions. A switch function to decide the rank of all the candidate solutions. If switch=0, then the algorithm according to leader objective function to rank the candidate solutions. If switch=1, then the algorithm according to follower objective function to rank the candidate solutions. $TempF_{leaders}^{best}$, $TempF_{followers}^{best}$ are two variables, which using to record the best optimization result of leader, follower objective function in the elapsed optimization. $\underline{e_{leaders}^{best}}$ is the best fitness string of leader objective function at current loop. $\underline{e_{followers}^{best}}$ is the best fitness string of follower objective function at current loop. $F_{leaders}^{best}$, $F_{leaders}^{judge}$ are fitness values of $\underline{e_{leaders}^{best}}$ from leader, follower objective function evaluation respectively. Similarly, $F_{followers}^{best}$, $F_{followers}^{judge}$ are fitness values of $\underline{e_{followers}^{best}}$ from follower, leader objective function evaluation respectively.

In pure strategy optimization:

If $F_{followers}^{best} \geq TempF_{followers}^{best}$, Switch=0
Else if $F_{leaders}^{best} \geq TempF_{leaders}^{best}$, Switch=1,
End

The above program code meaning, for leader objective function and follower objective function, each different strategy optimization only allowed jumping once at the algorithm. After one objective function have a better fitness value, and then the algorithm must turn to face another objective to do the optimization. If the algorithm running towards to leader objective function optimization, one selected strings vector $\overline{e_{new}^{first}}$ must let all the candidate solution take the leader variables values given by $\underline{e_{leaders}^{best}}$. The meaning is, except $\underline{e_{leaders}^{best}}$, other strings must copy the digits which represent the leader objective function variable $\underline{e_{leaders}^{best}}$ has. Similarly, if the algorithm running towards to follower objective function optimization, one selected strings vector $\overline{e_{new}^{first}}$ must let all the candidate solution take the follower variables values given by $\underline{e_{followers}^{best}}$.

The optimization result is, after “step by step”, or say one time by one time altering optimization, if one way of the optimization is stopped, which meaning one way of the strategy is successful, a pure strategy reaches the Nash equilibrium.

In mixed strategies optimization:

If $F_{leaders}^{judge} \geq TempF_{followers}^{best}$, Switch=0
Else if $F_{followers}^{judge} \geq TempF_{leaders}^{best}$, Switch=1,
End

The above program code meaning, instead of "step by step" altering optimization, the algorithm allowed optimization continues jumping at one direction. Only when the current best fitness is the best fitness of both leader and follower objective function, the algorithm allowed the optimization towards to another way. The optimization result is more balanced in this way, which can give many more Nash equilibrium for different strategies. The flow chart of Nash-lambda algorithm is showing in Figure 1.

3. A numerical example

In this section, we consider a bi-level programming with free followers in which the leader has a decision vector (x_1, x_2, x_3) and the three followers have decision vectors $(y_{i1}, y_{i2}), i = 1, 2, 3,$ see [1].

$$\begin{cases} \max\limits_{x_1,x_2,x_3} y_{11}^* y_{12}^* \sin x_1 + 2 y_{21}^* y_{22}^* \sin x_2 + 3 y_{31}^* y_{32}^* \sin x_3 \\ \text{subject to:} \\ x_1 + x_2 + x_3 \leq 10, x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, \\ \left(y_{11}^*, y_{12}^*, y_{21}^*, y_{22}^*, y_{31}^* y_{32}^*\right) \text{ solves the problems} \\ \begin{cases} \max\limits_{y_{11},y_{12}} y_{11} \sin y_{12} + y_{12} \sin y_{11} \\ \text{subject to:} \\ y_{11} + y_{12} \leq x_1, y_{11} \geq 0, y_{12} \geq 0 \end{cases} \\ \begin{cases} \max\limits_{y_{21},y_{22}} y_{21} \sin y_{22} + y_{22} \sin y_{21} \\ \text{subject to:} \\ y_{21} + y_{22} \leq x_2, y_{21} \geq 0, y_{22} \geq 0 \end{cases} \\ \begin{cases} \max\limits_{y_{31},y_{32}} y_{31} \sin y_{32} + y_{32} \sin y_{31} \\ \text{subject to:} \\ y_{31} + y_{32} \leq x_3, y_{31} \geq 0, y_{32} \geq 0 \end{cases} \end{cases} \tag{6}$$

A run of Nash-lambda algorithm 120 generations show that (pure strategy)

$$\left(x_1^*, x_2^*, x_3^*\right) = (1.0371, 0.7698, 8.1155)$$
$$\left(y_{11}^*, y_{12}^*\right) = (0.6569, 0.3796)$$
$$\left(y_{21}^*, y_{22}^*\right) = (0.3639, 0.3805)$$
$$\left(y_{31}^*, y_{32}^*\right) = (2.1396, 5.9705)$$

With optimal objective values

$$y_{11}^* y_{12}^* \sin x_1 + 2 y_{21}^* y_{22}^* \sin x_2 + 3 y_{31}^* y_{32}^* \sin x_3 = 37.4278$$
$$y_{11} \sin y_{12} + y_{12} \sin y_{11} = 0.4752$$
$$y_{21} \sin y_{22} + y_{22} \sin y_{21} = 0.2705$$
$$y_{31} \sin y_{32} + y_{32} \sin y_{31} = 4.3722$$

The pure strategy made leader objective value reaches maximum.

A run of Nash-lambda algorithm 26 generations show that (mixed strategy)

$$\left(x_1^*, x_2^*, x_3^*\right) = (0.4000, 1.6000, 8.0000)$$
$$\left(y_{11}^*, y_{12}^*\right) = (0, 0)$$
$$\left(y_{21}^*, y_{22}^*\right) = (0.7200, 0.8000)$$
$$\left(y_{31}^*, y_{32}^*\right) = (1.9968, 6.0000)$$

With optimal objective values

$$y_{11}^* y_{12}^* \sin x_1 + 2 y_{21}^* y_{22}^* \sin x_2 + 3 y_{31}^* y_{32}^* \sin x_3 = 36.7114$$
$$y_{11} \sin y_{12} + y_{12} \sin y_{11} = 0$$
$$y_{21} \sin y_{22} + y_{22} \sin y_{21} = 1.0440$$
$$y_{31} \sin y_{32} + y_{32} \sin y_{31} = 4.9058$$

A run of Nash-lambda algorithm 44 generations show that (pure strategy)

$$\left(x_1^*, x_2^*, x_3^*\right) = (3.5833, 3.1968, 3.1999)$$
$$\left(y_{11}^*, y_{12}^*\right) = (1.9833, 1.6000)$$
$$\left(y_{21}^*, y_{22}^*\right) = (1.5968, 1.6000)$$
$$\left(y_{31}^*, y_{32}^*\right) = (1.5999, 1.6000)$$

With optimal objective values

$$y_{11}^* y_{12}^* \sin x_1 + 2 y_{21}^* y_{22}^* \sin x_2 + 3 y_{31}^* y_{32}^* \sin x_3 = \text{-}2.0859$$
$$y_{11} \sin y_{12} + y_{12} \sin y_{11} = 3.4482$$
$$y_{21} \sin y_{22} + y_{22} \sin y_{21} = 3.1955$$
$$y_{31} \sin y_{32} + y_{32} \sin y_{31} = 3.1985$$

The pure strategy made followers objective value reaches maximum.

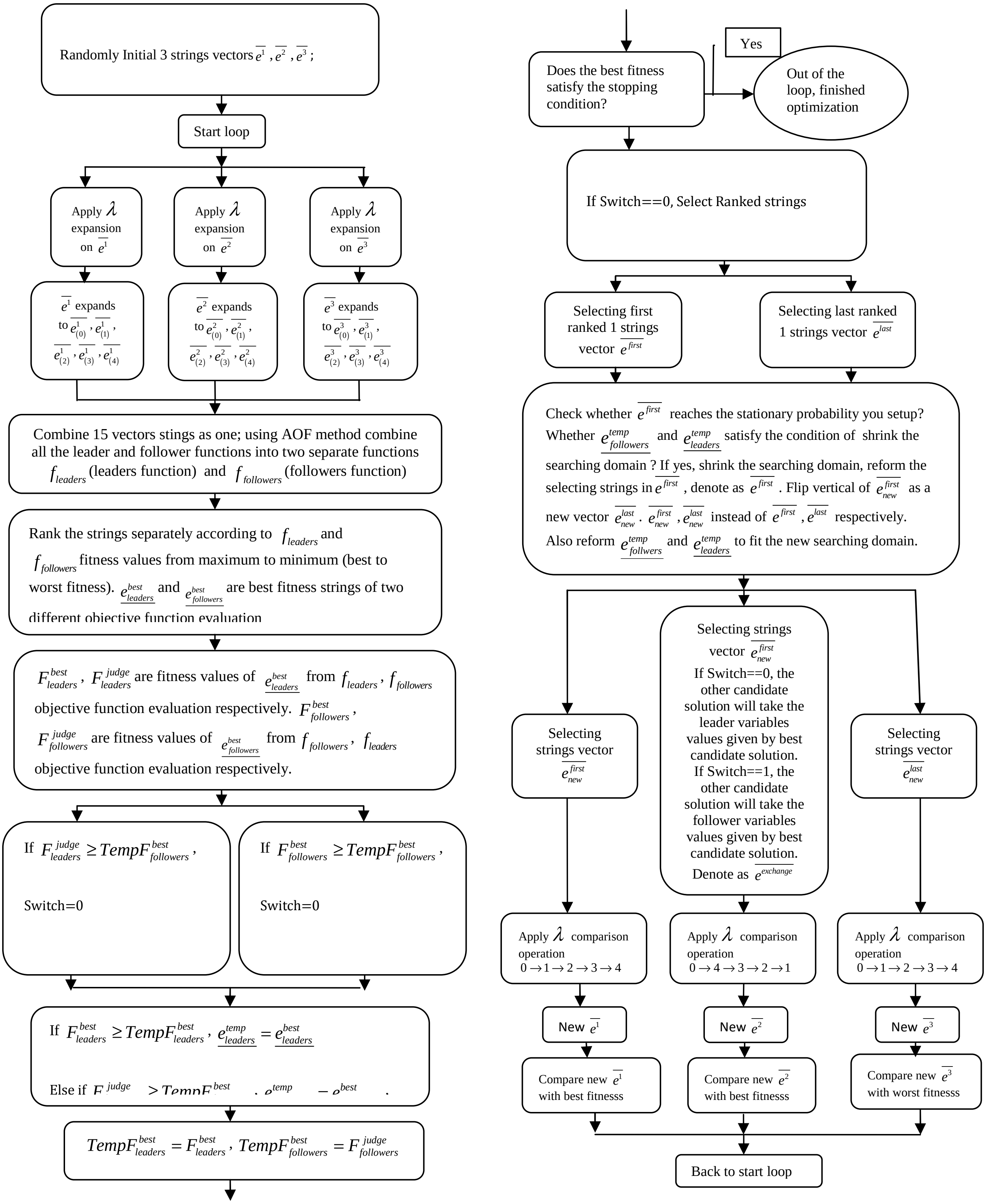

Figure 1. Bi-level programming using Nash-lambda algorithm operation process

4. Applications in safety and reliability

In this section, we will consider a few Nash-lambda algorithm applications in the safety and reliability field with the focus in Subsection 4.1.

4.1. Maintenance schedule problem

In this subsection, let us examine a maintenance scheduling application [18], where the authors defined a new index, named lost opportunity cost of market participation (LOCMP) since every individual generation company (GENCO) targets to maximize its profits except the reliability concerns, which is monitoring constantly by the Independent System Operator (ISO). "ISO as a market supervisor is responsible for power system reliability preservation" [17], and therefore a player in the dynamic game of GENCO against ISO.
The strategy of each GENCO will maximize the profits at the same time will minimize the LOCMP.
The LOCMP is calculated by

$$\text{LOCMP} = \sum_{t}^{week}\sum_{g}^{G}\left(p_t - \left(2\alpha p_{\max,g,t} + \beta\right)\right) p_{\max,g,t} h_t Y_{g,t} \qquad (7)$$

where

p_t	Price for a strategy at time t
$p_{\max,g,t}$	Power generated by units in stage t (MW)
α,β	Cost factors (i.e., x_1, x_2)
h_t	Maintenance hours of unit at stage t
$Y_{g,t}$	Maintenance status of units in stage t (1, or 0)

Let

$C_{g,t}$	Production cost of generation units in stage t (i.e., x_3)
p_t^M	Maintenance cost of generation units

Then the objective function for a GENCO

$$\Lambda = \sum_{t}^{week}\sum_{g}^{G}\left(\left(p_t - C_{g,t}\right) p_{\max,g,t}\left(1 - Y_{g,t}\right) - p_t^M Y_{g,t}\right) \qquad (8)$$

On the other hands, ISO as a player offers a disincentive strategy

$$p_t^{penalty} = \frac{S_t}{\sum_{t=1}^{52} S_t} C_t^{ISO\text{-}PAYMENT} \qquad (9)$$

where

$p_t^{penalty}$	Penalty Index
S_t	Quadratic Penalty Index $\left(EIR_t^{base} - EIR_t^{offered}\right)^2$
EIR_t^{base}	Energy Index Reliability calculated by ISO shows desirable reliability
$EIR_t^{offered}$	Energy Index Reliability calculated by ISO considering offers of GENCOs
$C_t^{ISO\text{-}PAYMENT}$	$\begin{cases}\text{Cost paid by ISO for penalty} \\ \text{Cost of energy not supplied}\end{cases}$

Then the objective function is

$$\Lambda = \sum_{t}^{week}\sum_{g}^{G}\left(\left(p_t - C_{g,t}\right) p_{\max,g,t}\left(1 - Y_{g,t}\right) - p_t^M Y_{g,t} - p_t^{penalty}\right) \qquad (10)$$

which is again a bi-level program suitable for Nash-lambda algorithm because the penalty paid by ISO needs to be minimized.
The authors of [18] engaged simulation approach for seeking the optimal solution. We engage the Nash-lambda scheme for searching the optimal solution. The objective function we used is

$$\Lambda = \sum_{t}^{week}\sum_{g}^{G}\left(\left(p_t - x_1 p_{\max,g,t}\right) p_{\max,g,t}\left(1 - Y_{g,t}\right) - x_3 p_t Y_{g,t}\right) \qquad (14)$$

and the constraint sub-objective function is

$$\text{LOCMP} = \sum_{t}^{week}\sum_{g}^{G}\left(p_t - \left(2x_2 p_{\max,g,t} + x_3\right)\right) p_{\max,g,t} h_{g,t} Y_{g,t} \qquad (15)$$

and thus the bi-level program formation is

$$\begin{aligned}&\max_{x_1,x_2,x_3} \Lambda\left(x_1,x_2,x_3\right)\\ &\text{s.t.}\\ &\min_{x_1,x_2,x_3} \text{LOCMP}\left(x_1,x_2,x_3\right)\end{aligned} \qquad (16)$$

Because we feel short of information, in the problem formulation we identify three cost variable, x_1, x_2, x_3 . The Nash-lambda uses 36.1881seconds, 100 loops for locating the equilibrium numerical solution:

$$x_1 = 99.840, x_2 = 4.992, x_3 = 0.000 \qquad (17)$$

which gives the $\max \Lambda\left(x_1,x_2,x_3\right) = 3.3816\text{E}+009$, subject to $\min \text{LOCMP}\left(x_1,x_2,x_3\right) = 8.8654\text{E}+004$.

4.2. Anti-terrorism

International terrorism has been a principal concern of policy makers and the public since the September 11 attack, 2001, [7], [13]. "The West" and the " International Terrorist Organization (ITO)" are two players in an incentive Stackelberg game model [16]. The objective function is

$$\Lambda\left(x,w,v\right) = -\int_0^T e^{-rt}\left(\gamma_1 x_t + \gamma_2 w_t + \gamma_3 v_t\right) dt + e^{-rT} s x_T \qquad (18)$$

and thus the optimization problem is

$$\max_{v_t} \Lambda\left(x,w,v\right) \qquad (19)$$

subject to

$$
\begin{aligned}
&\dot{x}_t = f(x_t) - \mu w_t - g(v_t) w_t - \phi v_t + h(v_t);\\
&f(x_t) = \gamma(1 - x_t) x_t;\\
&g(v_t) = \beta v_t;\\
&h(v_t) = \alpha v_t^2,
\end{aligned} \tag{20}
$$

where

$x_t \geq 0$	number of terrorists at time *t*
$v_t \geq 0$	intensity of the West's terror control activities at time t
$w_t \geq 0$	number of ITO attacks at time t
$f(x_t)$	endogenous growth of ITO at time t
$\mu \geq 0$	average number of terrorists killed or arrested per attack
$g(v_t)$	number of terrorists lost per terror attack due to terror control efforts v(t),
$\phi \geq 0$	rate at which terror control operations would deplete ITO if the West is on full counter-offensive
$h(v_t)$	growth of ITO at time t due to hatred caused by collateral damage induced by (low-specificity) terror control activities of the West.

with γ, β, and α being positive constants.

The constraint should be the ITO wants to maximize the attacks' damages. It can be solved by a bi-level program and hence Nash-lambda algorithm is able to search its solution by changing the equality constraints into a set of inequality constraints in terms of additional explanatory variables.

4.3. Reliability and free riding

Another interesting of application is the problem of the reliability of public systems. It is well-known fact that the public systems cost the tax payers dearly, however, certain corner of the society (typically those never paid one cent for tax) always steal or damage these goods for self-benefiting. The problem is again a *n*-player non corporative game. Let

x_i	The effort tried by agent *i = 1,2;*
$p(F(x_1, x_2))$	The probability of successful operation of the system;
v_i	The reward received by agent *i* from successful operation of the system;
$c_i x_i$	The cost paid by agent *i* from successful operation of the system.

Then the expected social payoff

$$p(F(x_1, x_2))(v_1 + v_2) - (c_1 x_1 + c_2 x_2) \tag{14}$$

As the specification of $F(x_1, x_2)$, Sandler and Hartley [20] and Varian [25] considers three regimes:

Total effort:	$F(x_1, x_2) = x_1 + x_2$;
Weakest link:	$F(x_1, x_2) = x_1 \wedge x_2$;
Best shot:	$F(x_1, x_2) = x_1 \vee x_2$;

Then the aim is

$$\max_{x_1, x_2} \left(p(F(x_1, x_2))(v_1 + v_2) - (c_1 x_1 + c_2 x_2) \right) \tag{15}$$

The constraint is to minimize the agents' cost. The Nash equilibrium solution depends upon the regime committed. Free-riding occurs under certain conditions. However, the functional form of $F(x_1, x_2)$ in [20] and [25] is oversimplified, if $F(x_1, x_2)$ is non-linear in x_1 and x_2, then the Nash-lambda algorithm needs to step in for searching numerical solutions.

4.4. Optimal maintenance services

In this subsection, we consider the problem of equipment maintenance by an external subcontractor. The owner of the equipment and the subcontractor are two players under non-corporative game. Both parts want the maximized profits [14]. This is a bi-level program and it is appropriate to use the Nash-lambda algorithm to search optimal solution.

5. Conclusion

In this paper, we investigate the merging with Nash equilibrium solution with lambda algorithm, which is a type of Bayesian network, [2], [3], [8], [17]. We have successfully created a merged algorithm and coded it in details, i.e. at bi-level program with two players. Frankly, the numerical example in Section 3 does not link to safety and reliability. However, just this example triggered our interest to investigate the merging and programming the new algorithm because the problem formulation is strictly revealing the requirements in *Definition 2.1*. To cope the spirit of the conference, we give a detailed reliability example in Subsection 4.1 for illustrations. In the future, we will strive to increase the number of the players first and then the 3-level, and so on. The application examples in this paper are not detailed because of the page limitation and time-constraints. We will improve the paper in this aspect.

Acknowledgements

This research is supported by the South African National Research Foundation (IFR2010042200062) and (IFR2009090800013), and by the University of Cape Town Postgraduate Fellowship 2011. Authors deeply thank Professor B.D. Liu and his PhD students: X.W. Chen, W. Dai, K. Yao, and Z. X. Peng for their invaluable debates and inputs.

References

[1] Liu, B. D. (1998). Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Computers Math. Applic.* 36, 7, 79-89.
[2] Ben-Gal, I. (2007). Bayesian Networks, in Ruggeri F., Faltin F. & Kenett R. *Encyclopedia of Statistics in Quality & Reliability*, Wiley & Sons.
[3] Buntine, W. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research* 2.
[4] Cui, Y. H., Guo, R. & Guo, D. (2009). A Naïve five-element string algorithm. *Journal of Software* 4, 9, 925-934
[5] Cui, Y. H, Guo, R., Dunne, T. & Guo, D. (2010). Lambda algorithm. *Journal of Uncertain Systems* 4, 1, 22-23.
[6] Cui, Y., Guo, R., Savsani, V., Rao, R. & Vakharia, D. (2008). Harmony element algorithm- A naïve initial searching Range. *Proc. of the International conference on 'Advances in mechanical engineering'*, 479-484.
[7] Heymann, P. B. (2003). Dealing with Terrorism after September 11, 2001: An Overview. In: Howitt, A., Pangi, R. (eds.) *Preparing for Domestic Terrorism*. MIT Press, Cambridge, MA, 57-72.
[8] http://en.wikipedia.org/wiki/Bayesian_network
[9] http://en.wikipedia.org/wiki/Bilevel_program
[10] http://en.wikipedia.org/wiki/Game_theory
[11] http://en.wikipedia.org/wiki/Multilevel_programming
[12] http://en.wikipedia.org/wiki/Nash_equilibrim
[13] http://en.wikipedia.org/wiki/Stackelberg_competition
[14] Jackson, C. & Pascual, R. (2008). Optimal maintenance service contract negotiation with aging equipments. *European Journal of Operational Research* 189, 387–398
[15] Kaplan, E.H., Mintz, A., Mishal, S. & Samban, C. (2005). What Happened to Suicide Bombings in Israel? Insights from a Terror Stock Model. *Studies in Conflict & Terrorism* 28, 225–235.
[16] Keohane, N.O, & Zeckhauser, R. (2003). The Ecology of Terror Defense. *Journal of Risk and Uncertainty* 26, 201–229.
[17] Kjrulff, U. (1992). A computational scheme for reasoning in dynamic probabilistic networks, Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, 121-129, Morgan Kaufmann, San Francisco.
[18] Manbachi, M., Mahdloo, F. & Haghifam, M-R. (2010). A New Solution for Maintenance Scheduling in Deregulated Environment based on Lost Opportunity Cost of Market Participation and Reliability. *Proc. of Modern Electric Power Systems* 2010, Wroclaw, Poland, MEPS'10 -paper P4.
[19] Masatoshi, S. & Ichiro, N. (2009). *Cooperative and Noncooperative Multi-Level Programming*. Springer.Dordrecht - Heidelberg - London - New York
[20] Sandler, T. & Hartley, K. (2001). Economics of alliances: The lessons for collective action. *Journal of Economic Literature* 34, 869-896.
[21] Sefrioui, M. & Perlaux, J. (2000). Nash genetic algorithms: examples and applications. *Proc. of the 2000 Congress on Evolutionary Computation* 1, 509-516.
[22] Simaan, M., & Cruz, J. B. (1973). A Stackelberg solution for games with many players. *IEEE transactions on Automatic Control* AC-18, 322-324.
[23] Simaan, M. & Cruz, J. B. (1973). On the Stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and App.* 11, 533-555.
[24] Spata, M. O. & Rinaudo, S. (2010). Merging Nash Equilibrium Solution with Genetic Algorithm: The Game Genetic Algorithm. *Journal of Convergence Information Technology* 5, 9, November 2010, 9-15.
[25] Varian, H. R. (1994). Sequential provision of public goods. *Journal of Public Economics* 53, 165-186.