Jurnal Metris, 15 (2014): 65 – 76

Jurnal
Metris
ISSN: 1411 - 3287

Soccer game optimization for continuous and discrete problem

Hindriyanto Dwi Purnomo

Department of Information Technology, Satya Wacana Christian University, 50711, Indonesia
Email: hindriyanto.purnomo@staff.uksw.edu

Received 1 March 2014; Accepted 1 September 2014

Abstract.

Soccer games optimization is a new metaheuristics method that mimics the soccer player's movement, wherein each player decides their best positions to dribble the ball towards the goal based on the ball position and other players' position. This paper discussed the method for continuous and discrete problems based on 'pair cooperation' between a player and the ball position. The algorithm is implemented in eight benchmark problems consisting of continuous unconstrained problems, continuous constrained problems and discrete problem. The performance of the algorithm for the continuous unconstrained problems is compared to two meta-heuristic algorithms, the genetic algorithm and the particle swarm optimization. The continuous constrained problems and the discrete problem are compared with the result in the literature. The experimental results show that the algorithm is a potentially powerful optimization procedure that can be applied for various optimization problems.

Keyword: *Inventory, Lost sales, Finite planning horizon*

1. INTRODUCTION

Meta-heuristic algorithms have been implemented to solve various optimization problems. These algorithms overcome the drawback of conventional, computational-based numerical linear and nonlinear programming methods in which gradient information is considered necessary. In many real problems, gradient search approaches could become very difficult and even unstable (Lee & Geem, 2005); therefore meta-heuristic algorithm eliminates the need for gradient information. The meta-heuristic algorithms employ randomness and set of rules to obtain an acceptable solution. They are frequently used when analytical solution is difficult to achieve. The algorithms provide a quicker and easier way of optimization problems. Due to the nature of heuristic method, meta-heuristic algorithms do not guarantee optimal solution; however, they typically produce acceptable solution.

Many of the existing meta-heuristic algorithms are inspired by natural phenomena. Some of these were compared from biological evolutionary: the genetic algorithm (Holland, 1975); from the animals' behavior: tabu search (Glover , 1998), ant colony optimization (Dorigo and Caro, 1999), and particle swarm optimization (Kennedy and Eberhart, 1995); from the physical processes: simulated annealing (Kirkpatrick et al, 1983); and from the music improvisation: harmony search (Geem, et al, 2001; Yang, 2009; Geem et al, 2005).

There are two important components of a meta-heuristic; intensification and diversification (Yang, 2009). Intensification is the ability to investigate the neighborhood of a potential solution. Intensification is important for improving the potential solution during the search because of its ability to find better solution near a potential solution. Diversification is the ability to explore the whole solution space and is important to avoid trapped in local optimal solution. In order to obtained high performance of metaheuristic, these two components should be laid in balance.

In this paper, a new meta-heuristic method called Soccer Game Optimization (SGO) is discussed. The algorithm mimics the soccer player's movement during the soccer game. The algorithm is implemented in continuous and discrete problems. The performance of the SGO is compared with genetic algorithm (GA) and particle swarm optimization (PSO) for the continuous unconstrained problems while its performance for the continuous constrained problems and the discrete problem are compared to previous researches reported in literatures. The rest of the

paper is organized as follows. Section 2 describes the background and the structure of the method. Section 3 describes the selected benchmark problems. Section 4 discusses the performance of the algorithm. Conclusion and future directions are presented in Section 5.

2. SOCCER GAME OPTIMIZATION

Soccer Game Optimization is proposed by Purnomo et al. (2012). The method is described using soccer player movement as its analogy. The method is a simplified a soccer game player's movements. Many terms used in the method are derived from the soccer game. The terms used in the method are:

1.	player's position	:	encodes a set of decision variables
2.	a player	:	a candidate solution
3.	a team	:	a simultaneous set of candidate solutions
4.	Kick	:	iteration
5.	ball	:	best solution so far
6.	soccer field	:	available search space
7.	soccer rules	:	constraints
8.	Goal	:	optimal solution

A player's position encodes a set of decision variables. The quality of a player is evaluated based on its advantageous position which represents its objective value. SGO is a population based method where a team consists of several players. The relationship between players, player's position, a team, and objective function is illustrated in figure 1.

During the search, a single kick will manipulate a simultaneous potential solution (a team). It is similar to the soccer games that all players move simultaneously at the same time. The performance of meta-heuristic algorithms is often assessed based on the number of objective function evaluation. For this reason, the number of kick is used to represent the number of iteration. The number of objective function evaluation for each kick depends on the team size. For example, if a team consists of 10 players, there are 10 objective function evaluations (each player is evaluated once) for each kick. Soccer field represent the available search space where the search is conducted. All players should be on the field. The soccer rules represent the problem constraints that define some prohibited conditions. The players should follow the rules. In constrained problems, not all solutions in the search space are feasible. The rules divide the solutions into feasible and infeasible solutions. The goal is the optimal solution to achieve. Unlike the soccer where the goal is on the edge of the field, the optimal solution of an optimization problem can be in any location in the search space. In addition, the method does not consider the opposite player's movements.

In order to ensure player's movement, two operations are introduced to manage the diversification and intensification. Diversification is controlled by random movement, called '*move off the ball*', to explore the search space. The movement minimizes the chance of premature convergence. Intensification is managed by performing movement based on the information sharing among players and between players and ball, called '*cooperation movement*'. *Cooperation movement* is characterized by the degree of information sharing, called *cooperation rate*. In a time, a player's movement can be a '*move off the ball'* or a '*cooperation movement'*.

The information sharing is divided into two types, local information and global information. Local information can only be accessed by other players nearby. Global information means all players can access the information. For example, ball position is accessible to all players. The general SGO's steps are: Problem and parameters initialization, initialize player's position and ball's position, player's movement, determine the new ball's position, termination criteria. The flowchart of SGO is given in Figure 2.

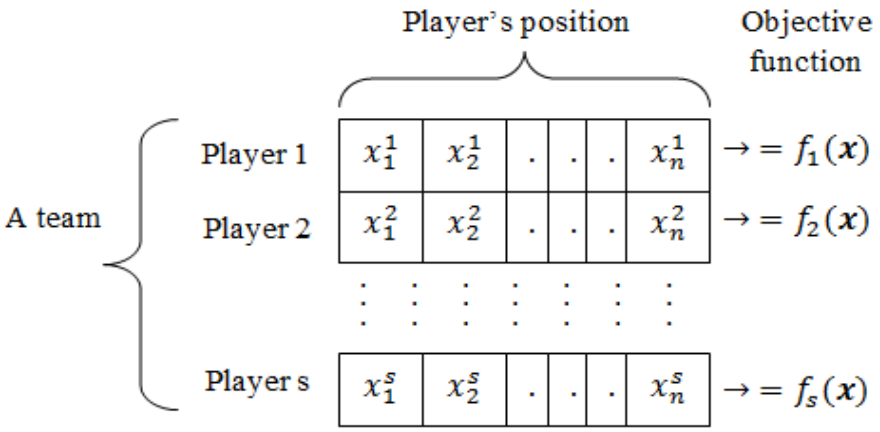

Figure 1. Players, player's position, a team and objective function

Figure 2. The flowchart of SGO

Due to the different characteristics of continuous and discrete problems, the approached for both problems are different. Therefore, the explanations of the method for continuous and discrete problems are discussed separately.

2.1 Continuous problem

2.1.1 Parameter initialization for continuous problem

In the first step, the optimization problem can be expressed as:

$$\min\ f(\boldsymbol{x}) \tag{1}$$

$$\text{s.t.}\quad \boldsymbol{x} = \{x_{1,t}, x_{2,t}, \dots, x_{n,t}\}$$

where $f(.)$ is the objective function, $x_{i,t}$ is the value of variable i at time t. For most continuous optimization problems, prior knowledge of the search space is known. Therefore, for a known search space of variable i [a,b], $a \le x_{i,t} \le b$ for $t \ge 0$. The algorithm parameters that need to be initialized are *move off the ball 'm', cooperation rate* '*w*', team size *'s'* and termination criteria.

2.1.2 Initialize player's position and ball's position for continuous problem

In this step, each initial player position is generated randomly. It can be written as:

$$\boldsymbol{x_0^j} = random\ (x_{1,0}^j, x_{2,0}^j, \dots, x_{n,0}^j) \tag{2}$$

$\boldsymbol{x_0^j}$ is the position of player j at time 0. After all players' positions are initialized, their positions are then evaluated using the objective function. A player has the most advantageous position if it has the smallest value in the team. As a player with the most advantageous position will dribble the ball, the initial ball position $BP_0(\boldsymbol{x})$ is formulated as:

$$\boldsymbol{B_0}(\boldsymbol{x_0^b}) = min\left(f_0^1(\boldsymbol{x_0^1}), f_0^2(\boldsymbol{x_0^2}), \dots, f_0^j(\boldsymbol{x_0^j}), \dots, f_0^s(\boldsymbol{x_0^s})\right) \tag{3}$$

$\boldsymbol{B_0}(\boldsymbol{x_0^b})$ is the ball's at time 0. The ball's position is defined by a set of solution vector $\boldsymbol{x_0^b}$. s is the team size and $f_0^j(\boldsymbol{x_0^j})$ is the objective value of player j at time *0*.

2.1.3 Player movement for continuous problems

Player movement is the most important step in the SGO. Two important mechanisms are presented for

a player movement: the '*move off the ball'* and the '*cooperation movement'* among players. *Move off the ball* is a global search where a player explores the entire search space. *Cooperation movement* is a local search where a player exploits the search space nearby and is determine by its *cooperation rate*. Each player movement is described as:

2.1.3.1 Move off the ball

By a probability of *m*, a player will move randomly to explore the search space.

$$\boldsymbol{x_t^j} = random\ (x_{1,t}, x_{2,t}, \dots, x_{n,t})$$

$\boldsymbol{x_t^j}$ is the player's position at time *t.*

2.1.3.2 Cooperation movement

By a probability of *1-m*, a player will run towards the position of the ball. The main consideration of the cooperation mechanism is the relationship between two options and the number of option in the search space. In continuous problems, the relationship between options is known and there are infinite options in the search space. Figure 3 is an example of continuous problem with single variable *i*. In the figure, the solution space is limited by [a, b]. For two known player position $x_{i,t}^1$ and $x_{i,t}^2$, their relation is known $\left(x_{i,t}^1 < x_{i,t}^2\right)$. The number of available option between *a* and *b* is infinite since a player can be in any position between *a* and *b*.

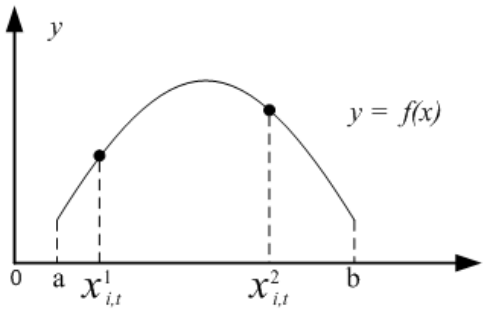

Figure 3. Example of continuous problem

The main idea of *cooperation movement* for continuous problems is exploiting the search space among players using the centroid principle. The new player's position is determined by players nearby and the ball. The equation is expressed as:

$$x_{i,t}^j = w_{i,t}^j . x_{i,t-1}^j + \frac{1}{z} \sum_{k \in A^j, k \in A^g} w_{i,t}^k \cdot x_{i,t-1}^k \qquad (4)$$

for $i = 1,2, \dots, n$

Where i is decision variable, A^j is a set of local players that affecting player j, A^g is a set of global players and the ball, k is member of A^j and A^g, z is the accumulative number of member of A^j and A^g, $w_{i,t}^k$ is cooperation rate of k at time t and $x_{i,t-1}^j$ is the previous player j position.

To illustrate the equation above, we use the following assumption: A player only influenced by its previous position and the ball position. Therefore A^j consists of no player and A^g consists of the ball position. As the member of A^j is zero and the member of A^g is 1, the value of z is 1. Using the assumption, equation 4 will become:

$$x_{i,t}^j = w_{i,t}^j . x_{i,t-1}^j + w_{i,t}^b . x_{i,t-1}^b \qquad (5)$$

where $x_{i,t-1}^b$ is the ball position at time *t-1*, $w_{i,t}^j$ is cooperation rate of player j at time t and $w_{i,t}^b$ is cooperation rate of the ball at time *t*. As only two positions are considered, the search space of each variable can be illustrated as a line. The search space can be divided in three areas: I, II and III, as shown in Figure 4.

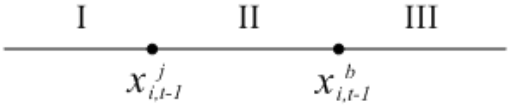

Figure 4. Search area,two agents are considered

In order to avoid explosion during the search, it is important to verify the search area. As for each variable, the previous player's value $x_{i,t-1}^j$ and the previous the ball's value $x_{i,t-1}^b$ are known (obtained in previous kick) and their relationship are known (for example $x_{i,t-1}^j \leq x_{i,t-1}^b$), then $w_{i,t}^j$ and $w_{i,t}^b$ become dependent control factors if the search space is specified. The relationships between these two controls variables for $x_{i,t-1}^j \leq x_{i,t-1}^b$ are formulated as (see Appendix A):

$$w_{i,t}^b < \frac{\left(1 - w_{i,t}^j\right) x_{i,t-1}^j}{x_{i,t-1}^b} \qquad \text{If } x_{i,t}^j < x_{i,t-1}^j \qquad (6.a)$$

$$\frac{\left(1 - w_{i,t}^j\right) x_{i,t-1}^j}{x_{i,t-1}^b} \leq w_{i,t}^b \leq 1 - \frac{w_{i,t}^j . x_{i,t-1}^j}{x_{i,t-1}^b} \qquad \text{If } x_{i,t-1}^j \leq x_{i,t-1}^j \leq x_{i,t-1}^b \qquad (6.b)$$

$$w_{i,t}^b > 1 - \frac{w_{i,t}^j . x_{i,t-1}^j}{x_{i,t-1}^b} \qquad \text{If } x_{i,t}^j > x_{i,t-1}^b \qquad (6.c)$$

In Figure 4, using the centroid principle, area II is the most feasible search space, as its boundaries are better known than the other areas. For this area, we could derive a simplification of eq. 6.b (see Appendix B):

$$w_{i,t}^j + w_{i,t}^b = 1 \qquad (7)$$

2.1.4 Determine the new ball position

After all player moves to the new position, the players are evaluated. The candidate for the next ball dribbler is determined based on the best player's position.

$$\boldsymbol{C_t}(\boldsymbol{x_t^c}) = min\left(f_t^1(\boldsymbol{x_t^1}), f_t^2(\boldsymbol{x_t^2}), \dots, f_t^s(\boldsymbol{x_t^s})\right) \quad (8)$$

s.t.

$$\boldsymbol{x_t^c} \in \{\boldsymbol{x_t^1}, \boldsymbol{x_t^2}, \dots, \boldsymbol{x_t^s}\}$$

$\boldsymbol{C_t}(\boldsymbol{x_t^c})$ is the candidate dribbler. Its position is defined by a set of solution vector $\boldsymbol{x_t^c}$.

If the current candidate of ball dribbler $\boldsymbol{C_t}$ is better than the ball position, the ball will then be passed to that current best player. Otherwise, the ball is dribbled by the current ball dribbler.

$$\boldsymbol{BP_t}(\boldsymbol{x_t^b}) = min\left(\boldsymbol{C_t}(\boldsymbol{x_t^c}), \boldsymbol{BP_{t-1}}(\boldsymbol{x_{t-1}^b})\right) \quad (9)$$

s.t.

$$\boldsymbol{x_t^b} \in \{\boldsymbol{x_t^c}, \boldsymbol{x_{t-1}^b}\}$$

2.1.5 Termination Criteria

Termination criteria can be based on the maximum number of kick, a small number of error ε or when the kick does not provide any improvement after several kicks. If the termination criterion is reached, kick is stopped. Otherwise steps 2.1.3 and 2.1.4 are repeated. Figure 5 illustrates the work of method for continuous problems when one player is affected by it previous position and the ball position.

2.2 Discrete problem

2.2.1 Initialization parameter for discrete problem

Initialization step for discrete problems is similar to the continuous problems. The optimization problem is the same as equation 1.

$$\min\ f(\boldsymbol{x})$$

s.t.

$$\boldsymbol{x} = \{x_{1,t}, x_{2,t}, \dots, x_{n,t}\}$$

However, when the variables are discrete, the number of option for each variable is finite. Therefore $x_{i,t} \in \boldsymbol{X_i}$, where $\boldsymbol{X_i}$ is a set of available options for variable *i*, that is $\boldsymbol{X_i} = \{x_i(1), x_i(2), \dots, x_i(o_i)\}$. The other parameters that need to be initialized are *move off the ball 'm', cooperation rate* '*w*', team size *'s'* and termination criteria.

2.2.2 Player's position and ball dribbler initialization

This is similar to 2.1.2. Each player's position is initialized randomly using eq 2.

$$\boldsymbol{x_0^j} = random\ (x_{1,0}^j, x_{2,0}^j, \dots, x_{n,0}^j)$$

since the number of options are finite, $x_i^j \in X_i$. After all player's positions are initialized, each player is evaluated based on the objective function. The best player will dribble the ball as expressed as:

$$\boldsymbol{B_0}(\boldsymbol{x_0^b}) = min\left(f_0^1(\boldsymbol{x_0^1}), f_0^2(\boldsymbol{x_0^2}), \dots, f_0^j(\boldsymbol{x_0^j}), \dots, f_0^s(\boldsymbol{x_0^s})\right)$$

2.2.3 Player movement for discrete problem

Move off the ball

By a probability of *m*, a player *j* will move randomly to explore the search space.

$$\boldsymbol{x_t^j} = random\ (x_{1,t}, x_{2,t}, \dots, x_{n,t})$$

Cooperation movement

In discrete problems, the relationship between finite options in the search space is not known necessarily. For example, two known options of variable *i* $x_{i,t}^j(1)$ and $x_{i,t}^j(2)$ does not necessarily fulfill the relationship:

$x_{i,t}^j(1) \leq x_{i,t}^j(2)$ or $x_{i,t}^j(1) \geq x_{i,t}^j(2)$.

The cooperation movement in this research is based on the weight that represent the contribution of the option to the output. The *cooperation rate w* is used as the weight of each option; as a consequence, each option has its own cooperation rate. This differs from the continuous problems where *w* is assigned to a variable not an option. The *w* is shared among players and between player and the ball. Every time a player moves into a new position, the value of *w* for each selected option is adjusted. Each *w* of the selected option in a player will be added if the player position is better than the ball position, otherwise it is reduced. The changes of *w* for selected option in player *j* are influenced by players or ball in A^j and A^g. To illustrate how to determine w, we used the same assumption as in the continuous problem: a player position is determined by its previous position and the ball position. In order to avoid wild fluctuation of w, we limit its value as $0 \leq w \leq 1$. Since $f_t(\boldsymbol{x})$ is the minimization function, $f_t(\boldsymbol{x})$ is better than $B_t(\boldsymbol{x})$ if $f_t(\boldsymbol{x}) < B_t(\boldsymbol{x})$. An option's weight is updated as follow:

$$w_{i,so_{i,t},t}^j = \begin{cases} w_{i,so_{i,t},t-1}^j + \left(1 - w_{i,so_{i,t},t-1}^j\right) * w_{i,so_{i,t},t-1}^b & if \quad f_t^j < BP_{(t-1)} \\ w_{i,so_{i,t},t-1}^j - w_{i,so_{i,t},t-1}^j * w_{i,so_{i,t},t-1}^b & if \quad otherwise \end{cases} \quad (10)$$

where $w_{i,so_t,t-1}^j$ is the weight of selected option *so* for variable *i* at time *t* for player *j* at time *t-1*. In other words, the weight *w* of the selected option at time *t* is updated based on the *w* of the same option in previous state (even though the option is not the selected option in at *t-1*).

Adjusting all related $w_{i,so_{i,t},t}^j$, the player position is updated based on the value of $w_{i,so_{i,t},t}^j$ for $i = 1,2, \dots, n$. We use centroid principle to select an option of each variable. For each variable *i*, an option with the highest value of weight is assumed as the center of the option and is decided as the selected option (see Figure 6).

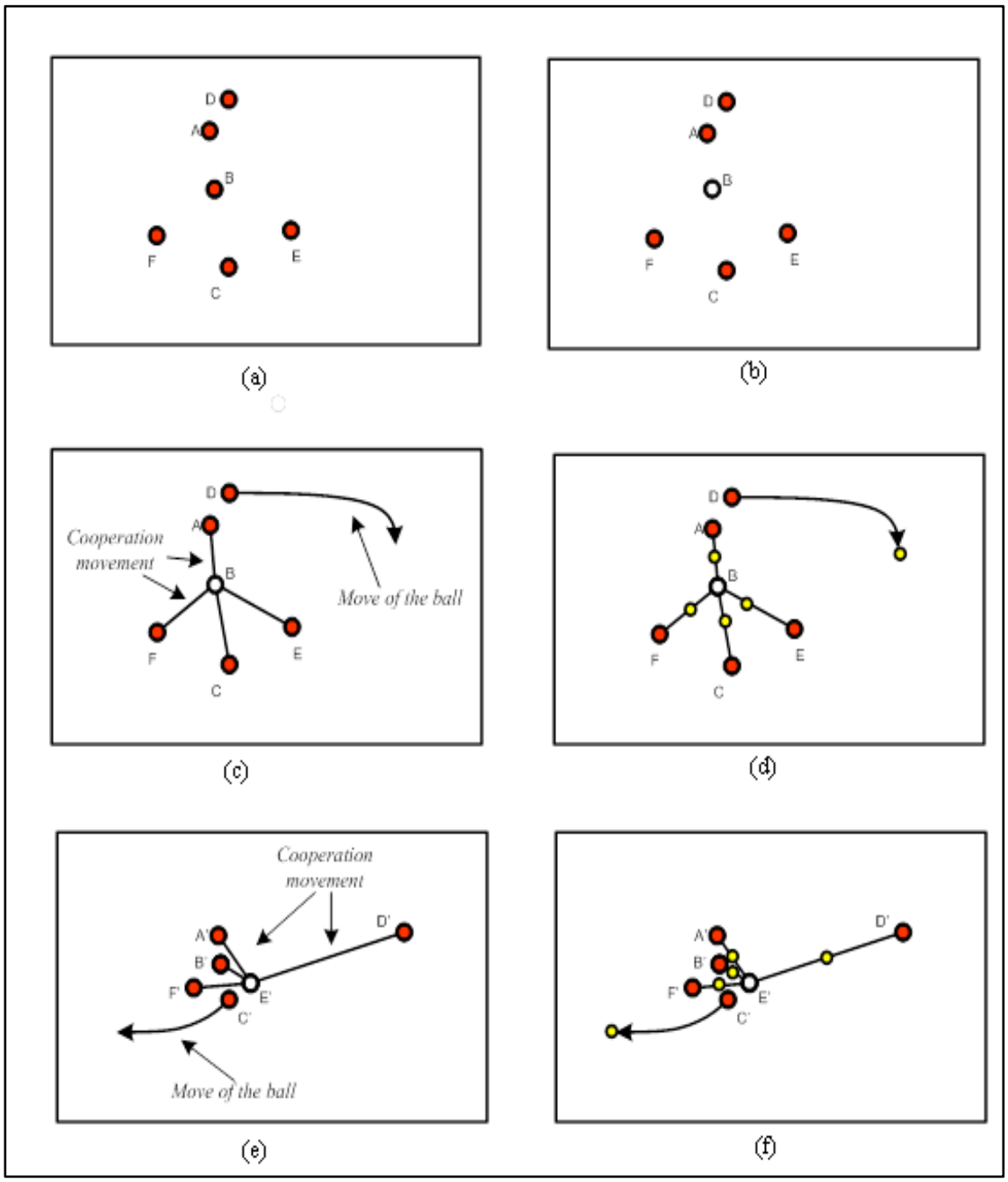

Figure 5. Illustration of the SGO

The new selected option can be written as:

$$_{,t} = x_i(so_{i,t}) \quad (11)$$

s.t.

$$(so_{i,t}) \in \boldsymbol{X_i}$$

$$_{i,so_{i,t},t} = \max(w_{i,1,t}, w_{i,2,t}, \dots, w_{i,o_i,t})$$

A new player *j* position can be written as:

$$\boldsymbol{x_t^j} = \{x_1^j(so_{1,t}), x_2^j(so_{2,t}), \dots, x_n^j(so_{n,t})\}$$

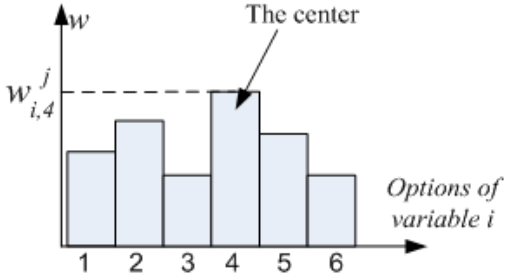

Figure 6. An illustration to select an option of a variable

2.2.4 Determine the new ball position

After all players move to the new positions, the players are evaluated using the objective function. The candidate for the next ball dribbler is determined based on the best new position of the players as expressed in eq 8.

$$\boldsymbol{C_t}(\boldsymbol{x_t^c}) = min\left(f_t^1(\boldsymbol{x_t^1}), f_t^2(\boldsymbol{x_t^2}), \dots, f_t^s(\boldsymbol{x_t^s})\right)$$

s.t.

$$\boldsymbol{x_t^c} \in \{\boldsymbol{x_t^1}, \boldsymbol{x_t^2}, \dots, \boldsymbol{x_t^s}\}$$

The new ball dribbler position is determined using eq 9.

$$\boldsymbol{BP_t}(\boldsymbol{x_t^b}) = min\left(\boldsymbol{C_t}(\boldsymbol{x_t^c}), \boldsymbol{BP_{t-1}}(\boldsymbol{x_{t-1}^b})\right)$$

2.2.5 Termination criteria

Termination criteria can be based on the maximum number of kick, a small number of error ε or when the kick does not provide any improvement after several kicks. If the termination criterion is reached, kick is stopped. Otherwise steps 2.2.3 and 2.2.4 are repeated.

3. NUMERICAL EXAMPLE

In this paper, 5 continuous unconstrained problems, 2 continuous constrained problems and 1 discrete problem are used to evaluate the performance of the SGO.

a. Six hump camel back function

Six hump camel back function is a function for global optimization testing. This function has four local minima and two global minima (Lee eta al, 2007). The function is formulated as:

$$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \quad (12)$$

The two global minima for this function are: $f(0.089842, -0.712656) = -1.031628453$ and $f(-0.089842, 0.712656) = -1.031628453$.

b. Rosenbrock's function

Rosenbrock function, also known as banana function, exhibits a global optimum near a long, narrow, parabolic-shaped flat valley (Rosenbrock , 1960). This function is often used for performance assessment since it is difficult to converge to the global optimum. In this study, two dimension of Rosenbrock function is used. The function is formulated as

$$f(x) = \sum_{i=1}^{n-1}\left(100.\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right) \quad (13)$$

The global minimum for this function is: $f(x) = 0$, $x_i = 1,\ i = 1, 2, .., n$

c. Rastrigin's functions

This function was first proposed by Rastrigin (Torn and Zilinskas, 1989). The function was extended to allow for the increase of the number of variables. The generalized Rastrigin is a nonlinear, multi-dimensional function with several local minima but exhibits only one global minimum (Saez et al, 2005). This function is highly modulated and is frequently used performance measurement. In this study, two dimension of the Rastrigin function is used. The function, in this paper, is formulated as:

$$f(x) = 10.n + \sum_{i=1}^{n}\left(x_i^2 - 10.cos(2.\pi.x_i)\right) \quad (14)$$

The global minimum for this function is: $f(x) = 0$, $x_i = 0,\ i = 1, 2, .., n$

d. Wood function

Wood function is a fourth-degree polynomial function (Lee and Geem 2005). This function is formulated as:

$$f(x) = 100.(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90.(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1) \quad (15)$$

The global minimum for this function is: $f(x) = 0$, $x(i) = 1,\ i = 1,2,3,4$

e. Goldstein and Price function-1

Goldstein and Price function-1 is an eight-degree polynomial function with 2 variables. This function has three local minima and one global minimum (Lee and Geem 2005). The function is formulated as:

$$f(x) = \{1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\} \times \{30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\} \quad (16)$$

The global minimum for this function is: $f(0, -1) = 3$.

f. Constrained function 1

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \quad (17)$$

Subject to

$$c_1(x) = x_1 - 2x_2 + 1 = 0$$

$$c_2(x) = -\frac{x_1}{4_1} - x_2^2 + 1 \geq 0$$

$$-10 \leq x_1 \leq 10\ ,\ -10 \leq x_2 \leq 10$$

The problem originally introduced by (Braken and MacCormick , 1968). It has the optimum solution at $f(0.82288, 0.91144) = 1.3935$.

g. Constrained function II

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \quad (18)$$

Subject to

$$c_1(x) = 4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \geq 0$$

$$c_2(x) = x_1^2 + (x_2 - 2.5)^2 - 1.84 \geq 0$$

$$0 \leq x_1 \leq 6\ ,\ 0 \leq x_2 \leq 6$$

The problem is found in (Deb , 2000). The unconstrained objective function has the optimal solution at $f(3,2) = 0$. Due to the constraint, the minimum solution is located at $f(2.246826,2.381865) = 13.59085$.

h. Discrete problem : School bus routing problem

School bus routing problem is multi-objective problem to minimize the number of operating bus and the travel time due to bus capacity and time window. This problem has two constraints: time constraint and capacity constraint. The diagram of school bus routing is illustrated in Figure 7.

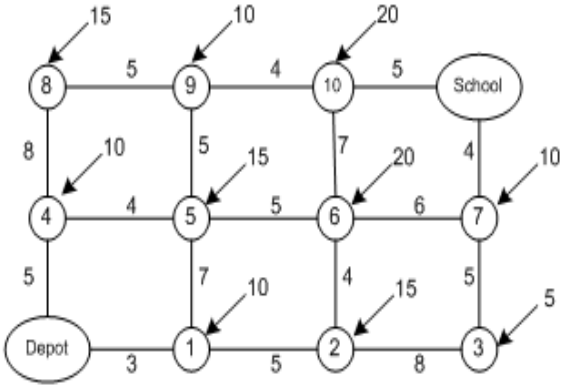

Figure 7. Diagram of school bus routing network (Geem et al, 2005)

The routing problem parameters are specified as: fixed cost for each operating bus (*fc*) is \$100,000/bus; routing cost (*rc*) is \$ 105/min; shortest path between two nodes is calculated by Floyd and Warshall's algorithm; penalty cost for capacity constraint is \$100,000 when any bus carries more than 45 students; penalty cost for time constraint is \$100,000 when any bus operates more than 32 minutes; boarding time is 6 second per students and the number of buses are 4. The problem model is described in (Geem et al, 2005).

4. EXPERIMENTAL RESULTS AND DISCUSSION

For continuous unconstrained problems, two existing algorithms, the GA and the PSO, are used to compare the performance of the SGO. GA is selected since it performs satisfactorily for many types of optimization problems (Silva and Sousa, 2008). Likewise, PSO is selected as this method has been considered powerful and has very good performance for continuous types of problems. While for the continuous constrained problems and the discrete problem, the performance of the SGO is compared with the result reported in literatures.

4.1 Continuous problems

In order to reasonably compare the three methods for continuous constrained problems, the number of times for objective function evaluation and the number of computational runs are made the same for all the methods. The population sizes are 10, numbers of different runs are 50, the random values are bounded between -5.0 and 5.0, and the number of kick for the problems is the same for all the algorithms. The SGO used the following assumption: a player is only affected by its previous position and the ball position, golden ratio is used to set the cooperation rate $wb = 0.618$, $wp = (1 - wb) = 0.382$, and the probability that a player will move randomly (m) is 0.1. The crossover and mutation rate for the GA are set as follows: crossover rate is 0.8 and mutation rate is 0.2. The GA toolbox in Matlab 7 is used since it is well designed and is widely accepted as a good optimization tool. The PSO is implemented using PSO toolbox-beta-0.3 developed by Jagatpreet Singh (Khosla et al, 2006). For the PSO algorithm, two types of adjustments are made in terms of the number of a particle neighborhood. Neighborhood sizes of 0 and 2 are used for the PSO 1 and the PSO 2 respectively but the other parameters are the same. They are set to: c1 = 2, c2 = 2, c3 = 1, $\omega_{start} = 0.95$, $\omega_{end} = 0.4$, $\omega_{varyfor} = 0.7$, $v_{max} = 100$. The method is implemented using Matlab 7.0. Table 1 provides the experimental results for the unconstrained continuous problems.

In this paper, the constrained problems are reformulated into unconstrained problem by applying penalty. Static penalty is used.

$$f_{new}(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{q=1}^{q_n} \alpha . v_q + \beta . g(\boldsymbol{x}) \qquad (19)$$

$$v_q = \begin{cases} 1 & if\ constraint\ q\ is\ violated \\ 0 & otherwise \end{cases}$$

where $f_{new}(\boldsymbol{x})$ is the new objective function, q_n is the number of constraints, α and β are the penalty coefficient, $g(\boldsymbol{x})$ is the degree of the constraint violation. Using the formulation above, we use the information about the number of constraints violated and the distance between the violations to the feasible solution. This has been prove to be very effective (Dasgupta and Michalewica, 1997; Coello, 2000; Coello, 1997). The number of kick for constrained function I and II is 1000. The comparison result for constrained problems is given in Table 2.

Table 1. Computational results for unconstrained problems

No	Algorithm	# it		PSO 1	PSO 2	GA	SGO
1	Rastrigin	1000	best	0	0	6.12E-07	0
			avg	0	0	6.47E-04	2.13E-16
			se	0	0	1.03E-03	1.11E-15
2	Rosenbrock	1000	best	1.52E-10	9.32E-14	1.35E-07	3.17E-17
			avg	1.52E-10	9.32E-14	4.21E-04	1.65E-07
			se	5.22E-26	6.37E-29	6.2E-04	3.36E-07
3	Six hump camel back function	1000	best	-1.031628453	-1.031628453	-1.0316274	-1.031628453
			avg	-1.031628453	-1.031628453	-1.0315	-1.031628453
			se	6.73E-16	6.73E-16	1.7E-04	6.73E-16
4	Wood	5000	best	0.106688	0.023393	8.43E-05	6.07E-06
			avg	0.106688	0.023393	9.94E-02	2.E-02
			se	0	0	9.E-02	5.E-02
5	Goldstein and Price function-1	1000	best	3.000000000	3.000000000	3.000011	3.000000000
			avg	3.000000000	3.000000000	3.001	3.000000000
			se	1.35E-15	3.59E-15	1E-03	5.51E-15

Table 2. Comparison of the constrained function I and constrained function II

Problems	Methods	Optimal design variables		Constraints		Objective function value
		x_1	x_2	c_1	c_2	
Constrained function I	Homaifar et al. 1994	0.8080	0.8854	$3.7x10^{-2}$	$5.2x10^{-2}$	1.4339
	Fogel, 1995	0.8350	0.9125	$1.0x10^{-2}$	$7.0x10^{-2}$	1.3772
	Lee et al., 2005	0.8343	0.9121	$5.0x10^{-3}$	$5.4x10^{-3}$	1.3770
	This paper	0.82288	0.91144	$2.7x10^{-13}$	$1.1x10^{-11}$	1.3935
	Optimal solution	0.82288	0.91144	$7.05x10^{-9}$	$1.73x10^{-8}$	1.3935
Constrained function II	Deb, 2000 GA					
	- PS (R= 0.01)	-	-	-	-	13.58958
	- PS (R = 1)	-	-	-	-	13.59108
	- TS-R	-	-	-	-	13.59085
	Lee et al, 2005	2.246840	2.382136	-	-	13.59084
	This paper	2.246825	2.381842	$1.2x10^{-15}$	$2.2x10^{-2}$	13.59084
	Optimal solution	2.246826	2.381865	-	-	13.59085

4.2 Discrete problems

The performance of the SGO is compared with the performance of harmony search and the genetic algorithm reported in (Geem et al, 2005). The problem is set the same as in (Geem et al, 2005). In order to fairly compare with the existing literature, the number of objective function evaluation is set as 1000 (team size x number of kick) the same as mentioned in the literature. The algorithm is run for 20 times with different probability of *move off the ball, m*.

5. DISCUSSION

The computational results for the continuous unconstrained problems show that the SGO clearly outperforms the GA and is comparable to PSO 1 and PSO 2. The SGO's best and average values shows that the algorithm produces better accuracy (low bias), and the standard error values have better precision (low variance) than the GA in all the five benchmark problems. Comparing the SGO to PSO 1 and PSO 2, the SGO produces better accuracy and precision in the Wood function. It has the same performance in the Six-hump camel back function and in the Goldstein and Price function-1, and produces lower precision in the other benchmark functions.

It is interesting to notice that the SGO produced the same or better result than the other three methods in term of the best value achieved during the 50 replications. The SGO's average value and standard error clearly indicate that the SGO solution range always covers the optimal solution. This signifies that the SGO has a good diversification method. On the other hand, the PSO

1 and PSO 2 do not always produce the solution range that covers the optimal solution. For example, the Wood function problem in PSO 2. As the standard error of PSO 1 and PSO 2 are very small (almost zero), using 95% confidence level, the solution range is $\bar{s} \pm 2 * se(\bar{s}) = 0.023393 \pm 0 = (0.02339, 0.023393)$. The optimal solution for the problem ($f(x^*) = 0$) is not covered in the solution range and adding the confidence level does not change the range of the solution. This problem may occur due to premature convergence.

The experiment for the continuous constrained problems also produces better result than the previous result reported in the literatures. In the previous research, constrained function I is solved using genetic algorithm (Homaifar and Qi, 1994; Fogel, 1995), evolutionary algorithm (Fogel, 1995) and harmony search (Lee and Geem, 2005). Constrained function II is solved using GA-based method (Deb, 2000) and harmony search (Lee and Geem, 2005). The GA-based method implemented Powell and Skolnick constraint handling method (PS method) and tournament selection (TS) (Deb, 2000). The results of continuous unconstrained and constrained problem reveals that the SGO can be used for any continuous optimization problems.

The experiment result for the discrete problem show that the SGO could reach the global optimum twice out of 20 different runs, the average cost is $ 400,889 and standard error is $31,319. The result is comparable to the literature (Geem et al, 2005) where the harmony search reached the global optimum twice and the GA reached it once out of 20 different runs. The average costs are $399,870 in harmony search and $409,597 in GA. The standard errors for both algorithms are not mentioned in the literature (Geem et al, 2005).

The SGO only considers pair *cooperation* between a player and the ball position as its information sharing mechanism. The experiment results show that the method works well for continuous or discrete problems. Based on the experiments, we infer that the SGO have the potential to become a powerful optimization technique and can be applied in various engineering optimization problems. The issues of complex information sharing and considering dynamic optimization problems are some extension for consideration in future research.

In this study, we propose a method to solve the lost sales inventory problem and finite planning horizon. From the analysis and numerical example, we can conclude that the minimum total cost of the finite planning horizon method is always greater or equal to the infinite planning horizon method. Future research can be done to consider multi items lost sales in finite planning horizon.

6. CONCLUSIONS AND FUTURE WORK

This paper describes a new optimization algorithm based on soccer game analogy. The method is considered for continuous and discrete problems. The performance of the algorithm is assessed using eight benchmark problems for continuous unconstrained problems, continuous constrained problem and discrete problems. The results for the continuous unconstrained problems are compared with PSO 1, PSO 2 and GA while the result of the continuous constrained problems and the discrete problem are compared with the researches in existing literature. The computational result has shown that the algorithm performs better than existing researches as well as providing evidence of better diversification mechanism in the continuous problems. This study reveals that the SGO is potentially a powerful optimization technique and can be applied for other problems as well. For future research, we will consider dynamic environment and information sharing mechanism among the players.

Appendix A:

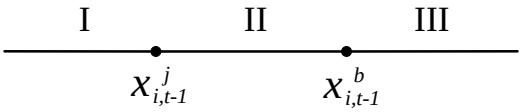

Figure A.1. Search area when only the previous player's position and the ball position are considered

Consider variable i,

$$x^j_{i,t} = w^j_{i,t}.x^j_{i,t-1} + w^b_{i,t}.x^b_{i,t-1}$$

The relation between x^j_{t-1} and $x^b_{i,t-1}$ is

$$x^j_{i,t-1} \leq x^b_{i,t-1}$$

Area I :

$$x^j_{i,t} < x^j_{i,t-1}$$
$$w^j_{i,t}.x^j_{i,t-1} + w^b_{i,t}.x^b_{i,t-1} < x^j_{i,t-1}$$
$$w^b_{i,t}.x^b_{i,t-1} < (1 - w^j_{i,t})\, x^j_{i,t-1}$$
$$w^b_{i,t} < \frac{(1 - w^j_{i,t})\, x^j_{i,t-1}}{x^b_{i,t-1}}$$

Area II :

$$x^j_{i,t-1} \leq x^j_{i,t} \leq x^b_{i,t-1}$$
$$x^j_{i,t-1} \leq w^j_{i,t}.x^j_{i,t-1} + w^b_{i,t}.x^b_{i,t-1} \leq x^b_{i,t-1}$$
$$(1 - w^j_{i,t})\, x^j_{i,t-1} \leq w^b_{i,t}.x^b_{i,t-1} \leq x^b_{i,t-1} - w^j_{i,t}.\, x^j_{i,t-1}$$
$$\frac{(1 - w^j_{i,t})\, x^j_{i,t-1}}{x^b_{i,t-1}} \leq w^b_{i,t}. \leq \frac{x^b_{i,t-1} - w^j_{i,t}.\, x^j_{i,t-1}}{x^b_{i,t-1}}$$
$$\frac{(1 - w^j_{i,t})\, x^j_{i,t-1}}{x^b_{i,t-1}} \leq w^b_{i,t}. \leq 1 - \frac{w^j_{i,t}.\, x^j_{i,t-1}}{x^b_{i,t-1}}$$

Area III :

$$x_{i,t}^{j} > x_{i,t-1}^{b}$$
$$w_{i,t}^{j}.\, x_{i,t-1}^{j} + w_{i,t}^{b}.x_{i,t-1}^{b} > x_{i,t-1}^{b}$$
$$w_{i,t}^{b}.x_{i,t-1}^{b} > x_{i,t-1}^{b} - w_{i,t}^{j}.\, x_{i,t-1}^{j}$$
$$w_{i,t}^{b} > \frac{x_{i,t-1}^{b} - w_{i,t}^{j}.\, x_{i,t-1}^{j}}{x_{i,t-1}^{b}}$$
$$w_{i,t}^{b} > 1 - \frac{w_{i,t}^{j}.\, x_{i,t-1}^{j}}{x_{i,t-1}^{b}}$$

Appendix B:

For area II, there are two extreme search spaces. The maximum search space (∞) occurs when $x_{i,t-1}^{j} = -\infty$ and $x_{i,t-1}^{b} = \infty$ and the minimum search space (0) occurs when $x_{i,t-1}^{j} = x_{i,t-1}^{b}$. The maximum search space does not provide any clue to do the search, so we only consider the minimum search space.

When $x_{i,t}^{j} = x_{i,t-1}^{b}$

$$\frac{x_{i,t}^{j}}{x_{i,t-1}^{b}} = 1,$$

then,

$$\frac{(1 - w_{i,t}^{j})\, x_{i,t-1}^{j}}{x_{i,t-1}^{b}} \leq w_{i,t}^{b}. \leq 1 - \frac{w_{i,t}^{j}.\, x_{i,t-1}^{j}}{x_{i,t-1}^{b}}$$

Become

$$1 - w_{i,t}^{j} \leq w_{i,t}^{b} \leq 1 - w_{i,t}^{j}$$
$$1 \leq w_{i,t}^{b} + w_{i,t}^{j} \leq 1$$
$$w_{i,t}^{j} + w_{i,t}^{b} = 1$$

6. REFERENCES

1. Braken, J., and MacCormick, G.P., Selected Applications of Nonlinear programming, John Wiley & Sons, New York, 1968.
2. Coello Coello C.A., Use of a Self-adaptive Penalty Approach for Engineering Optimization Problems, Computer in Industry, 41, (2000), pp. 113-127.
3. Coello Coello C.A., and Christiansen A.D., A Simple Genetic Algorithm for the Design of Reinforced Concrete Beams, Engineering with Computers 13(4), (1997), pp. 185-196.
4. Dasgupta, D. and Michalewica Z., (Eds), Evolutionary Algorithms in Engineering Applications, Springer-Verlag, Berlin, 1997.
5. Deb, K., An Effective Constraint Handling Method for Genetic Algorithms, Comput. Methods Appl Mech Engrg, 186, 2000, 311-338.
6. Dorigo, M., and Caro, G.D., Ant colony optimization: A new meta-heuristic, Congress on evolutionary computation (1999) 1470-1477.
7. Fogel, D.B., A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems, *Simulation*, vol. 64, pp. 397-404, 1995.
8. Geem, Z.W., Kim, J.H. and Loganathan, G.V., A New Heuristic Optimization Algorithm: Harmony Search, Simulation. 76 (2)(2001),60-68.
9. Geem, Z.W., Lee, K.S., and Park, Y., Application of Harmony Search to Vehicle Routing, American Journal of Applied Science 2 (12) (2005) 1552-1557.
10. Glover, F., Tabu Search - Wellsprings and Challenges, European Journal of Operational Research. 106 (1998) 221-225.
11. Holland, J. H., Adaptation in natural and artificial systems. Ann Arbor: MI, 1975.
12. Homaifar, A., and Qi , C.X., Constrained optimization via genetic algorithms, *Simulation*, vol. 62, pp. 242-254, 1994.
13. Kennedy, J. and Eberhart, R. Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks. 4 (1995) 1942-1948.
14. Kirkpatrick S., Gelatt, C.D., and Vecchi, M.P., Optimization by Simulated Annealing, Science. 220 (4598) (1983) 671-680.
15. Khosla, A., Kumar, S., Aggarwall, A.A., and Singh, J., A Matlab Implementation of Swarm Intelligence based Methodology for Identification of Optimized Fuzzy Models, in N. Nedjah, L. M. Mourelle (Eds) Swarm Intelligent Systems, Heidelberg: Springer-Verlag, 2006, pp. 175-184.
16. Lee, K.S., and Geem, Z.W., A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice, Computer methods in applied mechanics and engineering. 194 (2005) 3902-3933.
17. Lee, J., Song, S., and Yang, Y., H Shim, H. Lee, K. Lee, Y. Yoon, Multimodal function optimization based on the survival of the fitness kind of the evolution strategy, IEEE EMBS (2007), 3164-3167.
18. Purnomo**,** H.D., Wee H.M. (2012), Soccer game optimization: An innovative integration of evolutionary algorithm and swarm intelligence algorithm, in P. Vasant (Eds), "Meta-Heuristics optimization algorithms in engineering, business, economics, and finance" ed., IGI Global, pp. 386-420.
19. Rosenbrock, H.H., An automation method for finding the greatest or least value of a function, The Computer Journal, 3 (3) (1960) 175-184.
20. Saez, Y., Isasi, P. and Segovia, J., Interactive evolutionary computation algorithms applied to solve Rastrigin test function, in Advances in

soft computing (eds), Heidelberg: Springer-Berlin, (2005), pp. 682-691.
21. Silva, C.A., Sousa, J.M.C., and Runkler, T.A., Rescheduling and Optimization of Logistic Processes using GA and ACO, Engineering Application of Artificial Intelligent. 21 (2008) 343-352.
22. Torn, A., and Zilinskas, A., Global optimization, Lectures notes in computer science (1989), Springer-Verlag, Berlin.
23. Yang, X.-S., Harmony Search as a Meta-heuristic Algorithm, in Z. W. Geem (Eds.) Music-Inspired Harmony Search Algorithm. SCI 191, Heidelberg: Springer-Verlag, 2009, pp. 1-14.