Soccer Game Optimization: Fundamental Concept

Hindriyanto Dwi Purnomo

Abstracts

Optimization is important in science, engineering, economic as well as daily problems. The number of optimization methods increase continuously due to the growing optimization problems. In this study, we discussed the fundamental concept of soccer game optimization (SGO), a new algorithm inspired by the soccer player movements. The method consists of two basic movements; the *move off* and the *move forward,* to control the diversification and intensification and a player substitution procedure to keep the good solution during the search. The SGO elaborates the competition and the reproduction process in evolutionary algorithm with the powerful information sharing of swarm intelligence algorithm. As a new method, the SGO can be enhanced in many different ways such as elaborating more aspects of the soccer players' movement as well as implementing the proposed method to solve various optimization problems.

Keywords: information sharing, evolutionary algorithm, swarm intelligence, *move off, move forward,* soccer player movements

1. Introductions

Optimization is implemented in various fields of science as well as our daily life, such as finding the shortest route, profit maximization and cost minimization. Optimization is the process of finding the best solution (either minimum or maximum) using systematical procedures. In general, optimization can be classified as exact methods and approximate methods (Talbi, 2009). The exact methods will produce optimal solution; however, they are generally unable to solve large instances problems in a reasonable time. Branch and bound algorithm, dynamic programming and constraint programming are examples of well known exact method. On the other hand, the approximate methods produce acceptable solution in a reasonable time; however, there is no guarantee of finding an optimal solution.

Approximate solution can be divided into heuristics and metaheuristics. The heuristics methods contain problem-specific information and are customized for specific problems; therefore, it only performs well on specific class of problems. On the other hand, the metaheuristics algorithms have good performance in various classes of problems because it does not contain problem-specific information. Nevertheless, heuristics component could be added into the metaheuristics to increase its performance for specific problems.

In recent years, metaheuristics method has been developed very fast. These metaheuristics can be enhanced in four type approach:

1. Improving a single metaheuristics method: In this approach, major development is proposing new solution mapping procedure. However, some improvement may involve modify the basic metaheuristics methods to fit with the problems such as: 2-ANTBAL (Simaria and Vilarinho, 2009) and multiple colony of ant (Ozbakir et al, 2010).
2. Combining the metaheuristics with the heuristics method: in this approach, heuristics methods are integrated in the metaheuristics algorithm. Example of the heuristics methods for this approach are the use of rank positional weight to solve ALBP (e.g. Akpınar and Bayhan; 2011; Ozbakır and Tapkan, 2010).
3. Combining several metaheuristics methods: two or more metaheuristics methods, called hybrid metaheuristics, are used to solve the optimization problems. This approach becomes more popular in recent year. The motivation is to exploit different behaviors of metaheuristics in order to obtain better optimization algorithms. The hybridization lies on the benefit of synergy by exploiting the complementary character of different algorithms (Blum et al. 2011). There are many successful applications of hybrid metaheuristics in the literature and they can be used as guidance for developing new algorithms such as mentioned in Kao and Zahara (2008), Kiran et al. (2012), Katagiri et al., (2012) and Chen P.H. and Shahandashti S.M. (2009). The hybridization could solve more complex problems satisfactorily than single metaheuristics.
4. Proposing new conceptual model of metaheuristics method: in this approach, novel metaheuristic method is proposed based on a new analogy, such as: Bat Algorithm, BA, (Yang, 2010), Bee Colony Optimization, BCO, (Lučić and Teodorović, 2001) and Biogeography-Based Optimization, BBO (Simon, 2008).

Motivated by the advantages of hybridization, SGO algorithm was introduced by combining the advantages of evolutionary and swarm intelligence algorithms. The two

Department of Information Technology, Satya Wacana Christian University, Indonesia

metaheuristics families have different fundamental philosophies, however they can be integrated. In this study, we will discuss the fundamental concept of SGO. The rest of the paper is organized as follow: section 2 will describe the metaheuristics methods. Basic form of the SGO is provided in section 3. Section 4 provides the discussion regarding the characteristic of soccer inspires algorithm. Conclusion is provided in section 5.

2. Metaheuristics

Recently, metaheuristics has emerged as a powerful method to solve various optimization problems (Bianchi et al., 2008; Deep and Bansal., 2009). The method gains significant interests in research and industrial practices due to its effectiveness and general applicability. In many cases, the classical approaches based on mathematical and dynamic programming are feasible only for small size instances of problems and generally require a lot of computational efforts, therefore the metaheuristics turn up into promising alternatives to classical optimization methods (Bianchi et al., 2008).

The metaheuristics algorithms exploit randomness and set of rules to produce solutions. Due to the nature of heuristic methods, the metaheuristics methods do not guarantee optimal global solutions; however, they will provide acceptable solutions for large-sized and complex problems in a reasonable time (Talbi, 2009). Compared to the exact solution, metaheuristics algorithms are more flexible in their adaptability to fit the need of various optimization problems and they do not need to put formulation of the optimization problem. However, the algorithms need considerable problem specific adaptation in order to achieve good performance.

The metaheuristics have been greatly inspired by the natural phenomena and used the phenomena as the model or metaphor. Some of the methods mimic the biological evolution: the Genetic Algorithm (GA) (Holland, 1975), the animals' behavior: Ant Colony Optimization (ACO) (Dorigo and Caro, 1999) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995); the physical processes: Simulated Annealing (SA) (Kirkpatrick et al., 1983); and the music improvisation: Harmony Search (HS) (Geem et al., 2001). However, there are algorithms, e.g. Tabu Search (TS) that do not mimic the natural phenomena, but rather use the problem structure to develop search strategies.

There are two important elements of metaheuristics called intensification and diversification (Talbi, 2009, Yang, 2009; Hertz and Widmer, 2003; Blum and Roli 2003). Intensification is the ability to investigate the neighborhood of a potential solution while diversification is the ability to explore the whole solution space. The intensification plays an important role in improving the potential solution during the search. It exploits the area near a potential solution found during the search in order to obtain a better solution. On the other hand, diversification is very important to avoid being trapped in local optimal solution. In other words, intensification is a local search while diversification is a global search. These two components should be laid in balance to achieve a high performance.

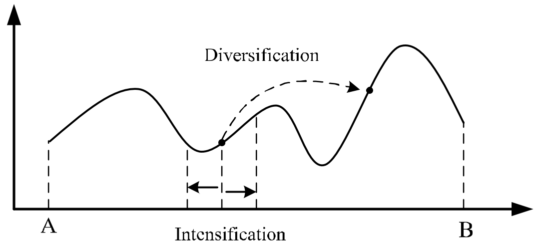


Figure 1. Intensification and diversification

Various metaheuristics methods have different strategies in balancing the intensification and diversification, depending on their conceptual model. However, they have common similarities in their procedure: initialization, solution's manipulation and solution update (Talbi, 2009). The initialization involves problem representation and generation of the first candidate solution. Problem representation is an approach to construct and to manipulate solution, e.g. the binary representation and real number representation. The problem representation must fulfill the completeness, connexity and efficiency criteria. Completeness means all solution associated with the problem must be represented, connexity refers to the existence of the connection or path way between any two solutions, while efficiency means the representation can be manipulated easily (Talbi, 2009). The initial solution is usually generated randomly. For the population based metaheuristics, the spread of the initial solution can be very important in order to explore the search space as widely as possible. This could minimize the probability that search process is trapped in local optima.

1	0	0	1	0	1	1	0	1

a. Binary representation

1.2	2.8	5.1	6.5	3.3	8.4	7.5	9.2	2.6

b. Real number representation

Figure 2. Problem representation permutasi

The solution's manipulation is the most important part of metaheuristics algorithms. There are various manipulation approaches among the metaheuristics algorithms that aim to balance the intensification and diversification. The differences come from different philosophy used in each metaheuristics algorithm and it distinguishes one metaheuristics algorithm from other metaheuristics algorithms.

Most metaheuristics algorithms will keep their current best solution during the search process and update it when necessary. The elitism concept is commonly used to update the current best solution, in which the best solution so far is replaced when the new candidate solution is better than the current best solution. However, there are algorithms that allow worse quality of solution to replace the current best solution, e.g. random walk in Simulated Annealing. The update strategy is important because it will drive the next search process. The random walk in Simulated Annealing is important to avoid being trapped in local optima. Some algorithms explicitly keep the good solution, e.g. Genetic Algorithm and Particle swarm optimization, while some other keep the good 'attribute' of the solution e.g. the pheromone in Ant Colony Optimization. In the population-based metaheuristics, the update strategies will guide the rolling of the population movement. Example of metaheuristics classification is illustrated by Dreo (2007).

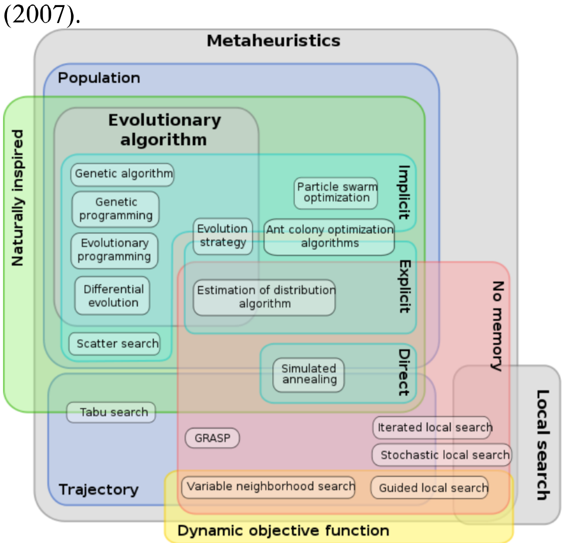


Figure 3. Classification of metaheuristics (Dreo, 2007)

3. Soccer Inspired Algorithm

Soccer Game Optimization (SGO), was introduced by Purnomo (2012). The algorithm mimics the behavior movements of soccer players' during a soccer games. In a soccer game, a player tries to be in a good position so that he can dribble the ball to reach the goal. In the game, cooperation among players in a team is important. The ball moves among the players and the ball position will become the main consideration of a player's movement. Players who do not dribble the ball try to move into better positions in order to become the ball dribblers. Some of the players will move closer to the ball (called *move forward*) and some other move on to explore the soccer field (called *move off*), regardless the ball position. Besides the ball position, a player's movement is also influenced by nearby players as well as their own experience. Through their experience, a player can recognize several good positions and they may tend to move to the nearby positions.

Beside the active players in the soccer field, a team also has several substitute players. The substitute players in the bench continuously keep track the movement of the active players. When an active player has low performance, he will be substituted by a substitute player. As the substitute players continuously observed the active player's movements, they will know some potential positions when they replaced the active players. The team strategies will also direct the players' movement; for example, a team with defensive strategy will place most of its players to run on its own half field while a team with offensive strategy will place many of its players on the opponent half field.

The behavior movements of soccer players can be associated with the optimization method. The player's effort to search a new position in the soccer field is similar to find the optimal solution in the search space during the optimization process. When a player moves into a new position, he will consider his environment (e.g. the ball position and the players nearby) as well as his own experience and this idea has been adopted in the swarm intelligence systems.

The SGO transforms the basic soccer player's movement into an optimization method by simplifying its environment and rules. The method implies a typical swarm intelligence system in which an individual in the population can act independently. Information sharing plays an important role in individual movement; however, the method also incorporates the basic idea of competition where each player tries to be the one who dribbles the ball and poor performance players will be substituted by the substitute players. The method takes the advantages of memory to store the individual knowledge. Many terms used in the method are derived from the soccer game. Soccer field represent the available search space. The number of kick corresponds to the number of iteration. The soccer rules illustrate the problem constraints that define some prohibited conditions. The goal is to achieve the optimal solution. However, the goal in this method is different with the goal in the real soccer game. Unlike in the real soccer games in which a team can score multiple goals in to opponent team, we define the goal as obtaining the best player position among the team, which mean reaching the best solution.

In the SGO, a team is a simultaneous set of vector solutions and each vector solution is called a player. Each player encodes a set of decision variables. The quality of a player is evaluated using the objective function. The ball dribbler represents the best solution obtained during the search process. The team, players, and decision variables can be illustrated in figure 4.

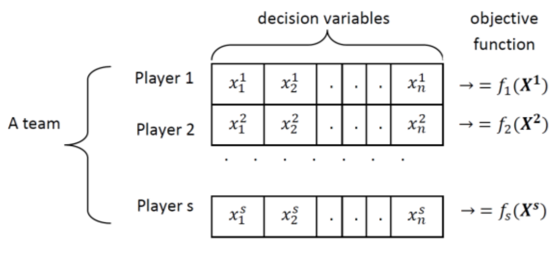


Figure 4. A team, players and decision variables

The SGO consists of a set of players in which a player can interact with other player in the set. A player with the most advantageous position will dribble the ball (called ball dribbler), and this player represents the best solution so far. The ball dribbler position is shared globally, so that all players can access this information (knowing which player has the best solution). This is similar to the real soccer game where all players consider the ball dribbler position in their movements. As the game continues, the ball dribbler can pass the ball to another player or remain holding the ball. In this method, the ball is used to point out which player is the best.

In order to control the player's movement, two main movements, called '*move off*' and '*move forward'*, are introduced to balance the diversification and intensification. The '*move off*' is mainly used to explore the solution space and it involves randomness. The movement minimizes the chance of premature convergence. The '*move forward*' is mainly used to explore the solution space nearby a player. The movement is determined by the cooperation or interaction between the player and other players. The interaction describes the information sharing among them. The information sharing is divided into two types, local information and global information. Local information means the information that can only be accessed by the nearby neighboring players. For example, a player's position is only considered by other players nearby. Global information means the information that can be accessed by all players. For instance, the ball dribbler position is very important to every player; therefore, all players should know his position.

Beside the players' movement, player substitution is another important procedure in the method. The substitute players' continuously observe the players movements and they memorize the 'good' player's positions. The substitute players update their knowledge based on the players' position from time to time. In the player substitution, players with poor performance will be replaced by the substitute players. The players that have been replaced then become the substitute players. They continuously update their knowledge based on the players' position from time to time to improve their quality. The SGO procedure can be described as follows:

1. Initializing players' position and the ball dribbler

   A team P_0 consists of p players $P_0=\{X_0^1, X_0^2, \ldots, X_0^p\}$ and substitute players consist of s players $S_0=\{X_0^{p+1}, X_0^{p+2}, \ldots, X_0^{p+s}\}$. Initially, a team P_0 and substitute players S_0 are generated randomly. A player encodes a potential solution $X_0^1=\{x^i_{1,0}, x^i_{2,0}, \ldots, x^i_{n,0}\}$ while a substitute player encodes a solution in a set of best solutions found so far $X_0^{p+1}= \{x^{p+1}_{1,0}, x^{p+2}_{2,0}, \ldots, x^{p+i}_{n,0}\}$. Objective function(s) is used to evaluate the player's and the substitute players. The best player in P_0 is selected as the ball dribbler B_0. The initial players' position is also considered as the initial players' best position $P_b = P_0$.

   Players' movement

   At each step, players in the team move into new positions. A player i can move into the new position by *move off* and *move forward* with certain probability (e.g. if m_i is the probability of *move off,* then $1\text{-}m_i$ is the probability of *move forward*).

Move forward: when a player selects the *move forward*, he will move toward the ball dribbler. The *move forward* can be described as follow:

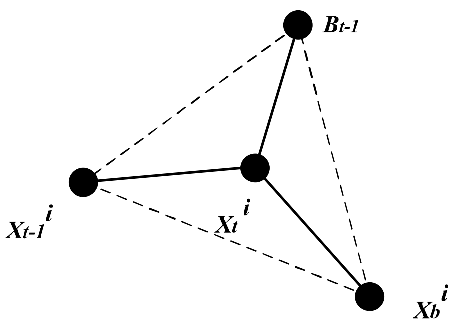
$$X_t^i = move\_forward\ (X_{t-1}^i, X_b^i, B_{t-1})$$

Which mean that the new player position X_t^i is determined based on the previous player position X_{t-1}^i, the player knowledge X_b^i and the ball position B_{t-1}.

To illustrate this operator, we considered a continuous problem as an example. For this problem, center of mass principle (Halliday et al, 1986) is used to calculate the player's new position. It is given as follow:

$$X_t^i = \frac{w_1 X_{t-1}^i + w_2 X_b^i + w_3 B_{t-1}}{w_1 + w_2 + w_3} \qquad (1)$$

where w_1, w_2, and w_3 are the weight for the previous player's position, the player's best position, and the ball dribbler's position respectively. Using the center of mass principle, we limited the search in the area among the previous player's position, the player's best position, and the ball dribbler's position. It can be illustrated in figure 5.



X_t^i is determined by the X_{t-1}^i, X_b^i, and B_{t-1}

Figure 5. Search domain

Move off: when a player selects the *move off*, the player explores the solution space, regardless of the ball dribbler's position. This movement involves randomness. The *move off* can be described as follow:

$$X_t^i = move\_off\ (X_{t-1}^i)$$

which means that the *move off* is self-manipulation (no need other players in order to get the new player position). After the *move off*, a player still has the opportunity to *move forward* by a probability l. The movement is illustrated in figure 6.

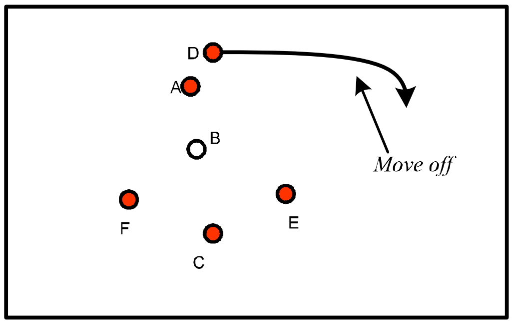


Figure 6. Illustration of *move off* for player D

2. The ball dribbler's update

When all players have moved into their new positions, the players will be evaluated based on objective function to select a candidate of the ball dribbler. If the ball dribbler candidate is better than the current ball dribbler, the ball is passed into that player. In other words, the current ball dribbler is replaced. Assuming a minimization problem, the replacement can be formulated as follow:

$$B_t = \begin{cases} x_t^i & if\, f x_t^i < B_{t-1}\ ,\ i \in CB \\ B_{t-1} & otherwise \end{cases} \qquad (2)$$

Where CB is the candidate of ball dribbler (e.g. the best player in an iteration).

3. Player's substitution

In each iteration, by a probability k, a player in the team is replaced by a substitute player if the position store by the substitute player better than the player position in the team. The replaced players become the substitute players.

4. The players' and the substitute players'

The active players will update their best position $x_{b,t}^i$ as follow:

$$x_{b,t}^i = \begin{cases} x_t^i & if\, f x_t^i < x_{b,t-1}^i \\ x_{b,t-1}^i & otherwise \end{cases} \qquad (3)$$

Similar to the active players, the substitute players also update their knowledge based on the current active players' positions. The role of substitute players is to store a set of best solution so far. This memory will be restored in the player's substitution mechanism.

$$x_t^i = \begin{cases} x_t^j & if\, f x_t^j < x_{t-1}^i,\ \ x_t^j \notin S \\ x_{t-1}^i & otherwise \end{cases} \qquad (4)$$

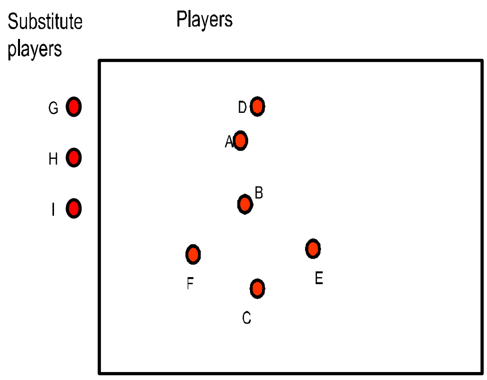
when the position of an active player have been stored by a substitute player, the position cannot be stored by other substitute player. Therefore, each substitute player will store different player's positions.

$$x_t^i \neq x_t^j\, for\ all\ x_t^j \in S, \qquad i \neq j \qquad (5)$$

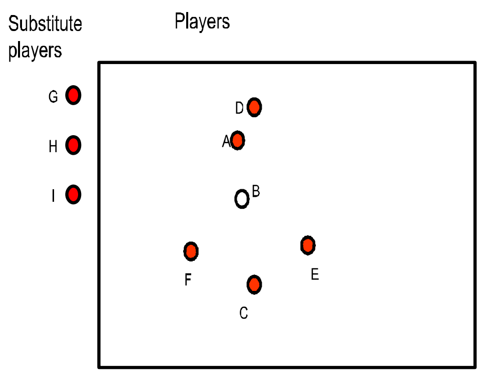
The players' movements, the ball dribbler's and players' knowledge update and player's substitution procedures are repeated until the stopping criteria are met.

There are various stopping criteria that can be applied to stop the search process, as are used in other metaheuristics algorithms. Some of the criteria are: the maximum number of iteration, the maximum CPU time, no significant improvement after a certain number of iteration and the best solution already reach the acceptable solution.

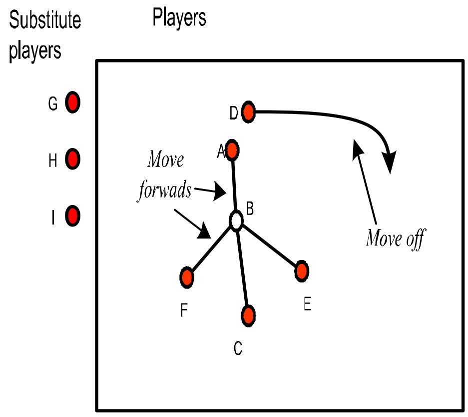
A player's movement is strongly driven by the information sharing among the players and the player's experience. In doing so, the proposed method takes the advantages of memory usage. In order to store the player's experience, the player's best position is kept in the memory. The *move off* incorporates random movements while the *move forward* is conducted by considering the player's current position, the player's best position and the other players', especially the ball dribbler's position (the *move forward* applies the *cognitive* and *social* learning). The ball dribbler's and the best player positions update adopts the elitism strategy in which replacement is done when the new solutions are better than the current solutions. The player substitution implements the competition concept in which players with low performance will be replaced by substitute players.
The players' movements for both *move forward* and *move off*, can be illustrated in figure 7.



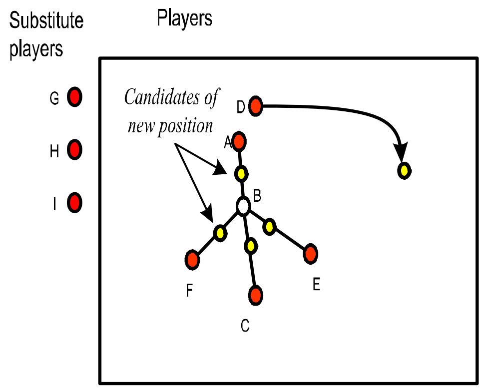
(a) Initialization



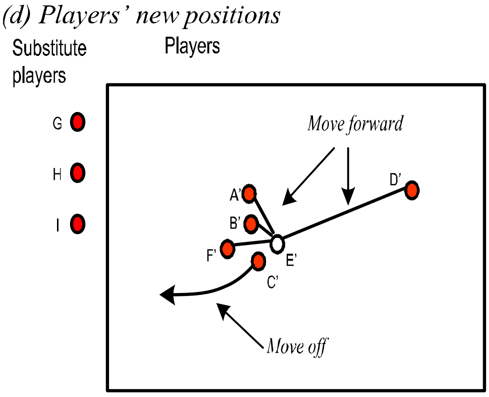
(b) Deciding the ball position-B



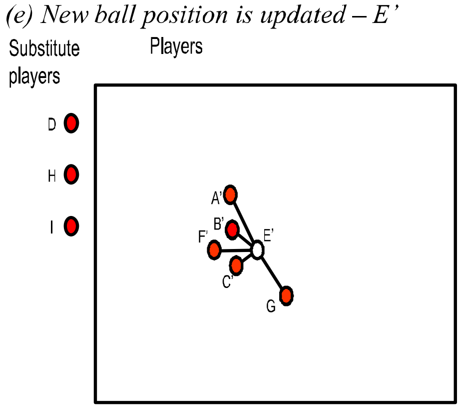
(c) Players' movements



(d) Players' new positions



(e) New ball position is updated – E'



(f) Players' substitution – player D is replaced by player G

Figure 7. Illustrating the players' movements

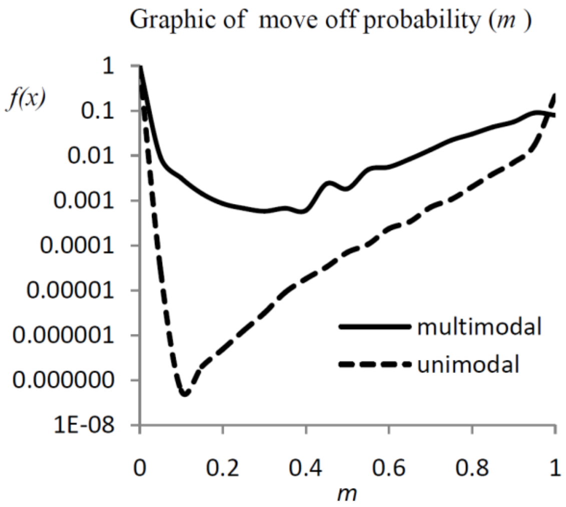
4. Discussion

4.1. Parameter sensitivity

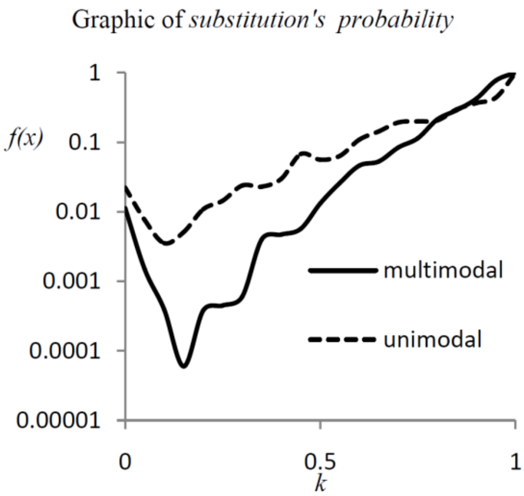
In order to investigate the effect of various parameters setting on the performance of the method, we implemented the method on unconstrained continuous problem with various parameter setting. We observe the effect of the probability of *move off* (*m*), the probability of *move forward* after the *move off (l)*, the probability of substitution (*k*) and the population size. We tested the method for both, the unimodal multimodal functions. Three unimodal functions, (Sphere function, Rosenbrock function and Schwefel's Problem 2.22) and three multimodal functions (Rastrigin's function, Ackley function and Generalized Penalized function 1) are considered. Both classes of problems are 10 dimensionality problems and the maximum objective function evaluation is 10000. For each parameter setting of a problem, the experiments are repeated 20 times, and then averaged. Next, the averaged value for each problem is normalized into [0, 1]. The graphic is derived by averaging the normalized value of all problems. The experiments were run on AMD Athlon™ II X2 260, 3.21 GHz and coded using Matlab 7.0. The experiments results are shown in figure 8.

Table 1. Test bed functions

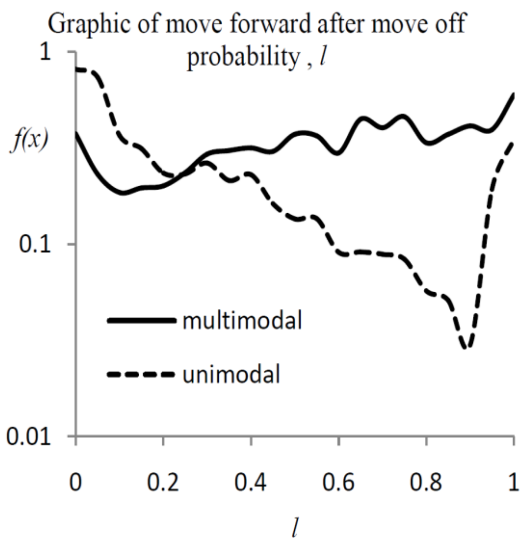
Name	Test function	S	f_{min}
Sphere function	$f_a = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$	0
Schewel's function 2.22	$f_b = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10, 10]^n$	0
Generalized Rosenbrock function	$f_c = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$	0
Rastrigin's function	$f_d = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	0
Ackley function	$f_e = \sum_{i=1}^{n} -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$	0
Generalized Penalized function 1	$f_f = \frac{\pi}{n}\{10sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^n$	0



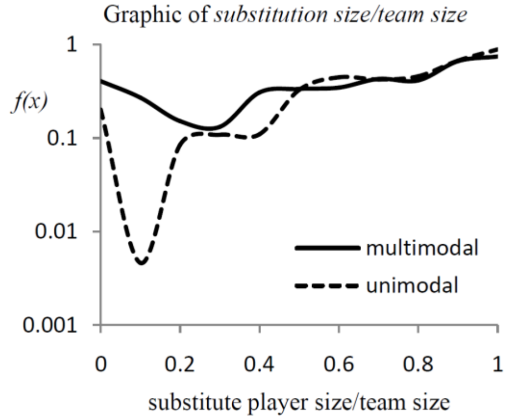
(a)



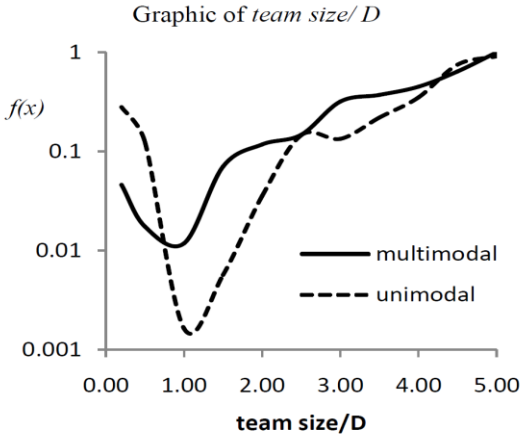
(b)



(c)



(d)



(e)

Figure 8. Graphic of parameters vs performance

Figure 8 (a) shows that the unimodal functions are more sensitive to the *move off*. Small value of *m* (around 0.05-0.1) will significantly increase the performance of the SGO. In the unimodal function, local optimal is also the global optimal, therefore, high probability of *move of* will be useless. The good value of *m* for multimodal functions is in the range [0.2 -0.4], which is higher from those in the unimodal functions. Very small value of *m* will make the search process being trapped in the local optima easily and very high value of *m* will make the search process closer to a random search.

The effect of substitution is more significant for the multimodal value than the unimodal value as can be seen in figure 8 (b). The substitution can store a set of best solution so far; therefore, the player substitution could boost local search nearby the *s* best solution so far. The good *k* value for both unimodal and multimodal functions are almost the same, [0.05-0.2].

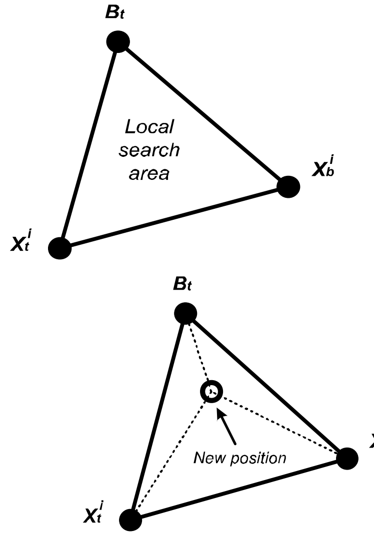
The influence of *move forward* after the *move off*, *l*, is significantly different for the multimodal and unimodal function. The effect of *l* is not so apparent for the multimodal functions. However, the range value of *l* between [0.1-0.3] could increase the performance of the method. On the other hand, the effect of *l* is quite obvious for the unimodal function. As a unimodal function only has one optimal value, when a player approaches the best player so far, he is likely increasing his performance (approaching the optimal value). The good value of *l* for the unimodal function is [0.8-0.95].

The substitute player size, *s*, also has more influence on unimodal functions than the multimodal functions as shown in figure 8 (d). The substitute player size is measured from 0 to 1. Zero means no substitute player at all and 1 means the number of substitute player is the same as the player in the team. A small number of substitute player, approximately 0.05 to 0.15 of the team size, could significantly increase improve the performance of the propose method for unimodal function. While the good range of substitute player for the multimodal functions are approximately 0.1-0.4 of the team size.

The effect of team size, *p*, is obviously observed for both unimodal and multimodal functions. Both of them have similar trend. The team size is observed from 2 players until 5 times of the problem dimension. The good range of the team size is around [0.5- 2.0] for both unimodal and multimodal functions.

4.2. Analysis Based on the Intensification and Diversification Frame

It is noted that social interaction plays an important role in the swarm intelligent because it will determine the population behavior. This social interaction can be viewed as a form of cooperation among individuals. In the SGO, this interaction is incorporated in the *move forward* operator. In this movement, a player run approaching the ball dribbler, therefore the player need to know the ball dribbler's position, his current position, his own experience and probably other players nearby. It can be seen that the movement integrates the social as well as cognitive learning where a player receives information about the ball dribbler's position, his current position and his own experience (his best position). This movement mainly used to maintain intensification. An illustration of search area for *move forward* is shown in figure 9. The *move off* is devoted to explore the whole search space in stochastic manner. The *move off* is used to maintain the diversity of potential solution and is similar to the biological mutation. Therefore, the *move off* is more likely a mutation in the evolution algorithm.



(a) (b)

Figure 9. The local search area

The I&D frame proposed by Blum and Roli (2003) is used to analyze the intensification and diversification of the proposed method. The intensification and diversification framework (I&D framework) give an integrated view of intensification and diversification in metaheuristics (Blum and Roli, 2003). The frame can be illustrated as a triangle in which its corners represent three extreme intensification and diversification (I&D) component. The corner denoted by objective guided (OG), Non objective guided (NOG) and random (R). Corner OG means I&D components only guided by the objective function. Corner NOG means I& D components guided by functions other than objective function. Both corner OG and NOG do not use any random component. Corner R means I& D components are completely random.

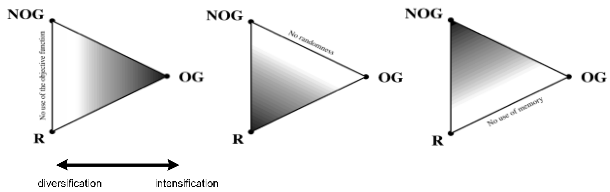
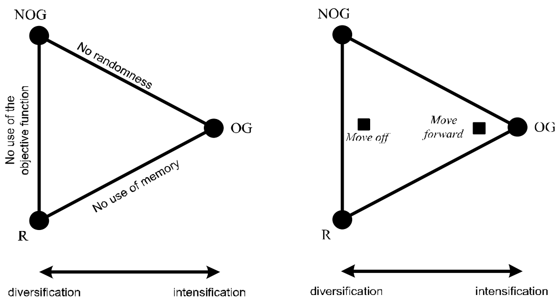


Figure 10. I & D frame (Blum and Roli, 2003)

Based on the *I&D frame*, the *move off* operator is not guided by the objective function and it involves randomness. Therefore, the *move off* operator is close to the R (randomness) and NOG (non objective guided) side, with higher diversification effect. The *move forward* uses memory and it is guided by the objective function (the ball dribbler selection is based on the objective function). In the frame, the *move forward* is close to the OG corner, which has higher intensification effect.



a. I&D frame b. The move off and move forward

Figure 11. The *move off* and *move forward* in the I&D frame

The substitute players can be assumed as adaptive memory, in which they store the *s* best solution achieved during the search, and is updated continuously. When an active player has poor performance, he will be substituted by a substitute player. Therefore, the substitute players can be assumed as the base search point when a player is getting lost during the search process. In addition, the substitution mechanism reflects the competition concept existing in the evolutionary algorithm in which poor performance players are replaced with better players.

5. Conclusion

SGO is a novel algorithm that try to integrates the basic concept of evolutionary (competition and evolutionary operator) and swarm intelligent algorithm (information sharing). The method has two types of player movements (the *move off* and the *move forward*) and player substitution mechanisms. The player movements are used to balance the intensification and diversification while the player substitution is used to keep the good solution so far during the search process.

The experiment on parameter tuning provides more insight on the behavior of the proposed method. The effect of *move off* is more obvious in multimodal functions than in unimodal functions. The finding is expected because the aim of the *move off* procedure is to enhance the exploration of the search space to find new promising area. In the multimodal function, the *move off* will significantly contribute avoiding being trapped in local minima while in the unimodal function, there is no local optima, therefore the *move off* is less useful. This behavior is also supported by the results finding on the effect of *move forward* after the *move off*, *l*. High probability of *l* is more expected in unimodal function than in multimodal functions. Its mean that *move forward* provide better contribution in the performance of the proposed method for unimodal function than the *move off*. It can be explained as follow: as a unimodal function only has one optimal value and no local optima, then, when a player approaches the best player so far, it is likely that the player also approaching the global optima.

The probability of player's substitution and the number of player substitution reveals to the same behavior. The proposed method performs well when the probability of player's substitution is small. This can be explained as follow: the players substitution can be assume as a mechanism to force the search process into the most promising area so far. The higher the probability of player's substitution, than more effort is allocated to search the solution into this area. Problems will occurs when the most promising area found so far is the local optima, because the search process will likely trapped in this area. The number of substitute players will also give the same effect as the probability of player's substitution. The higher the number of substitute players, the higher the effort provided to search in the most promising area found so far.

As a new method, there are still many opportunities to enhance the SGO. Through this thesis, several future researches can be identified. Some of them are briefly summarized as follow:

1. Investigating various aspects of soccer player' movement:

   The proposed method is simplification of the soccer player' movement, therefore, many aspect of the soccer player' movement have not been addressed. In this thesis, we only consider two main movements, *move off* and *move forward*, and a player substitution procedure. An aspect of the soccer player' movement that can be future investigates is the aggressiveness of players'. For example, the team strategy, offensive and defensive, can be related to the search strategy. In the defensive strategy, many of its players run on its own half field, which mean players exploits a small area of the field. This can be associated with highly effort on local search (intensification). On the other hand, the offensive strategy encourages players to explore both, its own and its opponent field, therefore it can be associated with highly effort on diversification (global search).In addition, a defender movement is usually not as aggressive as a striker movement. A playmaker has more opportunities to control the flow of the ball to both defender and striker. This reveals that the individual in the population could be classified into several types in which each of them has different task priorities.
2. Enhancing the player movement procedure

   In this thesis, we consider simple movement for both, *move forward* and *move off*. For the *move forward*, we consider the center of mass principle. Deciding the weight of each player that contributes in this movement could become a problem. In this thesis, we did not discuss this issue in more detail. Instead, we used the golden ratio that having good properties in narrowing the search spaces as demonstrated in the golden section search. In addition, we only consider the ball position, the current player position and the player best position in *move forward*. Investigating the effect of involving more neighborhood players in the *move forward* could become an interesting research topic.

The *move off* is introduced to facilitate the proposed method to explore the whole search space. The appropriate mechanism of this operator can be problem dependent. Therefore, investigating various *moves off* strategies in various problems can be considered in the future research.

3. Apply the method for various optimization problems

As a new method, it is expected that new version of SGO will improve its performance. Modification the basic procedure of the proposed method can also become a problem dependent. Therefore, investigating the implementation of SGO in various optimization problems is a significant contribution to the development of this method.

Various application of SGO could consider multi objectives problems and also various size of optimization problems. The proposed method discussed in this thesis only be applied on single objective functions with relatively small size problems. Therefore future research can consider the application of SGO on multi objective function and also large problems. Due to the rapid development of computing technology, many very large size problems are solved using parallel computing. Developing distributed version of SGO for parallel computing could become a significant contribution to develop SGO.

4. Investigate the general framework of metaheuristics

There is no solid theoretical framework in meta-heuristics field. This bring challenge to researchers to cope with several issues such as determine factors that are needed to speed up the convergence process for a given problem, increasing the effectiveness of the method for a specific problem and how to prove the global optima are reached by the method.

References

Akpınar, S., Bayhan, G. M. (2011). A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints. Engineering Applications of Artificial Intelligence, 24 (3), 449–457.

Bianchi L., Dorigo M., Gambardella L. M., & Gutjahr W. J. (2008). A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing, 8(2), 239-287.

Blum, C., Roli A. (2003). Metaheuristic in combinatorial optimization: Overview and conceptual comparison. ACM computing survey, 35(3), 268-308

Chen P.H., Shahandashti S.Y. (2009). Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. Automation in Construction, 18, 434–443.

Deep, K. & Bansal, J.C. (2009). Mean particle swarm optimization for function optimization. International Journal of Computational Intelligence Studies, 1, 72-92.

Dorigo, M., Caro, G. D. (1999). Ant colony optimization: a new meta-heuristic. Proceedings of the 1999 Congress on Evolutionary Computation, 1470-1477.

Dréo, J. (2007). Classification of metaheuristics. Retreived Mat 2nd 2013 from http://metah.nojhan.net/post/2007/10/12/Classification-of-metaheuristics.

Geem, Z. W., Kim, J. H. & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. Simulation, 76, 60-68.

Hertz, A. & Widmer, M. (2003). Guidelines for the use of meta-heuristics in combinatorial optimization. European Journal of Operational Research, 151, 247-252.

Halliday, D., Resnick, R., Walker, J. (1986). Fundamental of physics, 2nd edition, John Wiley & Sons, Inc.

Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor, University of Michigan Press.

Kao Y.T., Zahara E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Applied Soft Computing, 8, 849–857.

Katagiri H., Hayashida T., Nishizaki I., & Guo Q. (2012). A hybrid algorithm based on tabu search and ant colony optimization for k-minimum spanning tree problems. Expert Systems with Applications, 39, 5681–5686.

Kennedy, J. & Eberhart, R. (1995).Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, IV, 1942-1948.

Kıran M.S., Ozceylan E., Gunduz M., PaksoyT. (2012). A novel hybrid approach based on particle swarm optimization and ant colony algorithm to forecast energy demand of turkey. Energy Conversion and Management, 53, 75–83.

Kirkpatrick, S., Gelatt, C. D., JR. Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220, 671-680

Lučić, P., Teodorović, D. (2001). Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, Portugal, 441–445.

Ozbakir L., Tapkan P. (2010). Balancing fuzzy multi-objective two-sided assembly lines via beesalgorithm. Journal of Intelligent & Fuzzy Systems 21,317-329.

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.

Simaria, A. S. Vilarinho, P. M. (2009). 2-ANTBAL: An ant colony optimization algorithm for balancing two-sided assembly lines. Computers & Industrial Engineering, 56, 489-506.

Simon, D. (2008). Biogeography-Based Optimization. IEEE Transactions on evolutionary computation, 12 (6), 702-713.

Talbi, E.-G.(2009). Metaheuristics, from design to implementation. Lille, John Wiley & Sons.

Yang, X.-S.(2009). Harmony search as a meta-heuristic algorithm. Music-Inspired Harmony Search Algorithm, Studies in Computational Intelligence.Heidelberg, Springer-Verlag.

Yang, X. S. (2010). A New Metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, 284, Springer, 65-74