10.1109/SoCPaR.2009.21

2009 International Conference of Soft Computing and Pattern Recognition

League Championship Algorithm: A new algorithm for numerical function optimization

Ali Husseinzadeh Kashan
Department of Industrial Engineering
Amirkabir University of Technology
Tehran, Iran.
a.kashani@aut.ac.ir

Abstract— **Inspired by the competition of sport teams in a sport league, an algorithm is presented for optimizing nonlinear continuous functions. A number of individuals as sport teams compete in an artificial league for several weeks (iterations). Based on the league schedule in each week, teams play in pairs and the outcome is determined in terms of win or loss, given known the team's playing strength (fitness value) resultant from a particular team formation (solution). In the recovery period, each team devises the required changes in the formation/playing style (a new solution) for the next week contest and the championship goes on for a number of seasons (stopping condition). Performance of the proposed algorithm is tested in comparison with that of particle swarm optimization algorithm (PSO) on finding the global minimum of a number of benchmarked functions. Results testify that the new algorithm performs well on all test problems, exceeding or matching the best performance obtained by PSO. This suggests that further developments and practical applications of the proposed algorithm would be worth investigating in the future.**

Keywords: ***Global optimization, Numerical optimization, Metaheuristic algorithms, Sport league championships.***

I. INTRODUCTION

Inspired from the natural and social phenomena, metaheuristic algorithms have attracted many researchers from various fields of science in recent years. This interest is by far in applying the existing meta-heuristics for solving real word optimization problems in many fields such as business, industry, engineering, etc. However, beside all of these applications, occasionally a new metaheuristic algorithm is introduced that uses a novel metaphor as guide for solving optimization problems.

There are many ways to classify metaheuristic algorithms, e.g. nature inspired versus others; population based versus single point, etc. Two broad groups are evolutionary algorithms (EA) and the swarm intelligence based algorithms. EAs are inspired by nature's capability to evolve living beings well adapted to their environment. EAs can be succinctly characterized as computational models of evolutionary processes [1]. Genetic algorithm is among the first and most popular EAs. Swarm intelligence (SI) originated from the study of colonies, or swarms of social organisms. Studies of the social behavior of organisms (individuals) in swarms prompted the design of very efficient optimization and clustering algorithms. For example, simulation studies of the graceful, but unpredictable, choreography of bird flocks led to the design of the particle swarm optimization algorithm, and studies of the foraging behavior of ants resulted in ant colony optimization algorithm [2].

This paper proposes a new evolutionary algorithm called League Championship Algorithm (LCA) for global optimization, which mimics the sport league championships. A number of individuals making role as teams compete in an artificial league for several weeks (iterations). Based on the league schedule in each week, teams play in pairs and the outcome is determined in terms of win or loss, given known the team's playing strength (fitness value) resultant from a particular team formation (solution). Keeping track of the previous week experiences, each team devises the required changes in the formation/playing style (a new solution) for the next week contest and the championship goes on for a number of seasons (stopping condition).

The remainder of the paper is organized as follows. Section II reviews, in brief, the common terminology used in sport leagues. Section III puts forwards the idealized rules and describes the new metaheuristic algorithm in a step by step manner. Section IV deals with computational tests and comparisons. Finally section V concludes the paper.

II. A REVIEW ON THE KEYWORDS RELATED TO THE SPORT LEAGUE CHAMPIONSHIPS

Here we shall have a brief introduction on the terms that are commonly related to the sport league championships; especially those which will be metaphorically used in LCA.

A *sports league* is an organization that exists to provide a regulated competition for a number of people to compete in a specific sport. League is generally used to refer to competitions involving team sports, not individual sports. A league championship may be contested in a number of ways. Each team may play every other team a certain number of times. In such a set-up, the team with the best record becomes champion, based on either a strict win-loss-tie system or on a points system where a certain number of points are awarded for a win, loss, or tie, while bonus points might also be added for teams meeting various criteria [3].

Generally each team has a playing style which is realized during the game via team formation. *Formations* are a

978-0-7695-3879-2/09 $26.00 © 2009 IEEE
DOI 10.1109/SoCPaR.2009.21

method of positioning players on the pitch to allow a team to play according to its pre-set tactics. For example, the most common formations in soccer are variations of 4-4-2, 4-3-3, 3-2-3-2, 5-3-2 and 4-5-1 [4]. Different formations can be used depending on whether a team wishes to play more attacking or defensive. Usually each team has a best formation which is often related to the type of players available to the coach.

It is vital for a sport team to devise suitable game plans and formations for every match. After each match, coaches analyze theirs and their opponent's game to plan on how they can develop a style of play to improve on their *weaknesses* or afford more on their *strengths*. The Analysis also includes the evaluation of *opportunities* and *threats* that comes along with the unique dynamics of the team. This kind of analysis is typically known as strengths/weaknesses/opportunities/ threats (*SWOT*) analysis, which explicitly links internal (strengths and weaknesses) and external factors (opportunities and threats).

After analyzing the game, coaches must modify and manipulate the game setting in training to develop areas that the players and team need to practice. The game will continue to reveal improvement areas for future practices. Weather we want our team to play counter attacking or high pressure defense, we need to use the game to work out how we can help it establish a style that will suit the players and work against the opponents.

III. The League Championship Algorithm (LCA)

The above keywords can be matched to the most commonly used evolutionary terms as follows: "league" stands for the "population of solutions"; "team *i*" stands for the "i^{th} member of the population"; "a team formation" stands for a "solution"; "week" stands for "iteration"; and "playing strength" stands for the "function or fitness value". In the remainder of the paper we use both terminologies, alternately.

Before giving details of LCA, we first put forward several idealized rules which constitute the rationale of LCA.

1. It is more likely that a team with greater playing strength wins the game. The term "playing strength" refers to a team's ability to beat another team.
2. The outcome of a game is not predictable given known playing strength of the teams perfectly. In other words, it is not unlikely that FC Barcelona loss the game to Sandoghe_Zakhire_Robat_Karim from Iranian 3rd division.
3. The probability that team *i* beats team *j* is assumed equal from both teams point of view.
4. The outcome of the game is only win or loss; there is no tie.
5. Teams only focus on their upcoming contest without regards of the other future matches. Formation settings are done just based on the previous week(s) events.
6. When team *i* beats team *j*, any strength helped team *i* to win has a dual weakness caused team *j* to loss. In other words, any weakness is a lack of a particular strength. An implicit implication of this rule is that although the match outcome may be imputed to chance, technical staff may not believe it.

The following algorithm gives the basic steps of LCA in details.

The League championship algorithm (LCA)

```
1. Initialize the league size (L) and the number of
     seasons (S); t=1;
2. Generate a league schedule;
3. Initialize team formations (generate a population of L
     solutions) and determine the playing strengths
     (function or fitness value) along with them. Let the
     initialization be also the teams' current best
     formation;
4. While $t \le S.(L-1)$
5.     Based on the league schedule at week t,
       determine the winner/ loser among every pair of
       teams using a playing strength based criterion;
6.     t=t+1;
7.     For i=1 to L
8.         Devise a new formation for team i for the
           forthcoming match, while take into account
           the team's current best formation and
           previous week events. Evaluate the playing
           strength of the resulting arrangement;
9.         If the new formation is the fittest one (that is,
           the new solution is the best solution achieved
           so far for the ith member of the population),
           hereafter consider the new formation as the
           team's current best formation;
10.    End for
11.    If mod(t,L-1)=0
12.        Generate a league schedule;
13.    End if
14. End while.
```

Like many other evolutionary algorithms, LCA works with a population of solutions. Each member of the population at iteration *t* is a potential solution that is related to one of teams and is interpreted as the team formation at week *t*. Given a function *f* of *n* variables, each solution such *i* can be represented with a vector of *n* real numbers. We can see each variable as one of players where changing in the value of the variables may reflect changing in the job of the relevant player. We use $X_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{in}^t)$ to address the formation of team *i* at week *t*. By $f(X_i^t)$ we address the function value relevant to X_i^t (recall that $f : \Re^n \rightarrow \Re$ and we are seeking the global minimum of f that is, find $X^* \in \Re^n$ such that $f(X^*) \le f(X)$, $\forall X \in \Re^n$). This value is called the playing strength along with formation X_i^t. By $B_i^t = (b_{i1}^t, b_{i2}^t, \ldots, b_{in}^t)$ we address to the best formation for team *i*, experienced till week *t*. This is the best solution that has been obtained so far for the *i*th member. To determine

B_i^t, a greedy selection is done at each iteration between $f(X_i^t)$ and $f(B_i^{t-1})$. If $f(X_i^t)$ is better than $f(B_i^{t-1})$, then $B_i^t \leftarrow X_i^t$; otherwise $B_i^t \leftarrow B_i^{t-1}$.

Now we put forward greater details on the main steps of LCA; especially the manner of generating the league schedule, determining winner/loser and setting up a new team formation.

A. Generating the league schedule

Since LCA mimics the championship in a sport league, it becomes required to schedule matches in the artificial league. A *single round-robin schedule* is utilized where each team plays every other participant once in each season. For a league of size *L*, single round robin tournament requires $L(L-1)/2$ matches, because in each of $(L-1)$ rounds (weeks), $L/2$ matches will be run in parallel (if *L* is odd, there will be *L* rounds with $(L-1)/2$ matches, and one team have no game in that round).

The scheduling algorithm is simple and we illustrate it partly using a league of 8 teams $(L=8)$. Let assign each team a number and pair them off in the first week (Fig. 1a). Fig. 1a implies that team1 plays 8, 2 plays 7 and so on. For the second week, fix one team, say team one, and rotate the others clockwise (Fig. 1b). In this week, 1 plays 7, 8 plays 6 and so on. For the third week, once again rotate the order clockwise. So, 1 plays 6, 7 plays 5 and so on (Fig. 1c). We continue this process until getting the initial state. The last week (week 7) schedule can be obtained from Fig. 1d. If *L* is odd, a dummy team is added. In each week, the opponent of the dummy team does not play and gets rest.

a)	b)	c)	...d)
1 2 3 4	1 8 2 3	1 7 8 2	1 3 4 5
8 7 6 5	7 6 5 4	6 5 4 3	2 8 7 6

Figure 1. An illustrative example of the league scheduling algorithm.

It is worth mentioning that the round-robin tournament can be modelled as an edge-coloring problem in a diagraph [5].

In LCA, championship continues for *S* successive seasons where the league schedule for each season is a single round robin schedule, yielding $S.(L-1)$ weeks of contests. It is worth mentioning that in our implementation of LCA we use the same schedule for all of the *S* seasons. However, one can change the schedule at the start of each season.

B. Determining winner/loser

In a regular league system, teams compete on weekends and the outcome in terms of win, loss or tie is determined for each team. Based on this, each team is scored by 3 points for win, 0 for loss and 1 for tie. Disregarding the effect of external impositions that may entrap even great teams into occasional crisis, generally it is more likely that the more powerful team beats the weaker one. Therefore, proportional to its playing strength, each team may have a chance to win the game (idealized rule 1 and 2).

In LCA, the winner/loser is determined using a tournament selection which is a typically used in genetic algorithm. With the stochastic binary tournament, involving two individuals in competition, the best wins with a fixed probability *p* ranging between 0.5 and 1. Instead of considering a fixed chance of win (loss) for the fitter (unfit) one, we force teams to have their chance based on their degree of fit. The degree of fit is measured in comparison with an ideal reference point and is proportional to the team playing strength.

Let us consider opponents *i* and *j* at week *t*, having their formation strategy X_i^t and X_j^t, and playing strength $f(X_i^t)$ and $f(X_j^t)$, respectively. Let p_i^t be the chance of team *i* to beat team *j* at week *t*. p_j^t can be defined accordingly. Let f^* be the optimal function value. Based on the idealized rule 1 we can write

$$\frac{f(X_i^t)-f^*}{f(X_j^t)-f^*}=\frac{p_j^t}{p_i^t}. \qquad (1)$$

Based on the idealized rule 3 we can also write

$$p_i^t + p_j^t = 1. \qquad (2)$$

From (1) and (2) we thus get

$$p_i^t = \frac{f(X_j^t)-f^*}{f(X_j^t)+f(X_i^t)-2f^*}. \qquad (3)$$

To simulate win or loss, a random number in [0,1] is generated and if it is less than or equal to p_i^t, team *i* wins and team *j* losses; otherwise *j* wins and *i* losses.

If $f(X_i^t)$ be very close to $f(X_j^t)$, then $p_i \rightarrow \frac{1}{2}$ and if $f(X_j^t) >> f(X_i^t)$, then $p_i^t \rightarrow 1$. Since f^* is not known in advance, we estimate it by the minimum value of f (i.e. $\hat{f}^t$) found so far, that is $\hat{f}^t = \min_{i=1,\dots,L}\{f(B_i^t)\}$.

It is worthy of note that the above win/loss simulation is consistent with the first four idealized rules mentioned in section 3.

C. Setting up a new team formation

Before any strategy or intervention is applied, it is important for a coach to evaluate the *strengths* and *weaknesses* of the individual members and the team as a whole. This will serve as a guide as to how to approach them and the kind of professional relationship that should be developed, which area to focus on, and how to teach the required game skills to enhance their performance. The analysis also includes the evaluation of *opportunities* and *threats* that comes along with the unique dynamics of the team. Strengths and weaknesses are often internal factors while opportunities and threats often relate to external factors. This analysis is typically known as SWOT analysis and can be used when a desired objective is defined.

Considering the goal of beating the opponent, technical staff analyzes their own previous game and also the opponent's previous game. This analysis is called *post match analysis*. Those points sensed from the analysis of team's previous game (at week *t*) are considered as internal factors (strengths/weaknesses) and those perceived from analyzing opponent's previous game can be regarded as external factors (opportunities/threats).

In an artificial post match analysis of team *i*, if team has won (lost) the game from (to) team *j* at week *t*, then it is assumed that the prosper (loss) is the direct consequence of the team strengths (weaknesses) or based on the idealized rule 6 of section 3, it is the direct consequence of the weaknesses (strengths) of team *j*. Now, based on the league schedule at week $t+1$, assume that the next mach of team *i* is with team *l*. If team *l* has won (lost) the game from (to) team *k* at week *t*, then this success (loss) and the team formation behind it might be a direct threat (opportunity) for team *i*. Apparently, this success (loss) is the result of some strengths (weaknesses). Focusing on the strengths (weaknesses) of team *l*, gives us the knowledge to get avoidance of threats (advantages of opportunities). Therefore, based on idealized rule 6, we can focus on the weaknesses (strengths) of team *k*.

Following the above discussions and based on its and its opponent status at previous week (idealized rule 5), now team *i* can takes the suitable actions listed in Fig. 2. Let us assume that X_i^t, X_j^t and X_k^t are the formation strategies correspond to teams *i*, *j* and *k* at week *t*, respectively. By $X_k^t - X_i^t$ we address the gap between the arrangement of team *i* and team *k*, sensed via "focusing on the strengths of team *k*". In this situation, team *k* has won team *l* and to beat *l*, it is reasonable that team *i* devises a playing style rather similar to that was adopted by team *k* at week *t* (for example play counter attacking or high pressure defence). In a similar way we can interpret $X_i^t - X_k^t$ when "focusing on the weaknesses of team *k*". In other words, it is reasonable to avoid a playing style rather similar to that was adopted by *k* (for example avoid playing counter attacking or avoid high pressure defence).

Given the fact that each team such *i* plays based on its current best formation B_i^t (which has found it suitable over the time), while devising some changes resulted from post match analysis, we can setup the new formation $X_i^{t+1} = (x_{i1}^{t+1}, x_{i2}^{t+1}, \ldots, x_{in}^{t+1})$ for team *i* $(i = 1, \ldots, L)$ by one of the following equations.

If i was winner and l was winner, then

$$x_{id}^{t+1} = b_{id}^t + y_{id}^t (c_1 r_1 (x_{id}^t - x_{kd}^t) + c_1 r_2 (x_{id}^t - x_{jd}^t)) \ \forall d = 1, \ldots, n \qquad (4)$$

Else if i was winner and l was loser, then

$$x_{id}^{t+1} = b_{id}^t + y_{id}^t (c_2 r_1 (x_{kd}^t - x_{id}^t) + c_1 r_2 (x_{id}^t - x_{jd}^t)) \ \forall d = 1, \ldots, n \qquad (5)$$

Else if i was loser and l was winner, then

	i: winner ***l: winner*** ***Focusing on …***	***i: winner*** ***l: loser*** ***Focusing on …***	***i: loser*** ***l: winner*** ***Focusing on …***	***i: loser*** ***l: loser*** ***Focusing on …***
S	own strengths (or weaknesses of *j*)	own strengths (or weaknesses of *j*)	-	-
W	-	-	own weaknesses (or strengths of *j*)	own weaknesses (or strengths of *j*)
O	-	weaknesses of *l* (or strengths of *k*)	-	weaknesses of *l* (or strengths of *k*)
T	strengths of *l* (or weaknesses of *k*)	-	strengths of *l* (or weaknesses of *k*)	-

Figure 2. Actions suitable for team *i* when devising its formation for the next match (based on the win/loss statuses).

$$x_{id}^{t+1} = b_{id}^t + y_{id}^t (c_1 r_2 (x_{id}^t - x_{kd}^t) + c_2 r_1 (x_{jd}^t - x_{id}^t)) \ \forall d = 1, \ldots, n \qquad (6)$$

Else if i was loser and l was loser, then

$$x_{id}^{t+1} = b_{id}^t + y_{id}^t (c_2 r_2 (x_{kd}^t - x_{id}^t) + c_2 r_1 (x_{jd}^t - x_{id}^t)) \ \forall d = 1, \ldots, n \qquad (7)$$

End if

In the above formulas $d = 1, \ldots, n$ is the dimension index. r_1 and r_2 are uniform random numbers in [0,1]. c_1 and c_2 are constant coefficients used to scale the contribution of the strength and weakness components respectively. Note that the sign in parenthesis results in acceleration toward winner or avoidance from loser. For each team *i*, we have:

$l =$ Index of the team that will play with team *i* based on the league schedule at week $t+1$.

$j =$ Index of the team that has played with team *i* based on the league schedule at week *t*.

$k =$ Index of the team that has played with team *l* based on the league schedule at week *t*.

y_{id}^t is a binary change variable which indicates whether the *d*th element in the current best formation will change or not. The value of 1 for y_{id}^t allows making change in the value of b_{id}^t. Let us define $Y_i^t = (y_{i1}^t, y_{i2}^t, \ldots, y_{in}^t)$ as the binary change array with number of ones equal to q_i^t. It is rather unusual that coaches do changes in all or many dimensions of the team. Generally the number of changes is relatively low. By analogy, it may seem suitable that the number of ones in Y_i^t be small. To simulate the rate of changes (q_i^t), we use a truncated geometric distribution [6]. Using a truncated geometric distribution, we can set the rate of changes dynamically, while putting more weights on the smaller rate of changes. The following formula gives the random number of changes that are done in the body of B_i^t to get the new formation X_i^{t+1}:

$$q_i^t = \left\lceil \frac{\ln(1-(1-(1-p_C)^n)r)}{\ln(1-p_C)} \right\rceil : \; q_i^t \in \{1,2,...,n\} . \qquad (8)$$

Where r is a random number in [0,1] and $p_C \in (0,1)$ is an input parameter. Typically p_C is known as the probability of success in the truncated geometric distribution. The larger the value of p_C, the smaller the number of changes. After simulating the number of changes by (8), q_i^t dimensions are selected randomly from B_i^t for changing (i.e. their corresponding y_{id}^t gets 1).

It is possible to introduce other variants of LCA through introducing alternative equations for developing the new formation. Instead of considering the previous arrangement of teams (i.e., X^t) in SWOT analysis, we may assume that post match analysis are done based on teams' best formation. This means, using (b^t) in place of (x^t) in the right side of equations (4)-(7).

IV. Computational Experiments

In order to test the capability of LCA, a set of 5 benchmark functions are selected. These functions are:

Sphere:

$$f_1(\vec{x}) = \sum\nolimits_{i=1}^{n} x_i^2 \qquad x_i \in [-100,100]$$

Rosenbrock:

$$f_2(\vec{x}) = \sum\nolimits_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1-x_i)^2 \qquad x_i \in [-2.048,2.048]$$

Rastrigin:

$$f_3(\vec{x}) = \sum\nolimits_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \qquad x_i \in [-5.12,5.12]$$

Ackley:

$$f_4(\vec{x}) = -20\exp\left(-0.2\sqrt{1/_n \cdot \sum\nolimits_{i=1}^{n} x_i^2}\right) \qquad x_i \in [-32.76,32.76]$$

$$-\exp(1/_n \cdot \sum\nolimits_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$$

Schwefel:

$$f_5(\vec{x}) = 418.9829n + \sum\nolimits_{i=1}^{n} x_i \sin(\sqrt{|x_i|}) \qquad x_i \in [-512.03,511.97]$$

The global objective value of all functions is equal to zero. For all functions, the number of variables is considered as 5 ($n = 5$). All functions are optimized in the absence of any constraints, with the exception of the variable ranges constraints. We assume that when an algorithm could reach a value less than the target value, it has reached the global optimum. The target value for the first four functions is 1e-6 and for the last function (Schwefel function) is 6.3638E-5.

Comparison is done between LCA and the highly recognized particle swarm optimization (PSO) algorithm [7]. Due to popularity of PSO algorithm and to keep the paper short, we abstain to give further introduction to this algorithm. The parameters of PSO algorithm are set as follows: The inertia weight varies from 0.9 to 0.1 linearly with the iterations. The learning factors, φ_1 and φ_2 are set to be 2. The upper and lower bounds for velocities, (v_{min}, v_{max}) are set to be the maximum upper and lower bounds of the variables range. If the sum of accelerations would cause the velocity on one dimension to exceed v_{min} or v_{max}, then the velocity on that dimension is limited to v_{min} or v_{max}, respectively. The number of particles in the population space is set as 10. The maximum iteration number is set as 9000, yielding the total number of 90000 function evaluations. The parameters of LCA are set as follows: $L = 10$; $S = 1000$; $c_1 = 0.5$; $c_2 = 0.5$; $p_C = 0.01$.

Results obtained by the algorithms are reported in Table I. In this table, results are based on the mean and standard deviation of the lowest function values obtained in 10 runs of the algorithms on each problem. Since the Sphere function is a unimodal function, it not difficult for algorithms to find the global minimum. From Table I it is revealed that both algorithms could reach the global minimum in each run. The global optimum in Rosenbrock function is inside a long narrow valley and this makes the convergence to the global minimum difficult. Although both LCA and PSO could not reach the global optimum, LCA gets very close to the global optimum (the worst value obtained by LCA among its 10 runs is 6.91E-05). However, PSO stops far from optimum. The Rastrigin function is a multimodal function with very local optima. Although LCA could escape successfully from local optima in each run, PSO trapped several times. Like Rastrigin function, Ackley's function has many local optima. Yet, the complexity of this function is fair and both LCA and PSO are fully able to reach the global minimum in each run. Schwefel function has many waves and many algorithms are trapped in the second best minimum of this function. Once again LCA could reach the global minimum at each run, while PSO fails to do this. The very low standard deviations signify that LCA is more dependable and converges to the global minimum at each of its runs.

TABLE I. Mean and standard deviation of the best function values obtained by LCA and PSO.

Function	LCA		PSO	
	Mean	*Standard deviation*	*Mean*	*Standard deviation*
$f_1(\vec{x})$	0	0	0	0
$f_2(\vec{x})$	3.140E-05	2.425E-05	0.0837	0.0610
$f_3(\vec{x})$	0	0	0.0994	0.3146
$f_4(\vec{x})$	0	0	0	0
$f_5(\vec{x})$	6.3638E-5	0	11.8438	37.4534

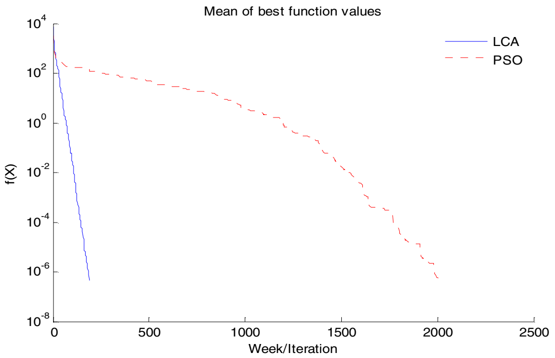

Figure 3. Evolution of best function value for Sphere function.

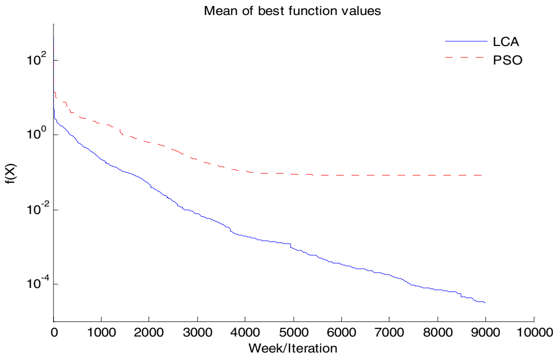

Figure 4. Evolution of best function value for Rosenbrock function.

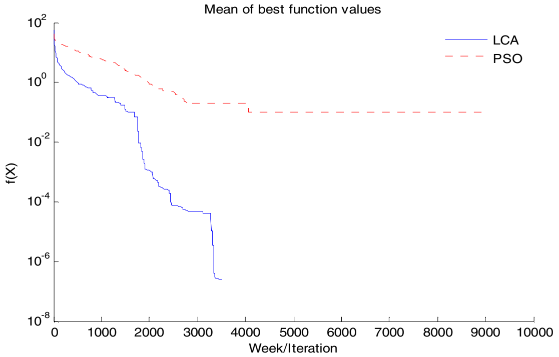

Figure 5. Evolution of best function value for Ratrigin function.

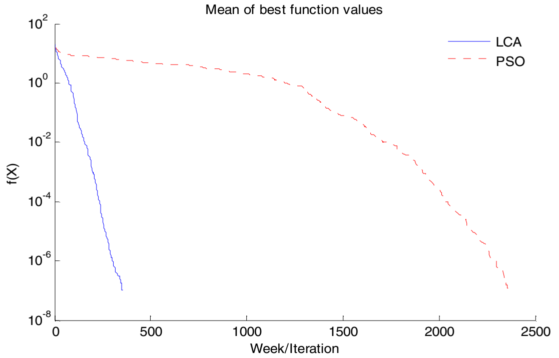

Figure 6. Evolution of best function value for Ackley function.

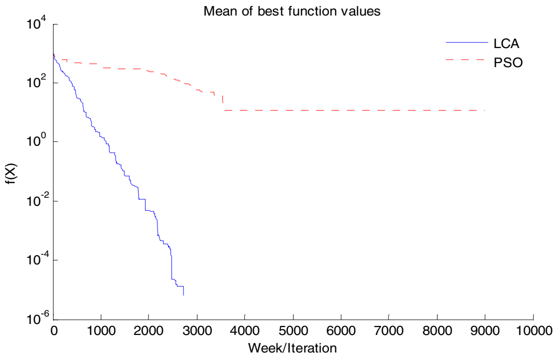

Figure 7. Evolution of best function value for Schwefel function.

To visualize how LCA and PSO converge to the global minimum, the evolution of the mean of best function values during the searches is depicted based on logarithmic scaling in Figs. 3 to 7 for different functions. It is clear that LCA converges to the global minimum faster than PSO, which indicates the less number of functions evaluated by LCA.

V. CONCLUSIONS

An optimization algorithm was proposed inspired by the competition of sport teams in a sport league. A population of individuals as artificial teams play in an artificial league for a number of seasons based on the win/loss system. At each week, teams devise a playing/formation strategy (new solution) based on their current best formation (the best solution obtained so far by the corresponding team) and post match SWOT analysis. From the experiments on finding the global optimum of five benchmark functions, it was revealed that the new algorithm is a dependable tool and converges very fast to the global minimum.

For future research, there would be a great interest on testing the performance of the new algorithm on constraint optimization problems and real word engineering problems.

ACKNOWLEDGMENT

The author would like to acknowledge the Iran National Elite Foundation for the financial support of this work.

REFERENCES

[1] C. Blum, and A. Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison,” ACM Comput Surv, vol. 35, pp. 268–308, 2003.

[2] A. P. Engelbrecht, Computational intelligence, an Introduction. John Wiley & Sons. Chichester, England, 2007.

[3] http://en.wikipedia.org/wiki/ Sports_league.

[4] http://www.talkfootball.co.uk/guides/football_ formations.html.

[5] E. Bruke, D. de Werra, and J Kingstone, “Colorings and related topics; Applications to timetabling,” in Handbook of graph theory, J. L. Gross, and J. Yellen, Eds.CRC PRESS, 2004.

[6] A. Husseinzadeh Kashan, B. Karimi, and F. Jolai, “Effective hybrid genetic algorithm for minimizing makespan on a single-batch-processing machine with non-identical job sizes,” Int J Prod Res, vol 44, pp 2337-2360, 2006.

[7] J. Kennedy, and R.C. Eberhart, “Particle swarm optimization,” Proc. IEEE International conference on Neural Networks, 1995, pp. 1942-1948.