Artif Intell Rev
DOI 10.1007/s10462-017-9587-x

# Sports inspired computational intelligence algorithms for global optimization

**Bilal Alatas**[1]



**Abstract** Many classical search and optimization algorithms are especially insufficient in solving very hard large scale nonlinear problems with stringent constraints. Hence, computational intelligence optimization algorithms have been proposed and used to find well-enough solutions at a reasonable computation time when the classical algorithms are not applicable or do not provide good solutions to these problems due to the unmanageable search space. Many existing algorithms are nature-inspired, which work by simulating or modeling different natural processes. Due to the philosophy of continually searching the best and absence of the most efficient method for all types of problems, novel algorithms or new variants of current algorithms are being proposed and seem to be proposed in future to see if they can cope with challenging optimization problems. Studies on sports in recent years have shown that processes, concepts, rules, and events in various sports can be considered and modelled as novel efficient search and optimization methods with effective exploration capabilities in many cases, which are able to outperform existing classical and computational intelligence based optimization methods within different types of search spaces (Kashan in Appl Soft Comput 16:171–200, 2014; Bouchekara in Oper Res 1–57, 2017; Razmjooy in J Control Autom Electr Syst 1–22, 2016; Osaba et al. in Appl Intell 41(1):145–166, 2014a, Sci World J, 2014b). These novel and interesting sports based algorithms have shown to be more effective and robust than alternative approaches in a large number of applications. In this work, all of the computational intelligence algorithms based on sports and their applications have been for the first time searched and collected. Specific modelling of real sport games for computational intelligence algorithms and their novelties in terms of comparison with alternative existing algorithms for optimization have been reviewed with specific characteristics, computational implementation details and main applications capabilities, in the frame of hard optimization problems. Information is given about these search and optimization algorithms such as League Championship Algorithm, Soccer League Optimization, Soccer Game Optimization, Soccer League Competition Algorithm, Golden Ball Algorithm, World Cup Optimization, Football Optimization Algorithm, Football Game Inspired Algorithm, and

✉ Bilal Alatas
balatas@firat.edu.tr

1 Department of Software Engineering, Firat University, Elazig, Turkey

Published online: 26 October 2017

Most Valuable Player Algorithm. Performance comparison of these sports based algorithms and other popular algorithms such as Genetic Algorithm, Particle Swarm Optimization, and Differential Evolution within unconstrained global optimization benchmark problems with different characteristics has been performed for the first time. A general evaluation has also been discussed with further research directions.

**Keywords** Computational intelligence · Global optimization · Sports inspired optimization

## 1 Introduction

Classical optimization algorithms need mathematical models for the interested problem. Ordinarily, construction of mathematical models may be hard for many complicated systems. Even though the model is constructed, it cannot be efficiently used due to very long run time. Classical search and optimization algorithms can be insufficient in many problems. Such algorithms are ineffective in adapting a general solution method. In most cases, these algorithms modify the interested problem and the solution found by these algorithms are the solution of the changed or modified problem. These modifications can be difficult and meaningless. Interested problem is modelled the way that classical algorithms can handle it. Solution strategies of classical search and optimization algorithms usually depend on type of objective function, constraint functions, and variables used in modelling. Furthermore, effectiveness of the classical algorithms highly depends on the solution space, the number of decision variables, and the number of constraints in problem modelling (Akyol and Alatas 2017). If there exist different types of decision variables, objective functions, and constraint functions, general solution strategy cannot be obtained for problem. In other words, most of algorithms solve models which have certain type of objective or constraint functions. However, problems in many different fields require concurrently different types of decision variables, objective and constraint functions. Therefore, computational intelligence algorithms are proposed and efficiently used. These adaptable algorithms have become very popular in recent years, due to their ability for providing a reasonable tradeoff between computing time and achievable solution quality (Salcedo-Sanz 2016; Akyol and Alatas 2017).

No Free Lunch theorem opens the door to design and develop many different computational intelligence algorithms that exploit problem-specific knowledge to acquire good performance over certain classes of optimization problems. Due to the philosophy of continually searching the best and absence of the most efficient computational intelligence method for all types of problems, novel algorithms or new variants of current algorithms are being proposed and seem to be proposed in future (Ozbay and Alatas 2016a). In this context, modern computational intelligence algorithms have been proposed for solving the aforementioned lack of efficient methods. One important current trend in computational intelligence consists of constructing novel searching mechanisms based on sports. These sports based algorithms have shown to be more effective and robust than alternative approaches in a large number of applications. The novelty of this approach is that; processes, concepts, and events in sports can be used in this context as inspiring guide for constructing effective search and optimization algorithms, which follow the rules in sport games. This work aims to review the most important concepts of novel existing sports-based computational intelligence algorithms, their specific characteristics, computational implementation details and main applications capabilities, in the frame of hard optimization problems. Performance comparison of these sports based algorithms and other popular algorithms such as Genetic Algorithm, Particle Swarm Optimization, and Differential Evolution within unconstrained global optimization benchmark problems with different characteristics has been performed for the first time.

The organization of this paper is as follows: Sect. 2 describes the computational intelligence based search and optimization methods. In Sect. 3, all of the sports based algorithms and their applications have been firstly introduced and analyzed. Section 4 describes the benchmark functions selected for performance comparisons of algorithms. Experimental results obtained from sports based algorithms and other algorithms have been explained in Sect. 5. Section 6 is a general evaluation of the algorithms and Sect. 7 concludes the paper along with further research directions.

## 2 Computational intelligence search and optimization algorithms

Adaptable and general purposed population based computational intelligence optimization algorithms have become a strong and popular part of modern optimization methods due to their efficient computing power and easy transformations. They are simple in terms of comprehensibility. They are easy to combine with other optimization methods and they provide very effective solutions of dealing with complex problems solution space of which are infinite or too large (Ozbay and Alatas 2016a). These methods are general purposed solution algorithms and utilize the information being gathered to guide the search towards the global optimum. They gain significant interest in science and engineering applications and are efficiently used in complex, high dimensional, multimodal, and nonlinear search and optimization problems due to their general applicability. They are population based methods and start the search within the search space with a number of candidate points instead of starting from a single point. They are capable of escaping from a local optima by iteratively producing new and better candidate solutions with different generation and operation rules of these methods and finally conclude to obtain best solution to objective function. In recent years, they have been very popular due to their good computation power and easily transformations (Ozbay and Alatas 2016b).

Computational intelligence optimization algorithms provide general solution strategies when the mathematical models cannot be derived or when derived model has different types of decision variables, objective functions, and constraint functions. These algorithms do not depend on the search space, the number of decision variables, and the number of constraint functions. They are efficiently used to find the optimum solutions within an acceptable time for the problems solution time of which are too long (Alatas 2011). Their computing power is good transformations and adaptations are easy. These algorithms do not require the assumptions and do not change the problem. They are adapt themselves for solving different types of search and optimization problems. Due to many advantages, these algorithms are densely being used in many different fields. Selecting an appropriate algorithm for a certain problem is not trivial, as problem characteristics can change significantly for different instances and the performance of a computational intelligence algorithm may vary remarkably for different parameters. Due to the philosophy of continually searching the best and absence of the most efficient computational intelligence algorithm for all types of problems, state-of-the-arts algorithms or new variants of existing algorithms are being proposed (Akyol and Alatas 2016a, b). All of these algorithms are population based iterative methods and work as shown in Fig. 1.

General purposed computational search and optimization methods can be categorized according to ten different inspiration fields such as biology, physics, swarm, sociology, music, chemistry, sports, plant, water, and mathematics. Furthermore, there are hybrid methods which are combination of these. Categorization is depicted in Fig. 2.

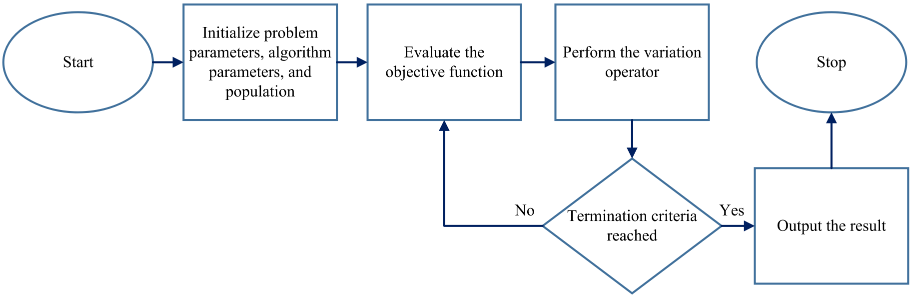


**Fig. 1** General flowchart of population based computational intelligence optimization algorithms

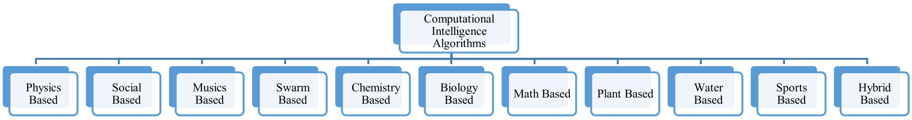


**Fig. 2** Computational intelligence search and optimization methods

Physics based metaheuristic optimization methods mimic the physical rules (Can and Alatas 2015). The most popular physics based methods are magnetic optimization algorithm (Tayarani-N and Akbarzadeh-T 2008), gravitational search algorithm (Rashedi et al. 2009), magnetic charged system search (Kaveh 2014), ions motion optimization (Javidy et al. 2015), electromagnetism-like algorithm (Birbil and Fang 2003), central force optimization algorithm (Xing and Gao 2014a, b), space gravitational algorithm (Hsiao et al. 2005), particle collision algorithm (Sacco and De Oliveira 2005), galaxy based algorithm (Shah-Hosseini 2011), big bang-big crunch algorithm (Genc et al. 2010), big crunch algorithm (Kripka and Kripka 2008), integrated radiation algorithm (Chuang and Jiang 2007), charged system search algorithm (Xing and Gao 2014a, b), and artificial physics algorithm (Xie et al. 2010).

Social based optimization algorithms are inspired from behaviors of people, human learning mechanism, and many features associated with social situation of the people (Ozbay and Alatas 2015; Kiziloluk and Alatas 2012). Some popular social based optimization algorithms can be listed as: Imperialist competitive algorithm (Atashpaz-Gargari and Lucas 2007), parliamentary optimization algorithm (Borji and Hamidi 2009), teaching learning based optimization (Rao et al. 2012), social emotional optimization algorithm (Xu et al. 2010), brain storm optimization algorithm (Shi 2011), group leaders optimization algorithm (Daskin and Kais 2011), hierarchical social algorithm (Duarte et al. 2004), human group formation algorithm (Thammano and Moolwong 2010), and social based algorithm (Ramezani and Lotfi 2013).

Harmony search (Geem et al. 2001; Rezoug and Boughaci 2016), melody search (Ashrafi and Dariane 2011), and musical composition algorithm (Mora-Gutiérrez et al. 2014) are music based methods (Ozbay and Alatas 2016c). Chemical reaction optimization (Lam and Li 2010) and artificial chemical reaction optimization algorithm (Alatas 2011) are chemistry based methods. Genetic algorithm (Holland and Goldberg 1989) and clonal selection algorithm (Castro and Zuben 2002) are well known biological based optimization methods.

Base optimization algorithm (Salem 2012), sine-cosine algorithm (Mirjalili 2016), and matheuristics (Maniezzo et al. 2009) are mathematics based methods.

Particle swarm optimization (Kennedy and Eberhart 1995), chicken swarm optimization (Meng et al. 2014), cat swarm optimization (Chu et al. 2006), whale optimization (Mirjalili and Lewis 2016), ant colony algorithm (Dorigo et al. 1991), firefly algorithm (Wang et al. 2016; Gálvez and Iglesias 2016) are some of the swarm based optimization algorithms developed by drawing inspiration from swarm intelligence systems in nature.

Plant based algorithms have been developed by inspiration from plant intelligence (Akyol and Alatas 2017). Flower pollination algorithm (Yang 2012), invasive weed optimization (Mehrabian and Lucas 2006), paddy field algorithm (Premaratne et al. 2009), root mass optimization algorithm (Qi et al. 2013), artificial plant optimization algorithm (Zhao et al. 2011), sapling growing up algorithm (Karci and Alatas 2006), photosynthetic algorithm (Murase 2000), plant growth optimization (Cai et al. 2008), root growth algorithm (Zhang et al. 2014), strawberry algorithm (Salhi and Fraga 2011), runner root algorithm (Merrikh-Bayat 2015), path planning algorithm (Zhou et al. 2016), and rooted tree optimization (Labbi et al. 2016) can be categorized as plant based.

Water based algorithms have been proposed inspiring by the intelligent movements of waters (Ozbay and Alatas 2016b). Intelligent water drops algorithm (Shah-Hosseini 2009), river formation dynamics algorithm (Rabanal et al. 2007), water flow-like algorithm (Yang and Wang 2007), water flow algorithm (Kamarudin et al. 2016), water cycle algorithm (Sadollah et al. 2015), water evaporation optimization (Kaveh and Bakhshpoori 2016), simulated raindrop algorithm (Ibrahim et al. 2014), water wave optimization (Zheng 2015), circular water wave algorithm (Colak and Varol 2015), and artificial showering algorithm (Ali et al. 2015) are interesting search and optimization algorithms based on water.

Processes, concepts, rules, and events in various sports, especially in football, have also been considered and modelled as novel efficient search and optimization methods with effective exploration capabilities in many cases (Kashan 2014; Bouchekara 2017; Razmjooy et al. 2016; Osaba et al. 2014a, b). League championship algorithm, soccer league optimization, soccer game optimization, soccer league competition algorithm, golden ball algorithm, world cup optimization, football optimization algorithm, and football game inspired algorithm are sports based computational intelligence algorithms.

Although there exist many efficient optimization methods in the literature; design, development, and implementation of novel methods are momentous tasks for always obtaining better. According to the free lunch theorem, there is not a single best algorithm for all types of problems. That is why, new variants of existing algorithms or novel computational intelligence algorithms are often proposed.

## 3 Sports inspired computational intelligence algorithms

In essence, researchers try to look for efficient algorithm (A) as the following generic scheme:

$$\begin{aligned}&[x_1, x_2, \ldots x_n]^{t+1}\\&\quad = A\left\{[x_1, x_2, \ldots x_n]^t, \left(pp_1, pp_2, \ldots pp_j\right), (ap_1, ap_2, \ldots ap_k), (r_1, r_2, \ldots r_m)\right\} \quad (1)\end{aligned}$$

which attempts to generate better solutions (a population of $n$ solutions) at iteration $t+1$ from the current iteration $t$ and its solution set $x_i$, $(i = 1, 2, \ldots, n)$. This iterative algorithm also uses some algorithm-dependent parameters $(ap_1, \ldots, ap_k)$, problem-dependent parameters $(pp_1, \ldots, pp_j)$, and some random variables $(r_1, \ldots, r_m)$.

Especially, computational intelligence search and optimization algorithms are used instead of analytical approaches, which make many assumptions on the problem surface, when the problems are incomplete, noisy, and non-continuous. Historically, many of the most successful computational intelligence approaches have had a biological inspiration, such as

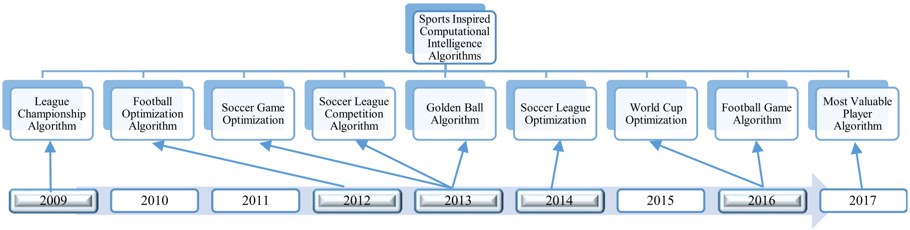


**Fig. 3** Sports based computational intelligence optimization algorithms

evolutionary computation or swarm intelligence paradigms, however in the last few years new approaches based on sports modeling have been proposed and applied with success. Processes, concepts, rules, and events in sports, modeled as computational intelligence algorithms, are able to produce completely new and effective search and optimization procedures, with effective exploration capabilities in many cases, which are able to outperform existing classical and metaheuristic based optimization approaches (Kashan 2014; Bouchekara 2017; Razmjooy et al. 2016; Osaba et al. 2014a, b). This section describes all of the existing computational intelligence algorithms based on sports.

All of the computational intelligence search and optimization algorithms which have been developed inspiring from sports have been depicted in Fig. 3 and explained in subsections.

## 3.1 League championship algorithm (LCA)

### *3.1.1 Overview*

LCA is the first proposed interesting sports inspired population based algorithm for global optimization which tries to imitate a championship situation where artificial football clubs participate in an artificial league for a number of weeks (Kashan 2009, 2014).

Each team (individual) in the league (population) represents a candidate solution for interested problem. These teams compete in an artificial league for several weeks (iterations). Based on the league schedule at each week, teams play in pairs and the outcome is determined in terms of win or loss based on each team playing strength (fitness value) resultant from a particular team formation (solution). In the recovery period, keeping track of the previous week events and using SWOT (Strength, Weakness, Opportunity, and Threats) analysis, each team changes its formation (a new solution) for the next week contest and the championship goes on for a number of seasons (termination criterion). Table 1 shows the terms that have same meaning for evolution and sports terminologies.

### *3.1.2 Algorithm*

LCA is the first proposed sports based computational intelligence algorithm and works with a population of candidate solutions. A league consists of $L$ teams is generated and their playing strengths are computed in the initialization step. Each team has $n$ players corresponding to the number of decision variables of the related problem. In this step, the teams' best formations take the initial setting values. Then, competition step is started. Based on the league schedule, the artificial teams compete in pairs for $S\times(L-1)$ weeks where $S$ is the number of seasons. After each competition between team $i$ and team $j$ the outcome is given in terms of win or loss.

**Table 1** Metaphors

| | Sports terminology | LCA |
|---|---|---|
| 1 | League | Population |
| 2 | Team | Individual |
| 3 | Player's strength in each team's formation | Decision variable |
| 4 | Playing strength | Fitness value |
| 5 | Formation | Solution |
| 6 | Week | Iteration |
| 7 | Number of seasons | Maximum iterations |
| 8 | Match analysis process | Selection |

LCA performs SWOT analysis for the variation. In the recovery step, each team elaborates a new formation taking into account the team's current best formation and previous week events. The selection in LCA is performed with a greedy approach. It replaces the current best formation with the new more productive team. LCA stops after the termination criterion is met and the team who has the best fitness value (playing strength) wins the championship (Kashan 2009, 2014).

Each team consists of $D$ individuals. Algorithm attempts to move a league of $D$ possible solutions to promising areas of the search space in successive S seasons according to its operators by maintaining constant league size. Assume that the formation of team $i$ ($i \in \{1, 2, \ldots, L\}$) in round $t$ ($t \in \{1, 2, \ldots, S \times (L-1)\}$) is $X_i^t = \left\{x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t\right\}$ and the best formation of team $i$ until iteration $t$ is $B_i^t = \left\{b_{i1}^t, b_{i2}^t, \ldots, b_{iD}^t\right\}$. Before each season, the match schedule is preset. A single round-robin schedule is followed in which each team competes against every other team once in each season. For a sport league consisting of $L$ teams, the single round-robin tournament requires $L\,/\,2$ matches being run in parallel in each of $(L-1)$ weeks when $L$ is even.

In order to win the matches with other teams, each team should design the formation according to the competitor which is played against. Hypothetically, team $i$ has competed against team $j$, and team $l$ against team $k$ at round $t$. For round $t+1$ (team $i$ against team $l$), team $i$ will update the formation according to the outcomes of its own and team $l$ at the previous round using updating Eq. 2.

$$x_{id}^{t+1} = \begin{cases} b_{id}^t + y_{id}^t \left(c_1 r_1 \left(x_{id}^t - x_{kd}^t\right) + c_1 r_2 \left(x_{id}^t - x_{jd}^t\right)\right) & if\ f\left(X_i^t\right) > f\left(X_j^t\right) \cap f\left(X_l^t\right) > f\left(X_k^t\right) \\ b_{id}^t + y_{id}^t \left(c_2 r_1 \left(x_{kd}^t - x_{id}^t\right) + c_1 r_2 \left(x_{id}^t - x_{jd}^t\right)\right) & if\ f\left(X_i^t\right) > f\left(X_j^t\right) \cap f\left(X_k^t\right) > f\left(X_l^t\right) \\ b_{id}^t + y_{id}^t \left(c_1 r_2 \left(x_{id}^t - x_{kd}^t\right) + c_2 r_1 \left(x_{jd}^t - x_{id}^t\right)\right) & if\ f\left(X_j^t\right) > f\left(X_i^t\right) \cap f\left(X_l^t\right) > f\left(X_k^t\right) \\ b_{id}^t + y_{id}^t \left(c_2 r_2 \left(x_{kd}^t - x_{id}^t\right) + c_2 r_1 \left(x_{id}^t - x_{jd}^t\right)\right) & if\ f\left(X_j^t\right) > f\left(X_i^t\right) \cap f\left(X_k^t\right) > f\left(X_l^t\right) \end{cases} \tag{2}$$

where $c_1$, $c_2$ represent the preset weights in the range of [0,1]; $r_1$, $r_2$ are random numbers distributed uniformly in the range of [0,1]; $y_{id}^t$ is binary number which determines if $d$th dimension of $X_i^t$ is updated. The updating amplitude of team $i q_i^t$ is

$$q_i^t = \sum_{d=1}^{D} y_{id}^t \tag{3}$$

$q_i^t$ is calculated by using the truncation geometric distribution method (Kashan 2011) which is defined as in Eq. 4.

$$q_i^t = \left\lceil \frac{\ln(1 - \left(1 - (1 - p_c)^D\right) r}{\ln(1 - p_c)} \right\rceil \tag{4}$$

where $r$ is random number in the range of [0,1], $p_c$ is the preset parameter in the range of (0,1).

Flowchart of LCA is shown in Fig. 4.

### *3.1.3 Related works*

LCA was firstly proposed in 2009 as a conference paper (Kashan 2009) and then its extended version was published in an esteemed journal in 2014 (Kashan 2014). After the introduction of LCA, some researchers tried to apply it in different research areas while some tried to improve it for obtaining efficient results within different types of problems. In 2010, Kashan and Karimi (2010) have adapted the LCA for constrained optimization problems and tested its performance on 13 constrained optimization problems. They have reported that LCA is a competitive method for solving these problems. LCA has been proposed as search method for constrained mechanical engineering design optimization problems in Kashan (2011). LCA has been tested on a set of 22 constrained benchmark functions and on 5 well-studied engineering design optimization problems and its superiority has been reported in this work.

Discrete version of LCA has been proposed and used for single machine earliness–tardiness scheduling problem with batch delivery costs and distinct due dates (Pourali and Aminnayeri 2011). Efficiency and potency of the LCA have been verified on 48 test problems. In (Kashan et al. 2012), original LCA has been modified via modeling a "between two halves like analysis" beside the post-match analysis in order to improve the convergence. They have named the algorithm as RLCA (R for realistic) and tested it on 5 benchmark functions. From the simulation results, it has been reported that RLCA is faster than original LCA in finding the same optimum point.

Sun et al. (2013) have proposed neural network and LCA based approach to obtain optimal allocation with the market surplus and total reputation goals in cloud computing. Feasibility has been validated and superiority on improving both market surplus and success ratio has been demonstrated. Lenin et al. (2013) have proposed a new LCA to solve objective constrained reactive power dispatch problem. The efficiency of the method is tested on IEEE 30-bus system and better results of LCA has been reported. Kejani (2013) has adapted LCA for reliability optimization and obtained competitive results.

LCA has been used for Algerian optimal power system network flow (Bouchekara et al. 2014). In this work, authors have implemented and applied LCA to the Algerian 59-bus power system for 9 different cases with different objectives. Effectiveness and the robustness of LCA have been reported. Sajadi et al. (2014) have used LCA for task sequencing and scheduling in a permutation flow-shop system. The simulation results have shown the dominance of LCA over genetic algorithm, particle swarm optimization, and differential evolution. Bouchekara et al. (2014b) have adapted LCA for optimization of electromagnetic devices. Kahledan (2014) has applied LCA for an NP-hard problem, namely travelling salesman problem, and obtained effective results compared to other known methods. Edraki (2014) has proposed a LCA based method for engineering design optimization of centrifuge pumps. Abdulhamid and Abd Latiff (2014) have applied LCA in job scheduling for infrastructure as a service cloud. Abdulhamid et al. (2014) have proposed LCA based make span time minimization

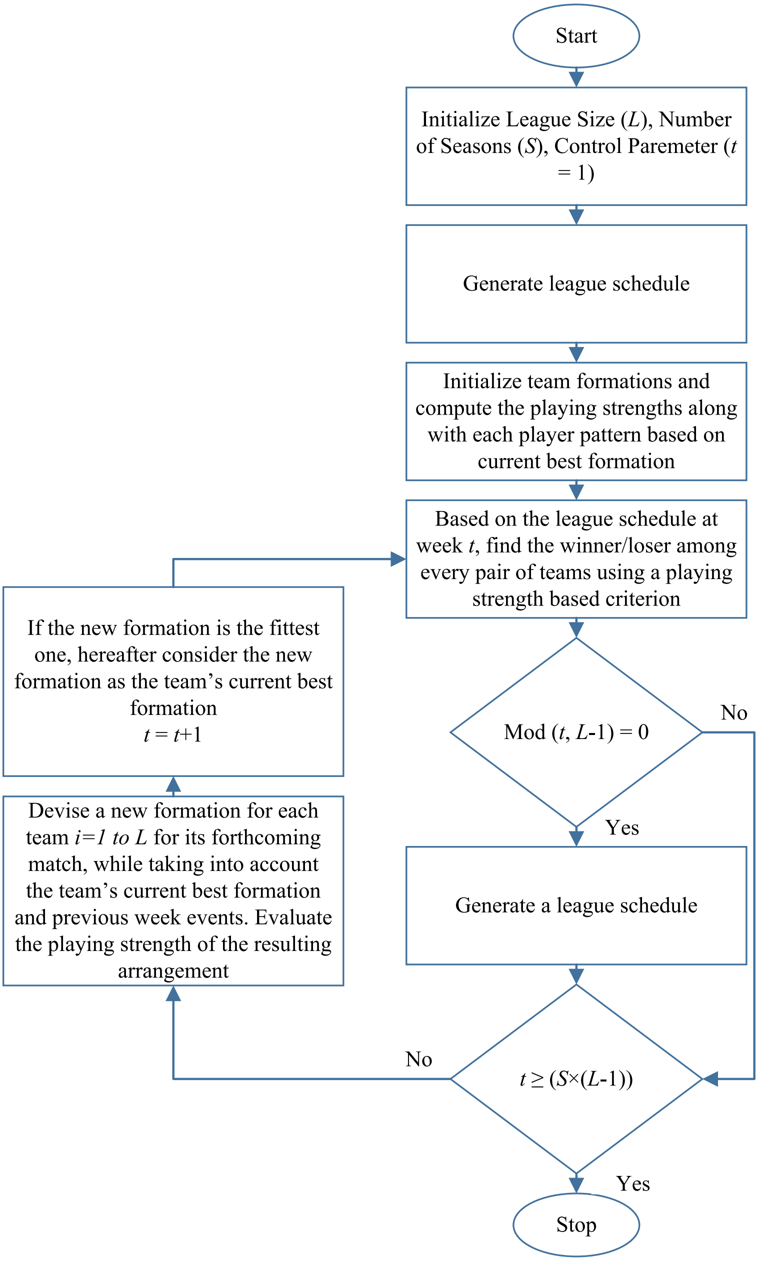


**Fig. 4** Flowchart of LCA

scheduling technique in IaaS cloud and reported better results in minimizing the make span time of scheduled tasks in IaaS cloud.

Eyvazi (2015) has used LCA as a search method for portfolio optimization problem with multi-period investment. Badrloo (2015) has used LCA in permutation based combinatorial optimization problems. Abdulhamid and Abd Latiff (2015) have improved the classical LCA for job scheduling technique for infrastructure as a service cloud. Group based LCA has been proposed for machine-part cell formation problem by Seyedhosseini et al. (2015). LCA has been used to predict the production scheduling performance in a shop floor by Xu et al. (2015a). Same authors have improved LCA by changing its parameters linearly with iteration and by designing a new match schedule, and by introducing a free search operator (Xu et al. 2015b). They employed LCA for optimizing the parameters of neural network for establishing the shop floor production scheduling model and obtained good results.

Bingol and Alatas (2016) have embedded the chaotic maps into LCA and proposed six novel chaotic LCAs in order to improve the convergence speed. Abdulhamid et al. (2016) have continued and improved their earlier presented researches and proposed a novel scientific application tasks scheduling technique for the cloud computing service using global LCA. Nedaie and Khoshalhan (2016) have proposed a new play-off approach for LCA by dropping off for each last ranked team at the end of each season in order to efficiently solve the large scale problems. Based on modeling the tie concept, two new versions of the LCA have been proposed by Jalili et al. (2016). They have used these LCA versions for optimum truss design and tested them on 5 well-known truss design problems with various constraints. The most recent work about LCA is performed by Saraswathi and Srinivasan (2017). They have proposed LCA to optimize the number of neurons in the hidden layers of ensembled fully complex valued relaxation network classifier for diagnosing the abnormalities in digital mammograms by extracting the curvelet fractal textures. They have obtained good classification accuracy.

All of the works performed about LCA have been demonstrated in Fig. 5.

## 3.2 Soccer league optimization

### *3.2.1 Algorithm*

Soccer League Optimization (SLO) is also an interesting algorithms based on the performance of the major football leagues within each season in European countries (Khaji 2014). The algorithm starts the search and optimization with an initial population which consists of three different teams: the strongest, the regular, and the poorest. Each individual in the population represents football team and each player represents a player. The strongest teams usually purchase the best players of the regular teams and in turn, regular teams purchase the best players of the poorest who should always discover young players instead of buying professionals. The best players are usually sold to the wealthiest clubs. The poorest clubs are financially limited which make them to discover young players and train them without paying for new players. The last sorts of teams are the ones which have a combination of these two policies for players. They buy good players of the poorest and worse players of the wealthiest teams. The system within the time, improve the whole football system and finally, there are few perfect clubs who enjoy the most perfect outcomes of the rest clubs and buy them (Khaji 2014). Comparisons of terms in real game and SLO have been shown in Table 2. In the unique paper that introduces SLO, formal mathematical formulations have not been provided and it is hard to understand and apply its operators. Figure 6 demonstrates the flowchart of SLO.

### *3.2.2 Related works*

In the literature, there is only one paper about SLO and it introduces the algorithm (Khaji 2014).

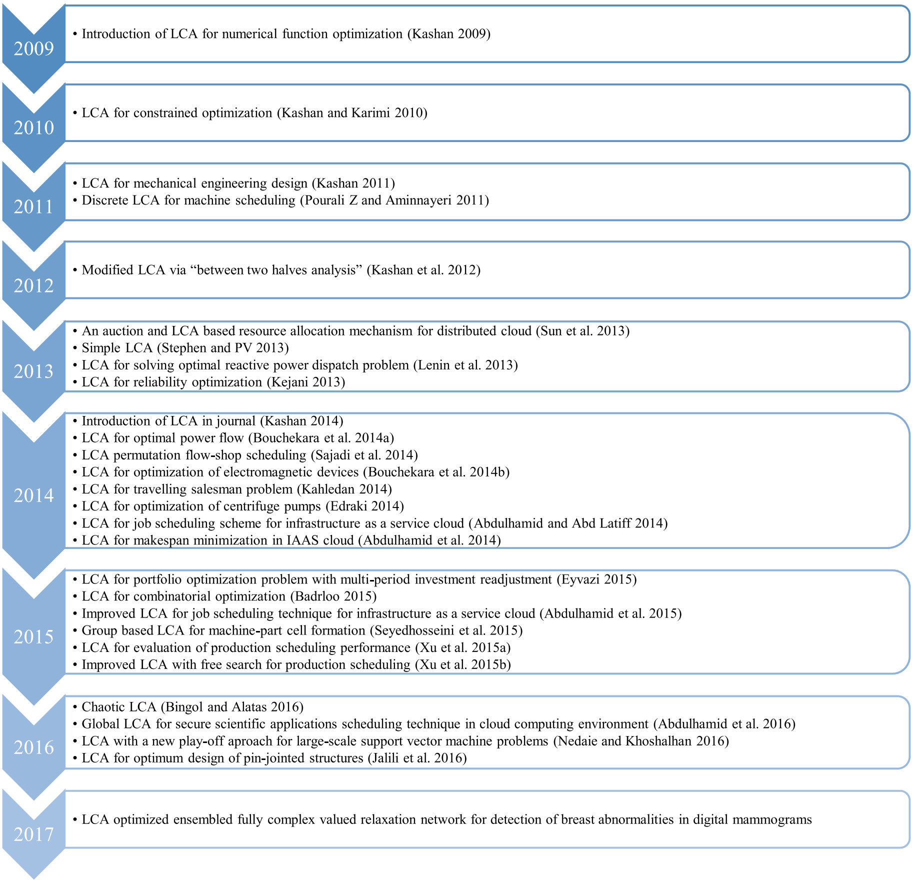


**Fig. 5** LCA in the literature

**Table 2** Comparisons

| | Game | SLO |
|---|---|---|
| 1 | Player | Cell of number |
| 2 | Team | Array of numbers |
| 3 | Training a player | Adding a range of random numbers to cell |
| 4 | Discovering a young player | Generating a random number |
| 5 | Post in a team | Dimension |

## 3.3 Soccer game optimization

### *3.3.1 Algorithm*

Soccer Game Optimization (SGO) proposed by Purnomo and Wee (2013) is another sports inspired computational intelligence algorithms and mimics the behavior movements of play-

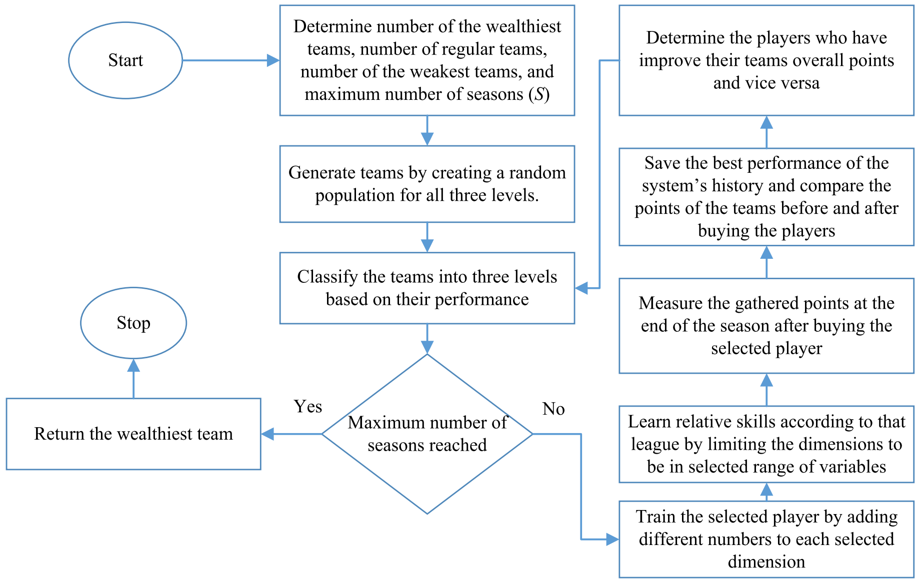


**Fig. 6** Flowchart of SLO

ers during soccer games. SGO is a simplified a soccer game player's movements. Many terms used in the SGO are derived from the soccer game and have been listed in Table 3.

In SGO; ball dribbler, a player with the most advantageous position, dribbles the ball and represents the best solution found so far. All players can use the globally shared position of the ball dribbler. The ball dribbler can pass the ball to another player or holds the ball. For balancing the diversification and intensification mechanism, two main operators called 'move off' and 'move forward' movements are performed. Diversification is managed by 'move off' operator which is mainly used to explore the search space using randomness. This operator of SGO aims to minimize the premature convergence. Intensification is controlled by 'move forward' operator which is mainly used to explore the search space nearby a player. This movement is determined by the cooperation or interaction between the player's current position, the player's best position and the other players' positions (Purnomo and Wee 2013).

In SGO, a team $P_0$ consists of $p$ players $P_0 = \left\{X_0^1, X_0^2, \ldots X_0^p\right\}$ encoding a potential solution and substitute players $S_0 = \left\{X_0^{p+1}, X_0^{p+2}, \ldots X_0^{p+s}\right\}$ encoding a solution in a set of best solutions found so far. Initially $P_0$ and $S_0$ are randomly generated. The best player in $P_0$ is called as ball dribbler $B_0$. At each iteration, players move in to new positions by move off and move forward according to determined probability. In move forward, player moves towards the ball dribbler and new position of player $X_t^i$ is computed according to the previous position $X_{t-1}^i$, the knowledge of player $X_b^i$ and the position of ball $B_{t-1}$. This operator is applied as in Eq. 5 where $w$s are selected weights using the center of mass principle. In move off, player explores the search space by randomly moving regardless of the other player positions. After move off, the move forward operator may be applied to the player with the determined probability.

$$X_t^i = \frac{w_1 X_{t-1}^i + w_2 X_b^i + w_3 B_{t-1}}{w_1 + w_2 + w_3} \tag{5}$$

**Table 3** Terms in SGO

| | Game | SGO |
|---|---|---|
| 1 | Player | Candidate solution |
| 2 | Player's position | Set of decision variables |
| 3 | Team | Simultaneous set of candidate solutions |
| 4 | Kick | Iteration |
| 5 | Ball | Best solution |
| 6 | Soccer field | Feasible search space |
| 7 | Soccer rules | Constraints |
| 8 | Goal | Optimum solution |

After the new positions of players have been computed, ball dribbler is updated according to the new objective functions of the players as shown in Eq. 6 for a minimization problem.

$$B_t = \begin{cases} x_t^i & if\ x_t^i < B_{t-1}\ i\ \in candidate\ of\ ball\ dribbler\ \ (best\ player\ in\ the\ iteration) \\ B_{t-1} & otherwise \end{cases} \tag{6}$$

In each SGO iteration, a player is replaced by a substitute player according to the objective functions of these players with a selected probability. The players update their best position $x_{b,t}^i$ as in Eq. 7.

$$x_{b,t}^i = \begin{cases} x_t^i & if x_t^i < x_{b,t-1}^i \\ x_{b,t-1}^i & otherwise \end{cases} \tag{7}$$

The substitute players also update their knowledge based on the current positions of players as in Eq. 8.

$$x_t^i = \begin{cases} x_t^j & if x_t^j < x_{t-1}^i \\ x_{t-1}^i & otherwise \end{cases} \tag{8}$$

Each substitute player will store different position of players as in Eq. 9.

$$x_t^i \neq x_t^j \quad for\ all\ x_t^j \in S, \qquad i \neq j \tag{9}$$

Movement of players, updating of ball dribblers and knowledge of player, and substitution procedure of players are repeated until the termination criteria are met. Figure 7 demonstrates the flowchart of SGO.

### *3.3.2 Related works*

SGO has been proposed by Purnomo and Wee (2013). Then, Purnomo (2014a) has improved it for discrete and continuous problems by proposing pair cooperation between a player and the ball position. He has implemented the SGO in 8 benchmark problems and obtained competitive results. Fundamental concepts of SGO have been discussed in Purnomo (2014b). A substitution mechanism has been embedded to the classical SGO in order to improve the performance by Purnomo and Wee (2015).

All of the works performed about SGO have been demonstrated in Fig. 8.

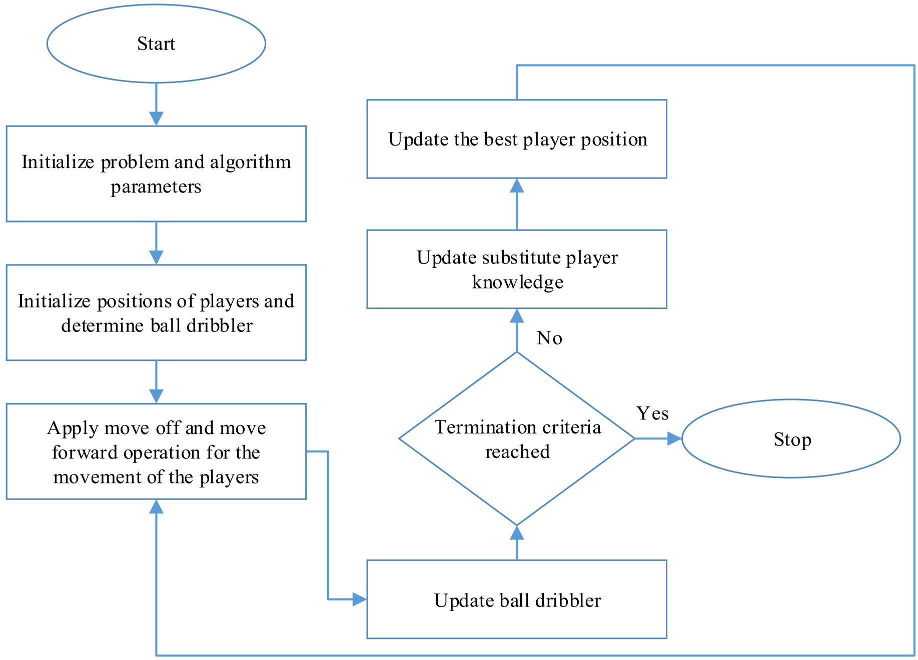


**Fig. 7** Flowchart of SGO

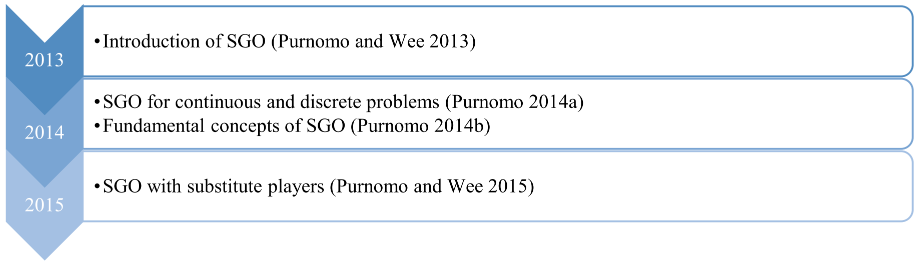


**Fig. 8** SGO in the literature

## 3.4 Soccer league competition

### *3.4.1 Algorithm*

Soccer League Competition Algorithm (SLCA) inspired by professional soccer league is based on the competitions among teams and internal competitions among players of each team (Moosavian and Roodsari 2013). Powers of teams are based on the powers of their players. Players within a team compete with each other to attract the head coach's attention by improving their performance which leads to a growth in the quality and power of their team. In each team, there is a key player, called Star Player (SP), which has the best performance among other players in the team. Furthermore there is a unique most powerful player in each league and is called the Super Star Player (SSP). Comparisons of terms used in the SLCA and the soccer game have been demonstrated in Table 4.

**Table 4** Terms in SLCA

| | Game | SLCA |
|---|---|---|
| 1 | Player | Candidate solution |
| 2 | Power of player | Objective function value |
| 3 | Star Player | Local optimum |
| 4 | Super Star Player | Global optimum |
| 5 | Team | Simultaneous set of candidate solutions |
| 6 | Seasons | Iterations |

After randomly initializing the fixed and substitute players, powers of teams are computed according to Eq. 10.

$$TP\,(i) = \frac{\sum_{j=1}^{NP} PP\,(i, j)}{NP} \tag{10}$$

*NP* is the total number of players in the team. $PP\,(i, j)$ is the power of $j$th player in the $i-$th team.

After every match, players in winner and loser teams of each match adopt different strategies for improving their future performance. These strategies have been performed by four operators used in SLCA.

*Imitation operator*: When a team wins a match, fixed players try to imitate the team's SP, and the SSP of the league. Solution vectors relating to the fixed players in the winner team move toward the best solution of the own team and the best solution vector of the league for this purpose. This operator accelerates the searching capability of SLCA and is performed by Eqs. 11 and 12. Fixed players (FP) in the winner team experiences big movement toward the direction of SP and SSP according to Eq. 11. Newly created solution vector is replaced with the old one if it is better. Otherwise, the solution vector experiences a medium movement toward the resultant vector using Eq. 12. If this solution is better than the older one, it is replaced. The player is kept in its position with no change if a better solution is not obtained.

$$FP\,(i, j) = \mu_1 FP\,(i, j) + \tau_1\,(SSP - FP\,(i, j)) + \tau_2\,(SP\,(i) - FP\,(i, j)) \tag{11}$$

$$FP\,(i, j) = \mu_2 FP\,(i, j) + \tau_1\,(SSP - FP\,(i, j)) + \tau_2\,(SP\,(i) - FP\,(i, j)) \tag{12}$$

$\mu_1 \sim U\,(\theta, \beta)$, $\mu_2 \sim U\,(0, \theta)\,, \tau_1 \sim U\,(0, 2)$, and $\tau_2 \sim U\,(0, 2)$ are random numbers. $1 \leq \beta \leq 2, 0 \leq \theta \leq 1$. $FP\,(i, j)$ stands for the $j$th fixed player of the $i$th team, and $SP\,(i)$ is the Star Player of the $i$th team.

*Provocation operator*: This operator is used to screen the substitutes. Winner's substitutes (S) try to have better performance than that of fixed players in the team. New candidates of substitutes of the winner team move toward the solution direction of the fixed players using Eqs. 13 and 14 where $\chi_1 \sim (0.9, 1)$, $\chi_2 \sim (0.4, 0.6)$ are random numbers and $C\,(i)$ is the mean value of fixed player's solution vectors in the $i$th team. $S\,(i, j)$ is the $j$th substitute of the $i$th team.

$$S\,(i, j) = C\,(i) + \chi_1\,(C\,(i) - S\,(i, j)) \tag{13}$$

$$S\,(i, j) = C\,(i) + \chi_2\,(S\,(i, j) - C\,(i)) \tag{14}$$

Solution vector of the weakest substitute player in the winner team experiences a forward move toward the gravity center of fixed players using Eq. 13. If new solutions are better than the older solutions, the old solutions are replaced. Otherwise, mentioned player will experience a backward movement toward the gravity center using Eq. 14. If this solution is

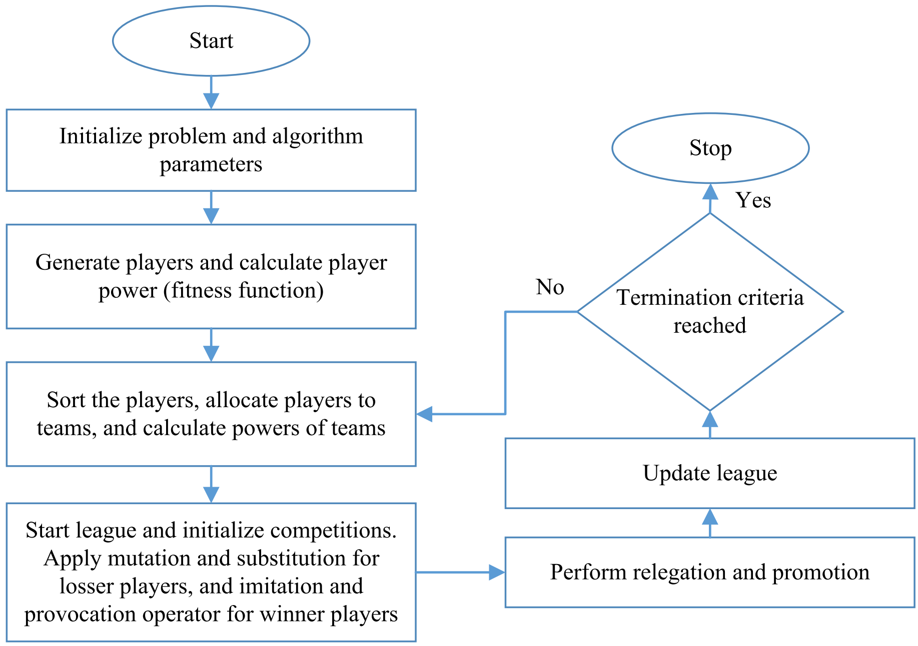


**Fig. 9** Flowchart of SLCA

better than the weakest solution, this vector is replaced with the old one. A new vector is randomly generated and replaced with the old one if if a better solution is not obtained. This operator provides high accurate solutions.

*Mutation operator*: Loser teams seek for ways of improving their performance for reaching better results in future matches. For this reason, fixed players of these teams have to revise their playing style. To perform this operation, the positions of some players are randomly changed for creating diversification in solutions similar to mutation process used in genetic algorithm. This operator helps SLCA to escape from local minimums and plateaus.

*Substitution operator*: Head coach usually considers new combinations of substitutes in order to stop the failures in the future. Similarly, a random-based approach is applied to reflect the head coach impact in SLCA. A pair of new substitutes is tested and if a suitable solution is obtained, this effective pair is entered to the team (Moosavian and Roodsari 2013). This operator helps SLCA to escape from local minimums and plateaus and is applied according to Eqs. 15 and 16.

$$S_{new}\left(i,j\right)=\alpha S\left(i,j\right)+\left(1-a\right)S\left(i,k\right) \tag{15}$$

$$S_{new}\left(i,k\right)=\alpha S\left(i,k\right)+\left(1-a\right)S\left(i,j\right) \tag{16}$$

$a\sim(0,1)$ is a random vector. The number of new tested pairs is proposed to be equal to the number of team substitutes.

Flowchart of SLCA is depicted in Fig. 9.

### *3.4.2 Related works*

SLCA has been proposed in Moosavian and Roodsari (2013). The same authors have utilized the algorithm to obtain optimal solutions for design of water distribution networks

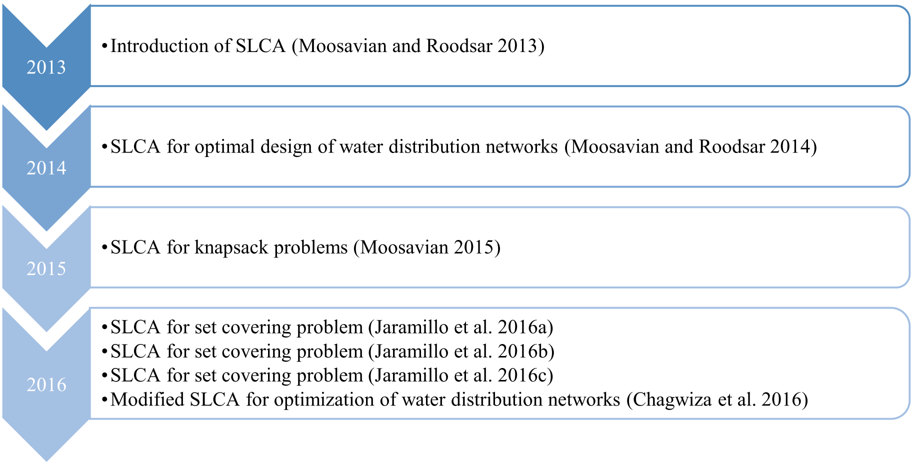


**Fig. 10** SLCA in the literature

(Moosavian and Roodsari 2014). They tested the performance of SLCA on 3 benchmark problems, the two-loop network, the Hanoi network, and the New York Tunnels network. Reliable and rapid solutions obtained from the algorithm have been reported. Moosavian (2015) has enhanced the SLCA to efficiently solve knapsack problems and obtained better results in terms of the search accuracy, reliability, and convergence speed over 10 small and 8 high dimensional knapsack problems. Jaramillo and other researchers have adapted SLCA to obtain effective solutions for set covering problems (Jaramillo et al. 2016a, b, c).

All of the works related to SLCA have been depicted in Fig. 10.

## 3.5 Golden ball algorithm

### *3.5.1 Algorithm*

Golden Ball Algorithm (GBA) introduced by Osaba et al. (2013) is simulated by the different concepts related to soccer. The population of GBA is represented by the soccer teams. Each team has the number of players that each player represents one solution. Let $p_{ij}$ mean "player $j$ of team $i$". The quality of each player $p_{ij}$ is represented by a $q_{ij}$ which is determined by a cost function f($p_{ij}$). The captain ($p_{icap}$) who is represented by the best solution (best $q_{ij}$) is the most important player in the team. This can be stated more formally as

$$p_{icap} = p_{ik} \in t_i \Leftrightarrow \forall j \in \{1, \ldots, PT\} : q_{ik} \geq q_{ij}, PT = \text{number of players per team}$$

To calculate the strength $TQ_i$ of each team, the GBA takes into account the quality of all the $p_{ij}$ that comprise that team. $TQ_i$ is expressed by Eq. 17.

$$TQ_i = \sum\nolimits_{j=1}^{PT} \frac{q_{ij}}{PT} \tag{17}$$

GBA divides the initial solutions into groups which represents a team. Each team works independently and competes with other teams to get the best solution. Once the teams have been formed, the season ($S_i$) begins. A $S_i$ has as many matches as necessary to complete a conventional league, where all teams face each other twice. For this reason, every season is divided into two parts of equal duration. In these parts, all teams have to face each other team once. Finally, a season has as many training phases as matchdays (Osaba et al. 2014a). Each team plays games against the other teams to create a league competition. The good players affect their win rate because the strength of the team depends on the quality of the players. Before the match, all players from each team have opportunity to improve themselves by using training procedures which are successor functions that work on a particular neighborhood structure in the solution space. For each training, this functions are applied a certain number of times (until its own termination criterion is reached) to improve a $p_{ij}$. Generated $p'_{ij}$ is accepted only if $q'_{ij} > q_{ij}$. When a player $p_{ij}$ exceeds the quality of the captain of its team the captain is changed with this training procedure.

Each match consists of *PT* chances. Each chance materializes in goal through a tournament between a $p_{ij}$ for each team, which are faced by their team position. The player with higher $q_{ij}$ wins the chance and secures a goal for his team. The team that scores more goals is the winner of the match. The points scored by each team are used to perform a classification, sorted by the number of points scored.

Any player can switch from his team to another by using a transfer procedure (Osaba et al. 2013, 2014a, b). The best team gets the best player of the worst team. On the other hand, the second best team receives the second best player of the penultimate team. An example of this transfer process for a league consists of 4 teams, taking into account that players are ranked in order of their $q_{ij}$ and the $t_i$ depending on their position in the league is shown below:

$$\text{Team } t_1 : \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}\}$$
$$\text{Team } t_2 : \{p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}, p_{27}, p_{28}\}$$
$$\text{Team } t_3 : \{p_{31}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}, p_{38}\}$$
$$\text{Team } t_4 : \{p_{41}, p_{42}, p_{43}, p_{44}, p_{45}, p_{46}, p_{47}, p_{48}\}$$

After the period of signings, these teams could be composed as follows:

$$\text{Team } t_1 : \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, \boldsymbol{p}_{41}\}$$
$$\text{Team } t_2 : \{p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}, \boldsymbol{p}_{32}, p_{28}\}$$
$$\text{Team } t_3 : \{p_{31}, \boldsymbol{p}_{27}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}, p_{38}\}$$
$$\text{Team } t_4 : \{\boldsymbol{p}_{18}, p_{42}, p_{43}, p_{44}, p_{45}, p_{46}, p_{47}, p_{48}\}$$

The other type of transfer is called special exchanges in which the player can decide to change team, regardless of whether the target is a worse or better team. If a $p_{ij}$ takes a certain number of trainings without improvements in its $q_{ij}$ even receiving custom trainings, it changes from its $t_i$ to another random $t_k$, with the aim of obtaining new different trainings. In addition, it is no matter if $TQ_k < TQ_i$. To keep the *PT* per team, there is an exchange with a random $p_{ij}$ of $t_k$ (Osaba et al. 2013, 2014a, b). An example of this transfer process might be as follows, assuming the following $t_1$ and $t_2$:

$$\text{Team } t_1 : \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}\}$$
$$\text{Team } t_2 : \{p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}, p_{27}, p_{28}\}$$

Supposing that $p_{14}$ is the player that has not experienced any improvement in its $q_{ij}$, an exchange is produced with a random player of another random team. In this case this team is $t_2$. Assuming that the selected player for the exchange is $p_{26}$, the teams would be as follows:

$$\text{Team } t_1 : \{p_{11}, p_{12}, p_{13}, \boldsymbol{p}_{26}, p_{15}, p_{16}, p_{17}, p_{18}\}$$
$$\text{Team } t_2 : \{p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, \boldsymbol{p}_{14}, p_{27}, p_{28}\}$$

**Table 5** Terms in GBA

| | Game | GBA |
|---|---|---|
| 1 | Player | Candidate solution |
| 2 | Quality of player | Cost function |
| 3 | Team | Subpopulation |
| 4 | Set of teams | Entire population |
| 5 | Captain | Best player |
| 6 | Components of the player | Variable values of the candidate solution |

Coaches are also replaced when their teams are not getting the expected results, or when they are getting bad results continuously in process called as cessation of coaches in GBA.

GBA execution is terminated according to three conditions. It can be stopped when the sum of the powers $TQ_i$ of all the teams does not improve in comparing to the previous season as shown in Eq. 18. It can be terminated if the sum of the quality $q'_{icap}$ of all the captains has no improvement over the previous season as shown in Eq. 19. It can also be terminated when there is no improvement in the best solution found by the whole system ($BestSolution'$) in relation to the previous season as in Eq. 20.

$$\sum\nolimits_{i=1}^{TN} TQ'_i \leq \sum\nolimits_{i=1}^{TN} TQ_i, \qquad TN = \text{ total number of teams} \tag{18}$$

$$\sum\nolimits_{i=1}^{TN} q'_{icap} \leq \sum\nolimits_{i=1}^{TN} q_{icap} \tag{19}$$

$$BestSolution' \leq BestSolution \tag{20}$$

Comparisons of terms in real game and GBA have been listed in Table 5.

Flowchart of GBA is depicted in Fig. 11.

### *3.5.2 Related works*

GBA has been introduced by Osaba et al. (2013) in a conference for combinatorial optimization problems and then it has been published as a journal article in Osaba et al. (2014a). They have tested the performance of GBA on traveling salesman problem and capacitated vehicle routing problem and reported better results than those of classical genetic algorithm, parameters adjusted genetic algorithm, distributed genetic algorithm, and parameters adjusted distributed genetic algorithm. Ruttanateerawichien et al. (2014) have proposed improved GBA for capacitated vehicle routing problem. They have tested their improved GBA within 88 problems and reported the efficiency of the method against classical GBA. Osaba et al. (2014b) have applied GBA to 62 new problems, namely the asymmetric traveling salesman problem, the vehicle routing problem with backhauls, $n$-queen problem, and one-dimensional bin packing problem. It has been concluded that GBA is a promising method for the selected combinatorial optimization problems. Sayoti and Riffi (2015) have proposed random keys representation to encode solutions in GBA and applied random-keys GBA for solving the traveling salesman problem. Random-keys GBA has been tested on 36 TSPLIB problems. GBA has been tested with evolutionary simulated annealing and tabu search algorithm on traveling salesman problems in Osaba et al. (2016). Sayoti and Ri (2016) have designed GBA in flow shop scheduling goal of which is to find a schedule that minimizes the make span. 22 problems from ORLibrary have been used for testing and it has been reported that GBA can

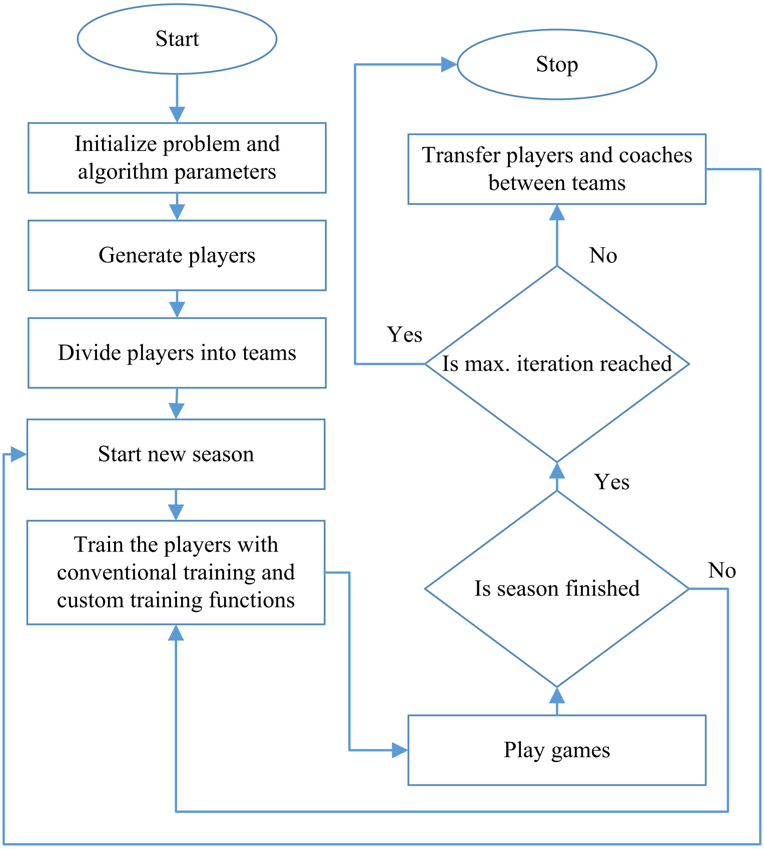


**Fig. 11** Flowchart of GBA

be practical for small OR-Library flow shop scheduling problems. Ruttanateerawichien et al. (2016) have proposed a new GBA using new solution representation for solution of capacitated vehicle routing problem. Efficient and effective solutions obtained from improved GBA have been reported.

All of the works related to GBA have been depicted in Fig. 12.

## 3.6 World cup optimization

### *3.6.1 Algorithm*

World Cup Optimization (WCO) is one of the most recent sports inspired search and optimization method and is based on the FIFA tournaments and competitions among the teams for reaching the World Cup (Razmjooy et al. 2016; Razmjooy and Ramezani 2016).

The entire team groups usually are classified into some seeds based on their strengths. Challenge starts after seeding the teams. In this challenge, teams compete with each other to win the game and collect more points for the next cups. After the preliminary competitions, two top teams of each group rise into the upper level and the other teams will be eliminated. The third place of each competition group is also competing with each other to get a chance

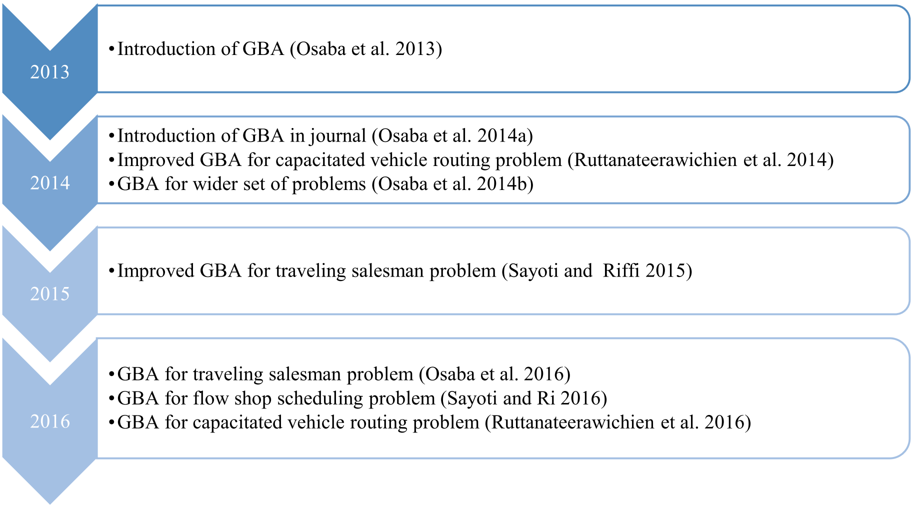


**Fig. 12** GBA in the literature

to improve them into the next stage. This competition resumes in multi-stages. The final stage holds between two powerful teams, and the winner wins the cup.

WCO starts by generating some candidate continents matrix of size $N_{pop} \times N_{var}$ where $N_{pop}$ defines the number of teams and $N_{var}$ is the number of variables in the problem. After initializing and seeding the teams, each score of each continent is evaluated. For this purpose, first all continents are acquired. Mean value and standard deviation of each continent is computed as shown in Eqs. 21 and 22 respectively. Here, $\bar{X}$ is the mean value and $\sigma$ is the standard deviation of the continent $X$; $n$ is the number of members in $X$. Then continents are scored according to Eq. 23 where *Rank* is a sorting operator and $\beta$ is a coefficient term for the increase or decrease in the standard deviation effect in the interval [0, 1].

$$\bar{X} = \frac{1}{n} \sum\nolimits_{i=1}^{n} X_i \tag{21}$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum\nolimits_{i=1}^{n} \left(X_i - \bar{X}\right)^2} \tag{22}$$

$$Rank = \frac{\beta \times \sigma + \bar{X}}{2} \tag{23}$$

Then all continents are sorted in a vector based on their rank. Assuming five continents as in the original FIFA and $n$ teams as default, this procedure is applied as follows:

$$\begin{aligned} X_1 &= [X_{11}, \ldots, X_{1n}]^T \\ X_2 &= [X_{21}, \ldots, X_{2n}]^T \\ &\ldots \\ X_5 &= [X_{51}, \ldots, X_{5n}]^T \\ X_{Total} &= [X_{11}, \ldots, X_{1n}, X_{21}, \ldots, X_{2n}, \ldots, X_{51}, \ldots, X_{5n}]^T \end{aligned}$$

Here $n$ is the number of teams for each continent, and $T$ is transposition of vector. After applying this procedure, two minimum values of each continent are separated and are placed into a vector variable ($X_{Rank}$) for the next championship and the minimum value of $X_{Total}$

**Table 6** Terms in WCO

| | Game | WCO |
|---|---|---|
| 1 | Team | Candidate solution |
| 2 | Continent | Candidate solution clusters |
| 3 | Score | Fitness |
| 4 | Champion of cup | Optimum solution |
| 5 | Competition | Iteration |

is selected as the current cup's champion as shown below:

$$X_{Rank} = [X_{11}, X_{12}, X_{21}, X_{22}, \ldots, X_{51}, X_{52}]^T$$
$$X_{Champion} = \min(X_{Tatol}) = min\left([X_{11}, \ldots, X_{1n}, X_{21}, \ldots, X_{2n}, \ldots, X_{51}, \ldots, X_{5n}]^T\right)$$

After the champion team of the current cup is determined, new population (continents and their teams) is defined based on the prior cup and the ranking of the teams (Razmjooy et al. 2016; Razmjooy and Ramezani 2016). Two part vector is utilized.

$$Pop = X_{Total} = [X_{Best}, X_{Rand}]$$

Pop is the total new population with the size of $(N \times M)$ where $N$ is the variable dimension and $M$ is the number of continents. $X_{Rand}$ is a random value between the problem limitation intervals, and $L < X_{Best} < U$ is a vector.

$$U = \frac{ac(Ub + Lb)}{2} \tag{24}$$
$$U = \frac{ac(Ub - Lb)}{2} \tag{25}$$

where $ac$ is accuracy coefficient between $Ub$ and $Lb$ as high and low bounds for the interested problem.

$X_{Rand}$ is applied as the exploration term, while $X_{Best}$ is applied as the exploitation term of the population. The size of $X_{Rand}$ and $X_{Best}$ can be separated and changed due to the problem statements with a value of Cross Point (CP) as shown in Eqs. 26 and 27:

$$X_{Rand} = Pop(1 : CP, M) \tag{26}$$
$$X_{Best} = Pop((CP + 1) : N, M) \tag{27}$$

After generating the new population, it will be divided into $m$ teams of $n$ continents.

Terms used in WCO have been listed in Table 6 and a flowchart of the WCO is shown in Fig. 13.

### *3.6.2 Related works*

WCO is one of the most recent method in computational intelligence field. There is only two works except from the introduction of WCO in Razmjooy et al. (2016). Razmjooy and Ramezani (2016) have used WCO for the complex model reduction problems in order to provide a low order system model with less hardware requirements. The works about WCO have been listed in Fig. 14. Shahrezaee (2017) has adapted WCO in order to classify the main components of an image (pixels) into different groups. He has reported that WCO based method achieved good performance when compared to Otsu, genetic algorithm based and APSO based image segmentation algorithms.

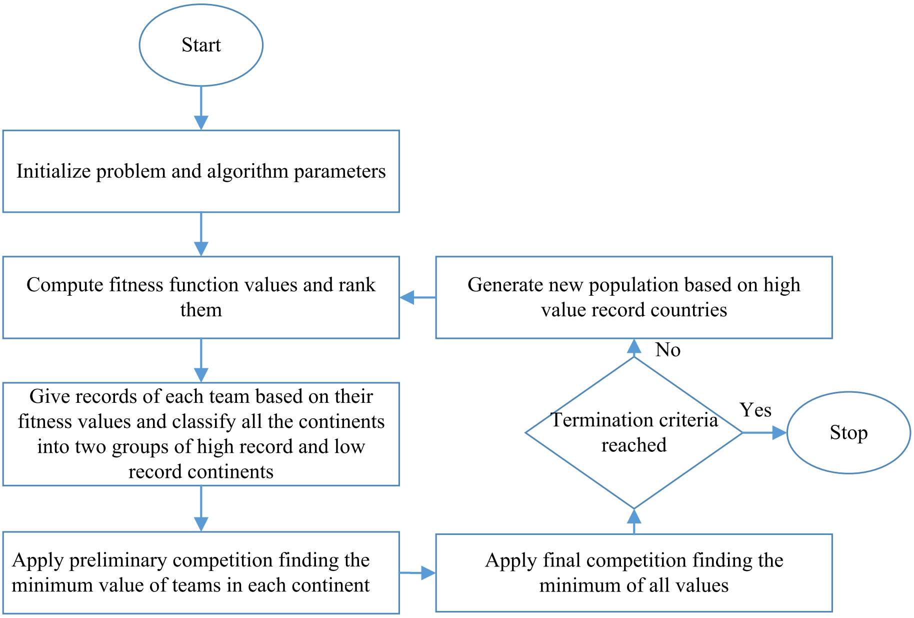


**Fig. 13** Flowchart of WCO

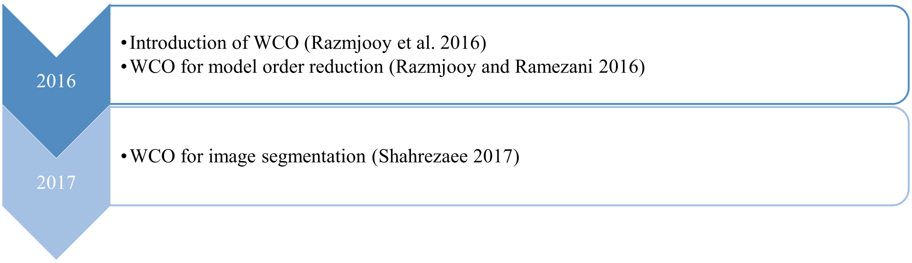


**Fig. 14** WCO in the literature

## 3.7 Football optimization algorithm

### *3.7.1 Algorithm*

Football Optimization Algorithm (FOA) is another football inspired population based search and optimization method proposed by Hatamzadeh and Khayyambashi Hatamzadeh and Khayyambashi (2012a, b).

FOA starts the search with an initial population called team. A team composed of good passers and mobile players. All the players are divided into two types: main players and substitute players.

$$main\ players = \begin{pmatrix} player_1 \\ \vdots \\ player_m \end{pmatrix}_{m \times k}$$

$$substitute\ players = \begin{pmatrix} player_{m+1} \\ \vdots \\ player_n \end{pmatrix}_{s \times k}$$

$$m = round\,(n \times \alpha)\,, s = n - m, n = Maximum\ number\ of\ players,$$
$$\alpha = Divide\ coefficient\ of\ players$$

Equation 28 determines the certain selection methods that rate the fitness of each candidate solution, adds random value according to uniform distribution ($U$) to it, and preferentially selects the best rank solution.

$$OwnerBall = BestIndex\,\{Rank_1, \ldots, Rank_m\} \tag{28}$$

where

$$Rank_i = Fitness\,(player_i) + U\,(-d, d) \tag{29}$$

In each iteration, the rank of every player in the population is computed, the best-ranked player is selected from the current main players, and parameters exchanged between passer and it. Once a player has passed the ball, other players move into a position where they can receive the ball and give more options to the player in possession according to Eq. 30 where the players move towards the best player by $x$ units.

$$x \in \left[0, U\,(v \times d) + \gamma\right] \tag{30}$$

$v$ is velocity coefficient of players. It is a number greater than zero and causes the players to get closer to the goal from any side. $d$ is the distance between the best player and other players and $\gamma$ is a parameter that adjusts the deviation from the original direction. Spectators' movement is modeled by random change in player's parameters as in Eqs. 31 and 32 where $m$ is a number of main players and $k$ is number of parameters of each player. $\epsilon$ is spectators' effect on players and $\sigma$ is spectators' effect on parameters in the interval [0, 1].

$$EffectPlayers = k \times \epsilon \tag{31}$$

$$EffectParameters = m \times \sigma \tag{32}$$

A number of substitution may be performed according to comparison between the weakest main player and the best substitute player. If the substitute player is stronger than the main player, a switch takes place. During this procedure FOA can use of λ (number of replacement) to adjust parameters.

Teamwork among main players is the main part of the proposed algorithm and expectantly causes the ball to converge to the goal. Teamwork is achieved when individuals make personal sacrifices to work together for the success of the group (Hatamzadeh and Khayyambashi 2012).

Terms used in FOA have been listed in Table 7. Flowchart of the FOA is shown in Fig. 15.

### *3.7.2 Related works*

There are only two works about FOA in the literature. In (Hatamzadeh and Khayyambashi 2012a), FOA has been introduced and performance comparisons within 4 benchmark problems have been presented. FOA has been proposed for neural network learning in Hatamzadeh and Khayyambashi (2012b). FOA has been tested within 5 datasets. Training and test errors obtained from FOA have been shown better than those of three state-of-the arts algorithms. The works about FOA have been listed in Fig. 16.

**Table 7** Terms in FOA

| | Game | FOA |
|---|---|---|
| 1 | Team | Population |
| 2 | Player | Candidate solution |
| 3 | Power of a player | Fitness function value |
| 4 | Football game area | Search space |
| 5 | Moving by spectators' effect | Random change in player's parameters |
| 6 | Passing the ball to the best player | Parameter exchange |
| 7 | Attacking players into free space | Moving all the players toward the best player |
| 8 | Substitute | Switch between substitute player and main player |

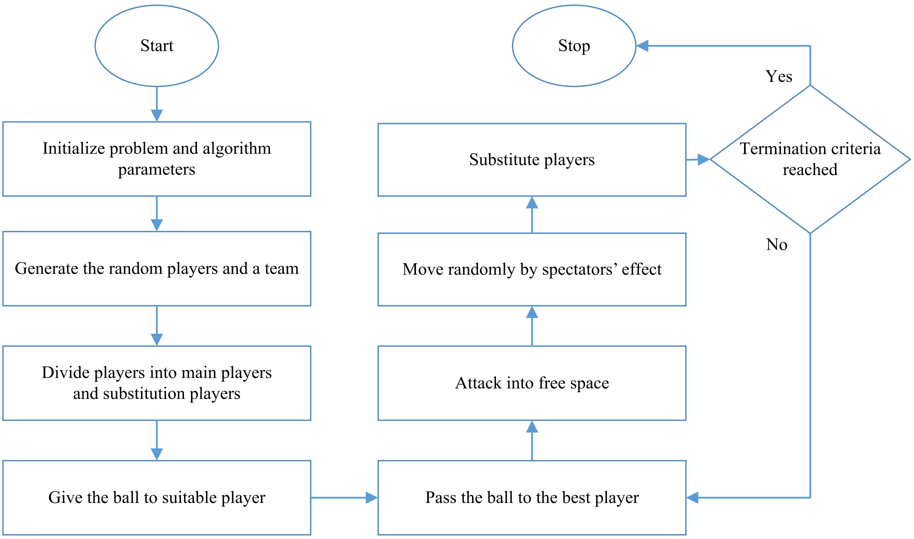


**Fig. 15** Flowchart of FOA

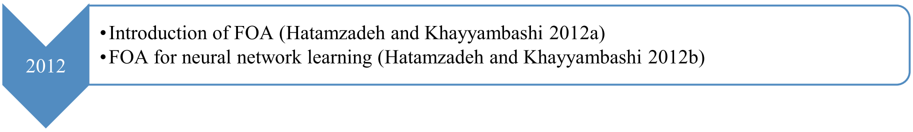


**Fig. 16** FOA in the literature

## 3.8 Football game algorithm

### *3.8.1 Algorithm*

Football Game Algorithm (FGA) is one of the most recent sports based search and optimization methods. It simulates football players' behavior during a game for finding best positions to score a goal under supervision of the team coach (Fadakar and Ebrahimi 2016). Terms used in FGA have been listed in Table 8. After initial formation, every player moves around his last

**Table 8** Terms in FGA

| | Game | FGA |
|---|---|---|
| 1 | Football players | Population members |
| 2 | Football pitch | Search space |
| 3 | Initial formation of the players | Initial population |
| 4 | Position | Fitness value |
| 5 | Move around the last position | Random walk |
| 6 | Scoring a goal | Termination criteria |
| 7 | Attacking | Hyper radius penalty |
| 8 | Substitution | Fitness penalty |

position biased toward the ball. Ball is passed between the players. Players in better position have more chance to take the ball. The new positions of players depend on simple random walk and a movement toward the ball without coaching effects as computed in Eq. 33.

$$X_i^t = X_i^{t-1} + \alpha_i \varepsilon + \beta \left( X_{Ball}^t - X_i^{t-1} \right) \tag{33}$$

$\varepsilon \in [-1, 1]$ and $\beta \in [0, 1]$ are random numbers drawn from a uniform distribution and $\alpha > 0$ is the step size according to the scales of the considered problem. $\alpha$ can be randomized by decreasing gradually while iteration continues to inmprove the convergence. $X_{Ball}^t$ is the position of the player who has the ball at iteration $t$. The ball is passed randomly between the players. The players with high fitness value have more chance to receive the ball.

Except from general movements of player, there are two general positioning strategies that can be applied by the coach to increase the chance of finding global optimum using local search. In attacking strategy, team coach memorizes the best positions during the game and uses them to guide players and pushing them forward. Coach encourages players to go forward and increase the pressure on the opposing team. Members with higher distance value in comparison with Hyper Radius Limitation Value (*HRLV*) will be pushed toward the nearest best positions. *HRLV* is decreased gradually as the iterations proceed. Hyper Radius Penalty method is applied as shown in Eq. 34.

$$HRLV^t = HRLV_{min} + \gamma \left( HRLV^{t-1} - HRLV_{min} \right) \tag{34}$$

$\gamma \in (0, 1)$ is the reduction constant of *HRLV*.

In substitution strategy, coach can change a defender in low quality position with a fresh striker in the best position to increase the chance of scoring. Weaker players will be replaced with other players around the nearest best position according to the coach memory. Every member of the population which has grater fitness value than Fitness Limitation Value (*FLV*) will be replaced with another one around the nearest best solution. *FLV* will be decreased accordingly as the iterations proceed. Fitness Penalty method is applied as in Eq. 35 where $\lambda$ has the same role as in Eq. 34.

$$FLV^t = FLV_{min} + \lambda \left( FLV^{t-1} - FLV_{min} \right) \tag{35}$$

After applying the strategies, the new position of players who are located beyond the limitation values will be achieved by using random walk from the nearest best solution to their old position according to Eq. 36

$$X_{new} = X_{nearest-best} + \alpha_i \varepsilon \tag{36}$$

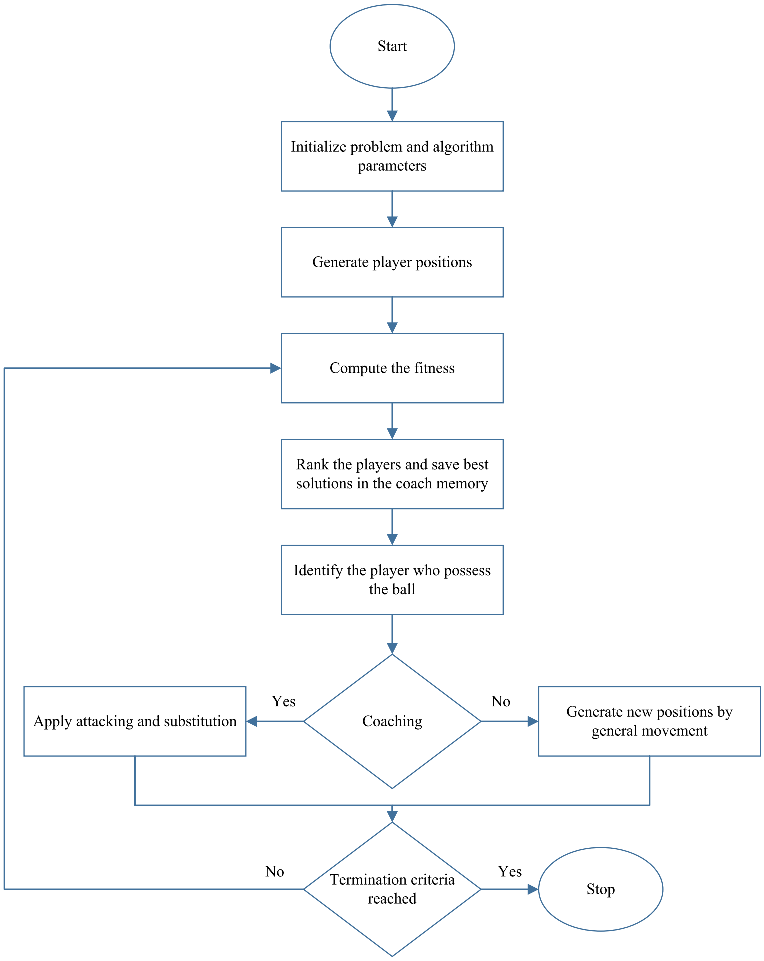


**Fig. 17** Flowchart of FGA

These strategies will continue until the end of the game or until the team scores a goal. FGA is very new and in the literature, there is not any work about FGA except from its introduction.

According to these explanations, flowchart of the FGA is shown in Fig. 17.

### *3.8.2 Related works*

FGA is very new method. In the literature, there is only one paper about FGA and it introduces the method (Fadakar and Ebrahimi 2016).

## 3.9 Most valuable player algorithm

### *3.9.1 Algorithm*

Most Valuable Player Algorithm (MVPA) is the most recent sport based optimization algorithm (Bouchekara 2017). A population of players compete collectively in teams in order to win the leagues' championship, and they compete individually in order to win the most valuable player trophy. The number of players' skills corresponds to the dimension of the problem and team is composed of a group of players. Once the players is created they are randomly distributed to form '*TeamsSize*' teams. As an example to form teams, the first '$nT1$' teams have '$nP1$' players, while the remaining '$nT2$' teams have '$nP2$' players where $nP1$, $nP2$, $nT1$, and $nT2$ are computed according to Eqs. 37–40 where *PlayersSize* represents the number of players in the league.

$$nP1 = \left\lceil \frac{PlayersSize}{TeamsSize} \right\rceil \tag{37}$$

$$nP2 = nP1 - 1 \tag{38}$$

$$nT1 = PlayersSize - nP2 \times TeamsSize \tag{39}$$

$$nT2 = TeamsSize - nT1 \tag{40}$$

Each player aims to be his team's franchise player and the league's most valuable player. That is why, in individual competition step of MVPA, the skills of the players of the selected $TEAM_i$ are updated according to Eq. 41.

$$TEAM_i = TEAM_i + rand \times (FranchisePlayer_i - TEAM_i) + 2 \times rand \times (MVP - TEAM_i) \tag{41}$$

where *rand* is a uniformly distributed random number in the interval (0, 1). Bouchekara has also indicated that the constant 2 can be changed to a different real number for other specific optimization problems, or $2 \times rand$ term can also be changed by another random number with different distribution.

In team competition step of MVPA, for the selected $TEAM_i$, another team $TEAM_j$ is randomly selected where $(i \neq j)$ and they play against each other. The probability that $TEAM_i$ beats $TEAM_j$ is calculated using Eq. 42 according to a normalization shown in Eq. 43.

$$Pr\left\{TEAM_i \text{ beats } TEAM_j\right\} = 1 - \frac{(fitnessN\,(TEAM_i))^k}{(fitnessN\,(TEAM_i))^k + \left(fitnessN\left(TEAM_j\right)\right)^k} \tag{42}$$

$$fitnessN\,(TEAM_i) = fitness\,(TEAM_i) - min\,(fitness\,(All\ Teams)) \tag{43}$$

If two teams have the same fitness they will have the same winning probability. Hence, a random number is generated; if this number is higher than 0.5 the first team wins, otherwise the second team wins (Bouchekara 2017).

In the team competition step, if $TEAM_i$ is selected and it plays against $TEAM_j$, if $TEAM_i$ wins the skills of players of $TEAM_i$ are updated using Eq. 44. Otherwise, the skills of players of $TEAM_i$ are updated using Eq. 45. Boundary constraints are also checked in the competition step.

$$TEAM_i = TEAM_i + rand \times \left(TEAM_i - FranchisePlayer_j\right) \tag{44}$$

$$TEAM_i = TEAM_i + rand \times \left(FranchisePlayer_j - TEAM_i\right) \tag{45}$$

After competitions, a comparison between the population before and after the competition phase is performed and a new solution is accepted if it gives a better objective function value than the initial one greedily. The worst players are replaced by the best ones in elitism step of

**Table 9** Terms in MVPA

| | Game | MVPA |
|---|---|---|
| 1 | Player | Candidate solution |
| 2 | Group of players | Population |
| 3 | Skills of players | Design variables (dimension) |
| 4 | Franchise player in a team | Candidate solution with the best fitness in a team |
| 5 | League's MVP | The the best solution so far |
| 6 | Fixture | Iteration |
| 7 | Efficiency (the rating or the strength) | Fitness |

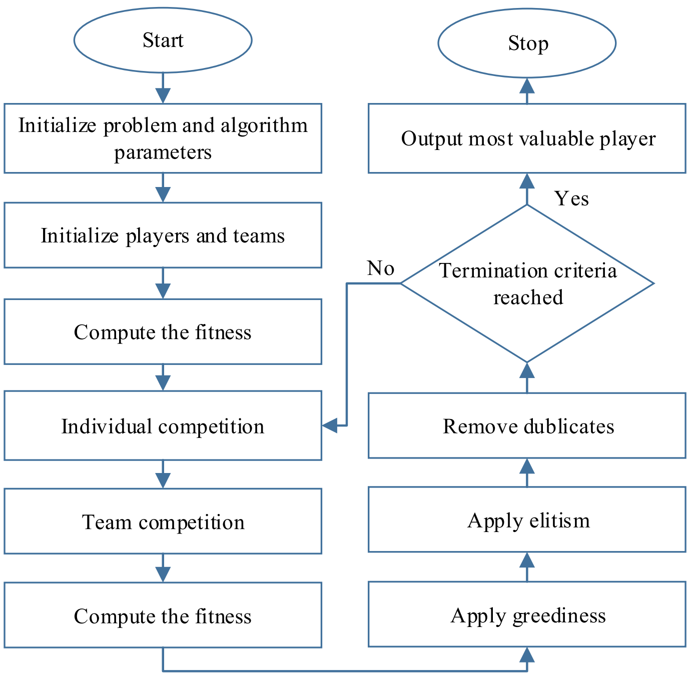


**Fig. 18** Flowchart of MVPA

MVPA. If two successive players in the population are exactly the same, the second player is replaced by another one.

Terms used in MVPA have been listed in Table 9 and flowchart of the MVPA is shown in Fig. 18.

## 4 Benchmark functions

Well-defined benchmark functions based on mathematical functions can be used to measure and test the performance of optimization methods. The nature, complexity, and other characteristics of these benchmark functions can easily be derived from their definitions and have the nature and complexity of most engineering problems. The difficulty levels of most benchmark functions are adjustable by setting their parameters. The selected benchmark functions and their properties have been demonstrated in Table 10. In this table, Min represents the minimum value and Dim represents the selected number of dimensions. Their detailed characteristics can be found in Surjanovic and Bingham (2013), Jamil and Yang (2013).

## 5 Experimental results

In this work, benchmark functions with different types, complexities, and dimensionalities have been selected for evaluating the performances of sports based algorithms and other known search and optimization algorithms in the literature. These benchmark functions are briefly decribed in Table 10. All of the sports based algorithms except from SLO and GBA have been included in performance comparisons. In the unique paper that introduces SLO, formal mathematical formulations have not been provided and it is hard to understand and apply its operators for a search problem. GBA has been proposed for discrete optimization and all of the works on GBA are related to discrete optimization. It cannot be meaningful to compare it with the algorithms proposed for numerical global optimization problems which consist of continuous decision variables. That is why, these two sports based algortihms have been excluded from performance comparisons within unconstrained global optimization benchmark problems. Furthermore, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE) have been included for performance comparisons as most known and popular optimization algorithms. The parameter settings of the algorithms are listed in Table 11.

Five experiments have been performed. In experiment 1 (E1), 30 runs are conducted with fixed 90.000 function evaluations to compare the comprehensive performance of algorithms within the global unconstrained benchmark optimization problems. The best, worst, mean, and standard deviation (SD) values for every problem are presented in Table 12.

As it can be seen from Table 12, sports based MVPA is the best algorithm in unimodal Sphere function according to the obtained best and mean values. Sports based FGA is the second best algorithm within the selected parameters. DE has given the worst performance in this function although it is relatively old and effective algorithm. Lower standard deviation values mean that obtained minimum values in this experiments are very close to each other. In other words, all results approach the mean value and the search or optimization algorithm can obtain a proper result in all executions without depending on the initial conditions. WCO, FGA, MVPA, and PSO seems more stable according to the obtained standard deviation values.

In Rosenbrock function, sports inspired WCO has given the best performances according to the obtained best and mean values. MVPA is the second best algorithm in this benchmark function. Parameter adjustment should be performed in this type of algorithm for GA due to the worst performance.

In multimodal Rastrigin function, the real minimum value of zero (0) has been obtained from all executions of LCA, MVPA, GA, and DE. SGO seems the worst algorithm for this problem.

In multimodal Schwefel function; LCA, FOA, SGO, WCO, FGA, MVPA, and DE have given the same performance finding the same values in all executions while SLCA, GA, and PSO seems the worst algorithms.

**Table 10** Benchmark functions

| Function | Definition | Min | Dim | Interval |
|---|---|---|---|---|
| Sphere | $f_1(x) = \sum_{i=1}^{n} x_i^2$ | 0 | 10 | $-10 \leq x_i \leq 10$ |
| Rosenbrock | $f_2(x) = \sum_{i=1}^{n-1} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right)$ | 0 | 10 | $-30 \leq x_i \leq 30$ |
| Rastrigin | $f_3(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ | 0 | 10 | $-5 \leq x_i \leq 5$ |
| Schwefel | $f_4(x) = n \times 418.9829 + \sum_{i=1}^{n} -x_i \sin\left(\sqrt{x_i}\right)$ | 0 | 10 | $-500 \leq x_i \leq 500$ |
| Ackley | $f_5(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + exp(1)$ | 0 | 50 | $-30 \leq x_i \leq 30$ |
| Griewank | $f_6(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$ | 0 | 10 | $-600 \leq x_i \leq 600$ |
| GoldsteinPrice | $f_7(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$ | 3 | 2 | $-2 \leq x_i \leq 2$ |
| Michalewicz | $f_8(x) = -\sum_{i=1}^{2} sin(x_i)\left(sin\left(\frac{ix_i^2}{\pi}\right)\right)^{2m}, m = 10$ | −1.8013 | 2 | $0 \leq x_i \leq \pi$ |
| Schwefel22 | $f_9(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ | 0 | 30 | $-10 \leq x_i \leq 10$ |

**Table 11** Parameters of the algorithms

| Algorithm | Parameters |
|---|---|
| LCA | $c_1 = 0.50$ |
| | $c_2 = 0.50$ |
| | $pc = 0.01$ |
| SGO | Cooperation rate $w_b = 0.618$ |
| | $w_p = 1 - w_b = 0.382$ |
| | Randomly moving probalibity = 0.10 |
| SLCA | Mutation probability = 0.10 |
| | Mutation rate = 0.20 |
| WCO | Play-off coefficient = 0.05 |
| | Coefficient $\beta = 0.50$ |
| FGA | Randomness reduction constant $\theta = 0.50$ |
| | Reduction constant $\gamma = 0.95$ |
| | $\lambda = 0.85$ |
| FOA | $\alpha = 0.5$ |
| | $\beta = 1$ |
| | $\upsilon = 0.10$ |
| | $\sigma = 0.10$ |
| MVPA | Exponent $k = 1$ |
| GA | Crossover rate = 0.80 |
| | Mutation rate = 0.10 |
| | Selection: Roulette wheel |
| PSO | $c_1 = 1.50$ |
| | $c_2 = 2$ |
| | Inertia weight $w$ linearly decreases from 0.90 to 0.40 along with iteration |
| DE | Crossover rate = 0.20 |
| | Upper bound of scaling factor = 0.8 |
| | Lower bound of scaling factor = 0.2 |

MVPA has obtained the best results for Ackley function with 20 dimensions among other algorithms. SLCA is the second best algorithm for this function according to the determined values. PSO has given the worst performance. Parameter adjustment for PSO can enhance its performance.

FOA, SLCA, WCO, GA, MVPA, and DE have found the best value for Griewank function as zero in all of their executions giving a value of zero for standard deviation. All of the algorithms have also found the optimum values of GoldsteinPrice and Michalewicz functions. Best results are obtained from WCO within Schwefel22 function.

In other experiments, an algorithm is considered as successful if the condition in Eq. 46 is fulfilled:

$$\left| f_{known\ minimum} - f_{algorithm\ minimum} \right| \leq 10^{-6} \tag{46}$$

Maximum number of functions evaluations (FEs) is set to 2000, 5000, 10000, and 25000 for experiment 2 (E2), experiment 3 (E3), experiment 4 (E4) and experiment 5 (E5), respec-

**Table 12** Results obtained from E1

| Algorithm | LCA | FOA | SGO | SLCA | WCO | FGA | MVPA | GA | PSO | DE |
|---|---|---|---|---|---|---|---|---|---|---|
| Function | $f_1$ : Sphere | | | | | | | | | |
| Best | 1.02E−14 | 2.21E−24 | 1.15E−12 | 2.65E−35 | 3.16E−28 | 2.13E−28 | **1.10E−28** | 3.45E−83 | 3.13E−27 | 1.41E−22 |
| Worst | 1.20E−13 | 4.35E−25 | 1.16E−10 | 5.65E−32 | 4.34E−26 | 1.44E−28 | 1.12E−28 | 1.65E−80 | 4.09E−26 | 3.45E−22 |
| Mean | 1.55E−14 | 2.05E−25 | 2.15E−E−11 | 1.15E−33 | 1.21E−27 | 1.56E−28 | **1.43E−28** | 2.45E−83 | 2.11E−27 | 2.86E−22 |
| SD | 6.21E−14 | 0 | 5.25E−11 | 2.28E−32 | 0 | 0 | 0 | 9.50E−81 | 0 | 1.60E−22 |
| Function | $f_2$ : Rosenbrock | | | | | | | | | |
| Best | 0.0090 | 1.1200 | 4.2402 | 3.8402 | **0.0001** | 0.7201 | 0.0062 | 0.7101 | 1.0101 | 0.4752 |
| Worst | 0.5147 | 1.1250 | 4.4802 | 4.8321 | 0.0962 | 1.0022 | 0.8211 | 7.1287 | 1.2742 | 3.5871 |
| Mean | 0.3112 | 1.1223 | 4.3012 | 4.2120 | **0.0056** | 0.8813 | 0.0269 | 4.1258 | 1.1221 | 1.9601 |
| SD | 0.1947 | 0.002 | 0.125 | 0.4502 | 0.0412 | 0.1413 | 0.4125 | 3.1511 | 0.1324 | 1.4571 |
| Function | $f_3$ : Rastrigin | | | | | | | | | |
| Best | **0** | 0.8891 | 7.0231 | 3.9798 | 7.12E−07 | 5.1254 | **0** | **0** | 3.0012 | **0** |
| Worst | 0 | 1.2562 | 8.0124 | 12.1021 | 9.85E−05 | 8.2124 | 0 | 0 | 11.2352 | 0 |
| Mean | **0** | 1.0256 | 7.6523 | 6.5323 | 4.26E−06 | 6.6532 | **0** | **0** | 4.2152 | **0** |
| SD | 0 | 0.1052 | 0.5148 | 2.7985 | 4.23E−05 | 1.2056 | 0 | 0 | 3.2565 | 0 |

**Table 12** continued

| Algorithm | LCA | FOA | SGO | SLCA | WCO | FGA | MVPA | GA | PSO | DE |
|---|---|---|---|---|---|---|---|---|---|---|
| Function | $f_4$ : Schwefel | | | | | | | | | |
| Best | **1.27E−04** | **1.27E−04** | **1.27E−04** | 888.2321 | **1.27E−04** | **1.27E−04** | **1.27E−04** | 118.4385 | 850.002 | **1.27E−04** |
| Worst | 1.27E−04 | 1.27E−04 | 1.27E−04 | 910.6232 | 1.27E−04 | 1.27E−04 | 1.27E−04 | 592.1918 | 1550.2136 | 1.27E−04 |
| Mean | **1.27E−04** | **1.27E−04** | **1.27E−04** | 890.2521 | **1.27E−04** | **1.27E−04** | **1.27E−04** | 394.778 | 1157.0202 | **1.27E−04** |
| SD | 0 | 0 | 0 | 6.5668 | 0 | 0 | 0 | 193.413 | 237.0225 | 0 |
| Function | $f_5$ : Ackley | | | | | | | | | |
| Best | 2.58E−13 | 1.26E−13 | 0.0008 | 8.88E−16 | 8.06E−16 | 0.0013 | **7.15E−21** | 0.0014 | 1.0271 | 0.0013 |
| Worst | 9.75E−10 | 1.57E−12 | 1.0021 | 8.88E−16 | 8.06E−16 | 0.0016 | 8.52E−19 | 0.0016 | 2.4781 | 0.0016 |
| Mean | 5.07E−11 | 1.16E−11 | 0.0802 | 8.88E−16 | 8.06E−16 | 0.0015 | **3.25E−20** | 0.0015 | 1.5290 | 0.0015 |
| SD | 2.19E−12 | 2.51E−13 | 0.0360 | 0 | 0 | 0.0001 | 2.155E−20 | 0.0001 | 0.5184 | 0.0001 |
| Function | $f_6$ : Griewank | | | | | | | | | |
| Best | 3.91E−11 | **0** | 0.1201 | **0** | **0** | **0** | **0** | **0** | 0.0763 | **0** |
| Worst | 2.12E−04 | 0 | 0.1812 | 0 | 0 | 0 | 0 | 0.0393 | 0.1205 | 0 |
| Mean | 3.27E−05 | **0** | 0.1332 | **0** | **0** | **0** | **0** | 0.0190 | 0.0819 | **0** |
| SD | 7.40E−05 | 0 | 0.1002 | 0 | 0 | 0 | 0 | 0.0130 | 0.0091 | 0 |

**Table 12** continued

| Algorithm | LCA | FOA | SGO | SLCA | WCO | FGA | MVPA | GA | PSO | DE |
|---|---|---|---|---|---|---|---|---|---|---|
| Function | $f_7$ : GoldsteinPrice | | | | | | | | | |
| Best | **3.00E+0** | **3.00E+0** | **3.00E+0** | | **3.00E+0** | **3.00E+0** | **3.00E+0** | **3.00E+0** | **3.00E+0** | **3.00E+0** |
| Worst | 3.00E+0 | 3.00E+0 | 3.00E+0 | | 3.00E+0 | 3.00E+0 | 3.00E+0 | 3.00E+0 | 3.00E+0 | 3.00E+0 |
| Mean | **3.00E+0** | **3.00E+0** | **3.00E+0** | | **3.00E+0** | **3.00E+0** | **3.00E+0** | **3.00E+0** | **3.00E+0** | **3.00E+0** |
| SD | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | $f_8$ : Michalewicz | | | | | | | | | |
| Best | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** |
| Worst | –1.80E+0 | –1.80E+0 | –1.80E+0 | | –1.80E+0 | –1.80E+0 | –1.80E+0 | –1.80E+0 | –1.80E+0 | –1.80E+0 |
| Mean | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** | **–1.80E+0** |
| SD | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | $f_9$ : Schwefel22 | | | | | | | | | |
| Best | 2.01E−6 | 3.21E−5 | 4.10E−5 | 1.10E−08 | **2.11E−10** | 3.22E−5 | 2.14E−10 | 0.22E−6 | 1.15E−09 | 1.12E−08 |
| Worst | 8.85E−4 | 2.62E−4 | 5.85E−4 | 4.41E−08 | 7.18E−10 | 5.66E−4 | 2.21E−09 | 4.96E−6 | 4.51E−05 | 5.74E−08 |
| Mean | 4.59E−5 | 8.92E−5 | 7.36E−5 | 1.80E−08 | **4.25E−10** | 7.36E−5 | 9.58E−09 | 9.36E−6 | 3.21E−07 | 3.15E−08 |
| SD | 1.02E−7 | 0.32E−6 | 2.39E−6 | 1.98E−09 | 1.02E−11 | 2.39E−6 | 4.40E−09 | 2.39E−8 | 0.59E−06 | 2.21E−08 |

**Table 13** Dimensions of the benchmark functions for experiments E2, E3, E4, and E5

| Function | Dimension | | | |
|---|---|---|---|---|
| | E2 (2000 FEs) | E3 (5000 FEs) | E4 (10000 FEs) | E5 (25000 FEs) |
| $f_1$ : Sphere | 2 | 2 | 10 | 25 |
| $f_2$ : Rosenbrock | 2 | 2 | 10 | 25 |
| $f_3$ : Rastrigin | 2 | 2 | 10 | 25 |
| $f_4$ : Schwefel | 2 | 2 | 10 | 25 |
| $f_5$ : Ackley | 2 | 2 | 10 | 25 |
| $f_6$ :Griewank | 2 | 2 | - | - |
| $f_7$ : GoldsteinPrice | 2 | 2 | - | - |
| $f_8$ : Michalewicz | 2 | 2 | - | - |
| $f_9$ : Schwefel22 | 2 | 2 | 10 | 25 |

tively. The goal of E2 and E3 is to test the rapidity of the algorithms while the goal of E4 and E5 is to test the scalability of the algorithms, in other words how the algorithms cope with higher dimensional problems. Dimensions of the benchmark functions for this type of experiments are listed in Table 13. SLCA cannot be executed for E2 and E3 due to the two dimensions. Summary of the percentage of success and the number of FEs have been reported in Table 14. This table shows the overall success of each optimization algorithm considering 100 executions for every benchmark function. Detailed percentage of success (PE) and the number of FEs of the tested optimization algorithms for E2, E3, E4, and E5 have been given in Tables 15, 16, 17, and 18 respectively.

The performances of tested algorithms are also depicted in Figs. 19, 20, 21, and 22 for E2, E3, E4 and E5, respectively. In these figures, the overall successes of algorithms and FEs are drawn as bars.

From Table 14, it can be seen that for E2, MVPA has been able to solve, on average, 71.67% of all the unconstrained benchmark problems for all the 100 random starting search points using, on average, 1381 FEs. For E2, the overall success of MVPA is higher than the second-best algorithm WCO by 11.56% while MVPA has an average number of FEs lower than WCO by 50 evaluations. When considered the objective of E2, that testing the rapidity of algorithm convergences allowing a low number of FEs, MVPA is fast converging algorithm than the other algorithms. In E3 with higher number of FEs, DE seems better in terms of overall success. Sports based WCO seems better when both FEs and overall success have been considered.

When the dimensions of benchmark problems increase as in E4, it can be noticed that WCO has better performances over other tested search and optimization algorithms. MVPA has been able to solve, on average, 33% of the tested benchmark problems using, on average, 6598 FEs which is higher than the second best algorithm, WCO, by 3.17% while the number of FEs of the MVPA is a little bigger than the one of WCO by 23 evaluations.

Finally for E5, SLCA has been able to solve, on average, 20.33% of the unconstrained global benchmark problems using, on average, 10,021 FEs which is higher than the second best algorithm, the WCO, by 1.33% while the number of FEs of the SLCA is lower than that of WCO by 2723 evaluations. Therefore, the obtained results show clearly that sports

**Table 14** Summary of the performances of algorithms

| Algorithm | E2 | | E3 | | E4 | | E5 | |
|---|---|---|---|---|---|---|---|---|
| | Overal success (%) | FEs | Overal success (%) | *FEs* | Overal success (%) | FEs | Overal success (%) | FEs |
| LCA | 45.44 | 1334 | 58.56 | 2209 | 24 | 5314 | 16.67 | 13,706 |
| FOA | 47 | 1328 | 69.67 | 1912 | 21.50 | 3771 | 16.67 | 15,401 |
| SGO | 42.89 | 1321 | 67 | 2415 | 16.67 | 2598 | 15.83 | 18,954 |
| SLCA | | | | | 8.17 | 5375 | **20.33** | 10,021 |
| WCO | 60.11 | 1431 | 81.33 | 1435 | 29.83 | 6621 | 19 | 12744 |
| FGA | 52.89 | 1357 | 72.22 | 2736 | 19.33 | 6255 | 14.67 | 18250 |
| MVPA | **71.67** | 1381 | 76.67 | 1405 | **33** | 6598 | 16.67 | 13936 |
| GA | 0.89 | 180 | 9.11 | 218 | 0.67 | 8566 | 0 | - |
| PSO | 16.78 | 1718 | 78.11 | 3092 | 0 | - | 0 | - |
| DE | 47.33 | 1556 | **82.67** | 2041 | 0 | - | 0 | - |

**Table 15** Percentage of success (PS) and the number of functions evaluations (FEs) of algorithms for E2

| | | LCA | FOA | SGO | SLCA | WCO | FGA | MVPA | GA | PSO | DE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_1$ | PS | 100 | 100 | 100 | | 100 | 100 | 100 | 0 | 99 | 100 |
| | FEs | 681 | 798 | 912 | | 889 | 1021 | 778 | – | 1700 | 1148 |
| $f_2$ | PS | 0 | 42 | 0 | | 45 | 26 | 77 | 8 | 2 | 60 |
| | FEs | – | 1578 | – | | 1885 | 1002 | 1709 | 180 | 1521 | 1794 |
| $f_3$ | PS | 61 | 20 | 16 | | 19 | 41 | 65 | 0 | 0 | 66 |
| | FEs | 1502 | 1060 | 1580 | | 1150 | 1689 | 1362 | – | – | 1862 |
| $f_4$ | PS | 0 | 25 | 22 | | 52 | 56 | 73 | 0 | 0 | 62 |
| | FEs | – | 1685 | 1821 | | 1498 | 1589 | 1312 | – | – | 1816 |
| $f_5$ | PS | 100 | 100 | 100 | | 100 | 100 | 96 | 0 | 0 | 0 |
| | FEs | 850 | 996 | 963 | | 1956 | 895 | 1834 | – | – | – |
| $f_6$ | PS | 12 | 5 | 0 | | 15 | 0 | 13 | 0 | 0 | 0 |
| | FEs | 1680 | 1890 | – | | 1658 | – | 1736 | – | – | – |
| $f_7$ | PS | 100 | 100 | 100 | | 100 | 100 | 100 | 0 | 23 | 100 |
| | FEs | 1541 | 1254 | 1084 | | 960 | 1806 | 1050 | – | 1968 | 1619 |
| $f_8$ | PS | 14 | 16 | 12 | | 25 | 39 | 24 | 0 | 27 | 38 |
| | FEs | 1187 | 998 | 1325 | | 998 | 1156 | 775 | – | 1684 | 1102 |
| $f_9$ | PS | 22 | 15 | 36 | | 85 | 14 | 97 | 0 | 0 | 0 |
| | FEs | 1899 | 1698 | 1568 | | 1889 | 1698 | 1875 | – | – | – |

**Table 16** Percentage of success (PS) and the number of functions evaluations (FEs) of algorithms for E3

| | | LCA | FOA | SGO | SLCA | WCO | FGA | MVPA | GA | PSO | DE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_1$ | PS | 100 | 100 | 100 | | 100 | 100 | 100 | 0 | 100 | 100 |
| | FEs | 681 | 887 | 689 | | 625 | 1052 | 778 | – | 1704 | 1150 |
| $f_2$ | PS | 0 | 79 | 45 | | 95 | 85 | 100 | 82 | 100 | 100 |
| | FEs | – | 2065 | 1690 | | 1852 | 1260 | 1840 | 218 | 2648 | 1956 |
| $f_3$ | PS | 100 | 100 | 100 | | 100 | 100 | 64 | 0 | 84 | 100 |
| | FEs | 2010 | 2650 | 2069 | | 1506 | 4689 | 1384 | – | 3233 | 1941 |
| $f_4$ | PS | 0 | 55 | 16 | | 78 | 52 | 68 | 0 | 60 | 73 |
| | FEs | – | 1690 | 4521 | | 1530 | 1982 | 1317 | – | 3006 | 1838 |
| $f_5$ | PS | 100 | 100 | 100 | | 100 | 100 | 100 | 0 | 100 | 100 |
| | FEs | 850 | 888 | 3210 | | 963 | 4026 | 1842 | – | 4512 | 2853 |
| $f_6$ | PS | 22 | 12 | 27 | | 29 | 8 | 22 | 0 | 23 | 33 |
| | FEs | 4210 | 1658 | 1985 | | 1289 | 4250 | 1805 | – | 4108 | 3069 |
| $f_7$ | PS | 100 | 100 | 100 | | 100 | 100 | 100 | 0 | 100 | 100 |
| | FEs | 1530 | 1306 | 1965 | | 960 | 1806 | 1050 | – | 2287 | 1573 |
| $f_8$ | PS | 17 | 25 | 19 | | 30 | 49 | 36 | 0 | 40 | 38 |
| | FEs | 1985 | 2145 | 2652 | | 1568 | 2302 | 752 | – | 1689 | 1133 |
| $f_9$ | PS | 88 | 56 | 96 | | 100 | 56 | 100 | 0 | 96 | 100 |
| | FEs | 4201 | 3650 | 2689 | | 2630 | 3265 | 1879 | – | 4648 | 2862 |

**Table 17** Percentage of success (PS) and the number of functions evaluations (FEs) of algorithms for E4

| | | LCA | FOA | SGO | SLCA | WCO | FGA | MVPA | GA | PSO | DE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_1$ | PS | 100 | 100 | 100 | 0 | 100 | 100 | 100 | 0 | 0 | 0 |
| | FEs | 4602 | 2256 | 2598 | – | 4520 | 2985 | 3594 | – | – | – |
| $f_2$ | PS | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 4 | 0 | 0 |
| | FEs | – | – | – | – | 8241 | – | – | 8566 | – | – |
| $f_3$ | PS | 0 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 |
| | FEs | – | – | – | – | 6389 | – | – | – | – | – |
| $f_4$ | PS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | FEs | – | – | – | – | – | – | – | – | – | – |
| $f_5$ | PS | 22 | 29 | 0 | 30 | 32 | 0 | 33 | 0 | 0 | 0 |
| | FEs | 4984 | 5287 | – | 4523 | 4458 | – | 7184 | – | – | – |
| $f_9$ | PS | 22 | 0 | 0 | 19 | 36 | 16 | 65 | 0 | 0 | 0 |
| | FEs | 6356 | – | – | 6228 | 9500 | 9526 | 9018 | – | – | – |

**Table 18** Percentage of success (PS) and the number of functions evaluations (FEs) of algorithms for E5

| | | LCA | FOA | SGO | SLCA | WCO | FGA | MVPA | GA | PSO | DE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_1$ | PS | 100 | 100 | 95 | 100 | 100 | 88 | 100 | 0 | 0 | 0 |
| | FEs | 13,706 | 15,401 | 18,954 | 12,060 | 15,803 | 18,250 | 13,936 | – | – | – |
| $f_2$ | PS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | FEs | – | – | – | – | – | – | – | – | – | – |
| $f_3$ | PS | 0 | 0 | 0 | 22 | 14 | 0 | 0 | 0 | 0 | 0 |
| | FEs | – | – | – | 7982 | 9685 | – | – | – | – | – |
| $f_4$ | PS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | FEs | – | – | – | – | – | – | – | – | – | – |
| $f_5$ | PS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | FEs | – | – | – | – | – | – | – | – | – | – |
| $f_9$ | PS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | FEs | – | – | – | – | – | – | – | – | – | – |

based MVPA, WCO, and SLCA are the best optimization algorithms as far as considering the current experiments.

## 6 General discussion and evaluation

Development of new algorithms that exploit problem-specific knowledge to achieve a good performance over certain classes of optimization problems is an important task. Sports based algorithms have made great progresses in recent years. They are more recent and evolving computational intelligence methods that are gradually gaining attention every passing day within literature. Their usages in many fields seem very high. Many comprehensive studies of generalization and standardization in these algorithms should be performed in order to

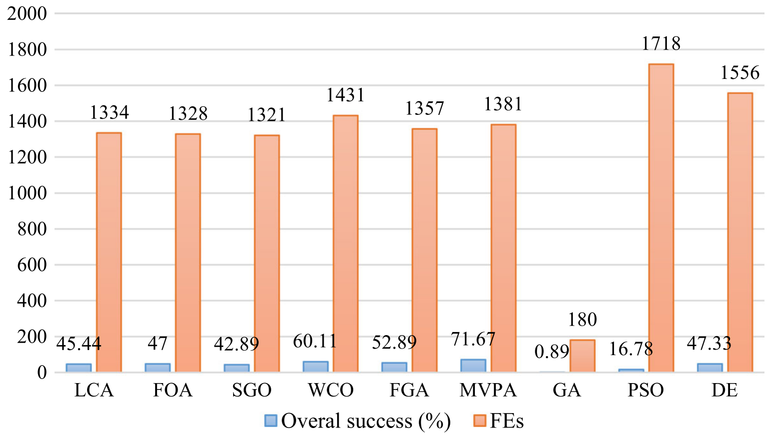


**Fig. 19** Performances of the algorithms for E2

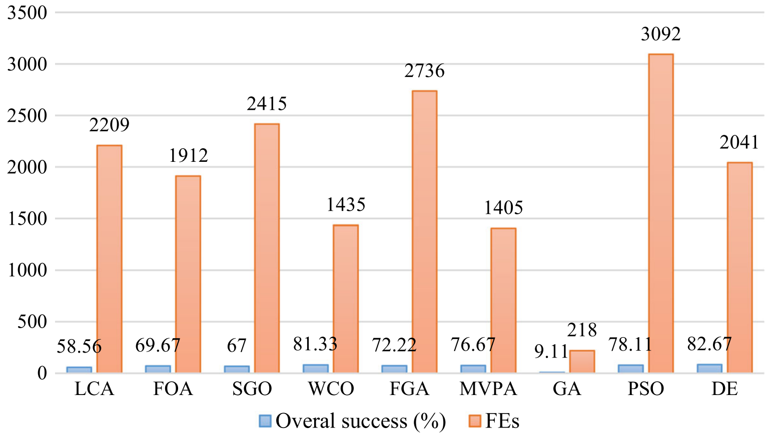


**Fig. 20** Performances of the algorithms for E3

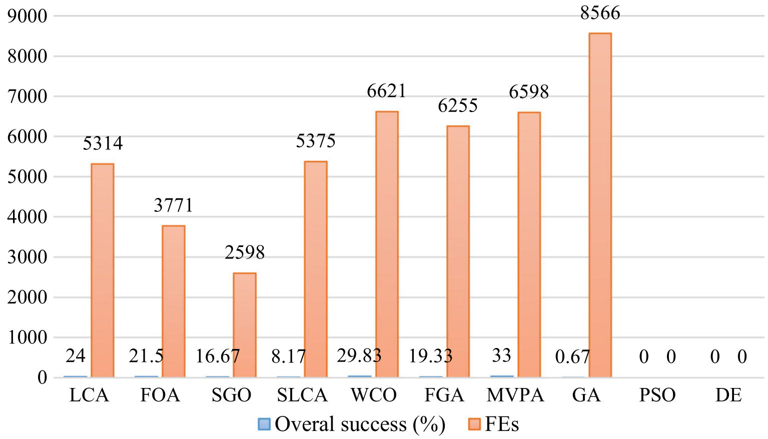


**Fig. 21** Performances of the algorithms for E4

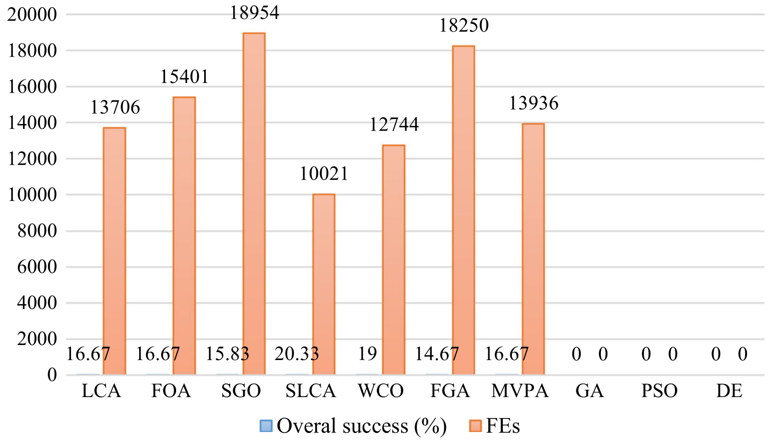


**Fig. 22** Performances of the algorithms for E5

greatly improve their applicability in many different fields. However it is also clear that, evaluating their generalization can be a complex task due to the diverse implementations.

In this study, all of the sports based computational intelligence search and optimization algorithms in the literature have been searched and collected for the first time to present a relatively comprehensive list and to inspire further research. Researches have experimentally shown that some sports based algorithms are better than other nature inspired methods. Especially, the oldest sports based algorithm, LCA, has been compared with genetic algorithm, differential evolution, bat algorithm, and particle swarm optimization within benchmark functions and real complex engineering problems. It has been reported that LCA is very efficient and has produced very promising and competitive results based on quality and reliability criteria and can compete in the best way with current popular algorithms (Kashan 2014). Algorithms should have mixing and diversity control among the solutions. Most of the sports based global optimization algorithms which have been searched and collected for the first time in this paper have been recently proposed and there is not any work about comparing all of their performances in the selected complex benchmark functions or real complex engineering problems in equal conditions. Equal conditions mean starting the search with the same conditions, using the same termination criteria, selecting the same benchmark function with equal dimensions and intervals, running the algorithms using the same programming language and hardware. A unified software may be developed for comparing the performances of all these new sports based algorithms over different types of complex benchmark functions.

More works should be performed in relatively more recent sports based algorithms in order to have the capability of both mixing and diversity control which facilitates to explore the vast search space efficiently and to converge more quickly. Some good and relatively old algorithms such as LCA and GBA seem to have both global search and intensive local search capabilities, which balance the exploration and exploitation, and they seem partly more efficient in unimodal and multimodal functions or problems. Solution quality of these relatively old algorithms in terms of mean objective and the standard deviation values indicates their consistency. More works should be performed for more recent algorithms such as WCO, SLO, and FOA in order to increase the performance in terms of robustness, convergence, precision, and general performance. SLO and FOA have not been well described

**Table 19** Frameworks for computational intelligence optimization algorithms

| Name | Web address |
|---|---|
| EO 1.2 (Keijzer et al. 2002) | http://eodev.sourceforge.net/ |
| EvA2 (Kronfeld et al. 2010) | http://www.ra.cs.uni-tuebingen.de/software/EvA2/ |
| JCLEC 4.0 (Ventura et al. 2008) | http://JCLEC.sourceforge.net/ |
| HeuristicLab 3.3 (Wagner 2009) | http://dev.heuristiclab.com/ |
| MALLBA 2.0 (Alba et al. 2007) | http://neo.lcc.uma.es/software/mallba/index.php/ |
| Opt4j 2.1 (Lukasiewycz et al. 2009) | http://opt4j.sourceforge.net/ |
| OAT 1.4 (Brownlee 2007) | http://optalgtoolkit.sourceforge.net/ |
| FOM (Parejo et al. 2003) | http://www.isa.us.es/fom |

and discussed. The experimental studies are not enough to conclude general evaluation about their performances. Formulations of SLO is vague without any pseudo-code. That is why, specific works should be done in order to increase the exploitation and exploration capabilities of SLO. Furthermore, after reviewing all of the sports based algorithms it can be seen that, some algorithms are just similar version of existing approaches, with a different name and different metaphor to be based on. Some algorithms contain similarities in terms of titles and parameters depending the same inspiration source (sport). For example FOA resembles FGA although they seem to perform different approaches for exploration and exploitation.

In this work, performance comparison of these sports based algorithms and other popular algorithms such as Genetic Algorithm, Particle Swarm Optimization, and Differential Evolution within unconstrained global optimization benchmark problems with different characteristics has been performed for the first time. The goal of some experiments has been determined as to test the rapidity of the algorithms while the goal of some experiments is to test the scalability of the algorithms, in other words how the algorithms cope with higher dimensional problems. MVPA is the best algorithms in unimodal benchmark function according to the performed experiments in this work. MVPA, WCO, and SLCA seem to be the best algorithms among other sports based and popular computational intelligence search algorithms as far as considering the current experiments on both unimodal and multimodal problems.

Reusable implementation of a number of computational intelligence algorithms may be useful and there are some frameworks for this task as shown in Table 19. These frameworks usually include solution encodings and specific-operators for different types of search and optimization problems. They also provide tools to monitor and evaluate the optimization and support specific applications to obtain the best set of parameters to tune the algorithms. These software or programs have some limitations. The majority of real optimization problems have peculiarities and specific constraints that are difficult to be taken into account directly within these frameworks. And they mainly implement the most classical approaches. The extension of these software to modern and novel sports based algorithms reviewed in this paper is possible and can be a very interesting research direction.

In future, more comprehensive evaluation of all sports inspired algorithms with graphical and tabular analysis may be focused on. These algorithms may extended for solving various other types of combinatorial, discrete, NP hard, multi-objective, complex optimization problems, etc. Their variants containing dynamic parameter selection, multi objectivity, different initial population methods, discrete versions, different termination criterion, and their hybrids with other methods may be proposed for efficient solutions. Researchers should be encouraged to implement a detailed performance analysis of these algorithms truly to pick up the best methods for different types of hard problems. Therefore, searching and collecting all of the sports based algorithms in the literature for the first time and inspiring more research to gain better insight into efficient algorithms and solve real-world large-scale, complex problems have been aimed with this paper.figu

It can be concluded that computational intelligence algorithms inspired from sports have much room to grow due to their novelties. Thus, this paper would act as a boon to the research community in identifying the research prospects in the field of computational intelligence.

## 7 Conclusions

In this paper, formational aspects of all the algorithms inspired from sports have been studied. This study will definitely be beneficial for new researchers and motivate them to formulate great solutions from inspiration of different phenomena to optimization problems. Performance comparison of these sports based algorithms and other popular algorithms such as Genetic Algorithm, Particle Swarm Optimization, and Differential Evolution within unconstrained global optimization benchmark problems with different characteristics has been performed for the first time in this paper.

These algorithms are very new computational methods and they can be improved in many ways. They can be combined with heuristic methods or other global optimization algorithm and their hybrid versions may be proposed for efficient results for different search and optimization problems. Using effective systematic techniques for initial candidate generation, modifying the operators for efficient exploration and exploitation, embedding external operators such as chaos, using different models such as islands or multiple populations, adjusting the parameters adaptively, using statistical information in search to reduce the computational complexity, and etc.

More validation studies should be performed to discover the capabilities of these algorithms in dealing with the search and optimization problems. Different comparisons test with other classical or computational intelligence algorithms using more complicated benchmark functions and complex constraint real world problems may illustrate the power and weakness of these sports based algorithms. Furthermore, sensitivity analysis of parameters of these algorithms and their effects on providing an efficient balance between exploitation and exploration steps can be another important research topic. Nonparametric statistical test should also be presented to conclude reliable and effective comparison results obtained in experimental studies of sports based optimization algorithms.

Computational intelligence algorithms have come a long way in terms of research and development. On the other hand; design, development, and implementation of novel algorithms is an important task under the philosophy of improvement in the scientific field and always searching better. There are positive challenges in terms of their efficiency and best possible use of these algorithms. Introduction and review that this paper provided may lead to many future contributions in exciting and emerging area of engineering problems. Research on different sports to enhance computational intelligence algorithms may be huge in the next years. Developing a unified framework for analyzing, evaluating, and comparing computational intelligence algorithms seem also an interesting and important task.

## References

Abdulhamid SM, Abd Latiff MS (2014) League Championship Algorithm based job scheduling scheme for infrastructure as a service cloud. In: 5th international graduate conference on engineering, science and humanities (IGCESH2014), Universiti Teknologi Malaysia, Johor Bahru, Malaysia

Abdulhamid SM, Abd Latiff MS, Abdullahi M (2015) Job scheduling technique for infrastructure as a service cloud using an improved league championship algorithm. In: The second international conference on advanced data and information engineering (DaEng-2015)

Abdulhamid SM, Abd Latiff MS, Ismaila I (2014) Tasks scheduling technique using league championship algorithm for makespan minimization in IAAS cloud. ARPN J Eng Appl Sci 9(12):2528–2533

Abdulhamid SM, Abd Latiff MS, Abdul-Salaam G, Hussain Madni SH (2016) Secure scientific applications scheduling technique for cloud computing environment using global league championship algorithm. PLoS ONE 11(7):1–18

Akyol S, Alatas B (2017) Plant intelligence based metaheuristic optimization algorithms. Artif Intell Rev 47(4):417–462

Akyol S, Alatas B (2016a) Efficiency evaluation of crow search algorithm in benchmark functions for optimization. In: 2nd international conference on engineering and natural sciences (ICENS), pp 939–944

Akyol S, Alatas B (2016b) Chaotically initiated flower pollination algorithm for search and optimization problems. In: 2nd international conference on engineering and natural sciences, pp 2934–2940

Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38(10):13170–13180

Alba E, Luque G, García-Nieto J, Ordonez GG, Leguizamon G (2007) Mallba: a software library to design efficient optimisation algorithms. Int J Innov Comput Appl 1:74–85

Ali J, Saeed M, Chaudhry NA, Luqman M, Tabassum MF (2015) Artificial showering algorithm: a new meta-heuristic for unconstrained optimization. Sci Int 27(6):4939–4942

Ashrafi SM, Dariane AB (2011) A novel and effective algorithm for numerical optimization: melody search (MS). In: 11th IEEE international conference on hybrid intelligent systems (HIS), pp 109–114

Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE congress on evolutionary computation, pp 4661–4667

Badrloo S (2015) A new method for solving combinatorial optimization problems with permutation based solution structure using league championship algorithm. M.Sc. Thesis, Azad University, Science and Research Branch, Iran (in Persian)

Bingol H, Alatas B (2016) Chaotic league championship algorithms. Arab J Sci Eng 41(12):5123–5147

Birbil SI, Fang SC (2003) An electromagnetism-like mechanism for global optimization. J Glob Optim 25:263–282

Borji A, Hamidi M (2009) A new approach to global optimization motivated by parliamentary political competitions. Int J Innov Comput Inf Control 5(6):1643–1653

Bouchekara HREH (2017) Most Valuable Player Algorithm: a novel optimization algorithm inspired from sport. Oper Res 1–57

Bouchekara HREH, Abido MA, Chaib AE, Mehasni R (2014a) Optimal power flow using the league championship algorithm: a case study of the Algerian power system. Energy Convers Manag 87:58–70

Bouchekara H, Abdallh A, Hamza Kherrab LD, Mehasni R (2014b) Design optimization of electromagnetic devices using the League Championship Algorithm. In: International workshops on optimization and inverse problems in electromagnetism (OIPE)

Brownlee J (2007) Oat: The Optimization Algorithm Toolkit, Technical Report, Complex Intelligent Systems Laboratory, Swinburne University of Technology

Cai W, Yang W, Chen X (2008) A global optimization algorithm based on plant growth theory: plant growth optimization. In: 2008 international conference on intelligent computation technology and automation (ICICTA), pp 1194–1199

Can U, Alatas B (2015) Physics based metaheuristic algorithms for global optimization. Am J Inf Sci Comput Eng 1(3):94–106

Chagwiza G, Jaison A, Masamha T (2016) Parameter improvement of the soccer league competition algorithm by introducing stubborn players: application to water distribution network. Math Prob Eng

Chu S-C, Tsai P-W, Pan J-S (2006) Cat swarm optimization. In: PRICAI 2006: trends in artificial intelligence. Springer, New York, pp 854–858

Chuang CL, Jiang JA (2007) Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space-time. In: 2007 IEEE congress on evolutionary computation, pp 3157–3164

Colak ME, Varol A (2015) A novel intelligent optimization algorithm inspired from circular water waves. Elektronika ir Elektrotechnika 21:3–6

Daskin A, Kais S (2011) Group leaders optimization algorithm. Mol Phys 109(5):761–772

De Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 6(3):239–251
Dorigo M, Maniezzo V, Colorni A (1991) The ant system: an autocatalytic optimizing process. Technical Report
Duarte A, Fernández F, Sánchez Á, Sanz A (2004) A hierarchical social metaheuristic for the max-cut problem. In: European conference on evolutionary computation in combinatorial optimization. Springer, Berlin, Heidelberg, pp 84–94
Edraki S (2014) A new approach for engineering design optimization of centrifuge pumps based on league championship algorithm. Science and Research Branch, Azad University, Tehran
Eyvazi M (2015) Portfolio optimization problem with multi-period investment readjustment using league championship algorithm. M.Sc. Thesis, Tarbiat Modares University, Iran (in Persian)
Fadakar E, Ebrahimi M (2016) A new metaheuristic football game inspired algorithm. In: 2016 1st IEEE conference on swarm intelligence and evolutionary computation (CSIEC), pp 6–11
Gálvez A, Iglesias A (2016) New memetic self-adaptive firefly algorithm for continuous optimisation. Int J Bio-Inspired Comput 8(5):300–317
Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
Genc HM, Eksin I, Erol OK (2010) Big bang - big crunch optimization algorithm Hybridized With Local Directional Moves and Application to Target Motion Analysis Problem. IEEE Int Conf Syst Man Cybern (SMC) 2010:881–887
Hatamzadeh P, Khayyambashi MR (2012a) Football optimization: an algorithm for optimization inspired by football game. In: ICSIl ISSSI, 2012, Kharazmi University
Hatamzadeh P, Khayyambashi MR (2012b) Neural network learning based on football optimization algorithm. In: Proceedings of the third international conference on contemporary issues in computer and information sciences (CICIS 2012) (8). Universal-Publishers
Holland JH, Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc., Boston
Hsiao YT, Chuang CL, Jiang JA, Chien CC (2005) A novel optimization algorithm: space gravitational optimization. In: 2005 IEEE international conference on systems, man and cybernetics, vol 3, pp 2323–2328
Ibrahim A, Rahnamayan S, Martin MV (2014) Simulated raindrop algorithm for global optimization. In: IEEE 27th Canadian conference electrical and computer engineering (CCECE), pp 1–8
Jalili S, Husseinzadeh Kashan A, Hosseinzadeh Y (2016) League championship algorithms for optimum design of pin-jointed structures. J Comput Civ Eng. doi:10.1061/(ASCE)CP.1943-5487.0000617
Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimisation problems. Int J Math Model Numer Optim 4(2):150–194
Jaramillo A, Crawford B, Soto R, Misra S, Olguín E, Rubio ÁG, Villablanca SM (2016b) An approach to solve the set covering problem with the soccer league competition algorithm. In: International conference on computational science and its applications. Springer, pp 373–385
Jaramillo A, Crawford B, Soto R, Villablanca SM, Rubio ÁG, Salas J, Olguín E (2016a) Solving the set covering problem with the soccer league competition algorithm. In: International conference on industrial, engineering and other applications of applied intelligent systems. Springer, pp 884–891
Jaramillo A, Gýmez A, Mansilla S, Salas J, Crawford B, Soto R, Olguýn E (2016c) Using the soccer league competition algorithm to solve the set covering problem. In: 11th Iberian conference on information systems and technologies (CISTI), pp 1–4
Javidy B, Hatamlou A, Mirjalili S (2015) Ions motion algorithm for solving optimization problems. Appl Soft Comput 32:72–79
Kahledan S (2014) A league championship algorithm for travelling salesman problem. Najaf Abad Branch, Azad University, Tehran
Kamarudin AA, Othman ZA, Sarim HM (2016) Water flow algorithm decision support tool for travelling salesman problem. In: Proceedings of the international conference on applied science and technology 2016 (ICAST'16), vol 1761(1). AIP Publishing
Karci A, Alatas B (2006) Thinking capability of saplings growing up algorithm. Intelligent data engineering and automated learning–IDEAL 2006, vol 4224. Lecture notes in computer Science. Springer, Berlin, pp 386–393
Kashan AH (2009) League Championship Algorithm: a new algorithm for numerical function optimization. In: SoCPaR, pp 43–48
Kashan AH, Karimi B (2010) A new algorithm for constrained optimization inspired by the sport league championships. In: IEEE congress on evolutionary computation, pp 1–8

Kashan AH, Karimiyan S, Karimiyan M, Kashan MH (2012) A modified League Championship Algorithm for numerical function optimization via artificial modeling of the "between two halves analysis". In: IEEE joint 6th international conference on soft computing and intelligent systems (SCIS) and 13th international symposium on advanced intelligent systems (ISIS), pp 1944–1949

Kashan AH (2011) An efficient algorithm for constrained global optimization and application to mechanical engineering design: league championship algorithm (LCA). Comput Aided Des 43(12):1769–1792

Kashan AH (2014) League Championship Algorithm (LCA): an algorithm for global optimization inspired by sport championships. Appl Soft Comput 16:171–200

Kaveh A (2014) Magnetic charged system search. In: Advances in metaheuristic algorithms for optimal design of structures. Springer, pp 87–134

Kaveh A, Bakhshpoori T (2016) Water evaporation optimization: a novel physically inspired optimization algorithm. Comput Struct 167:69–85

Keijzer M, Merelo JJ, Romero G, Schoenauer M (2002) Evolving objects: a general purpose evolutionary computation library. Artif Evol 2310:829–888

Kejani T (2013) A new approach for reliability optimization based on league championship algorithm (LCA). Najaf Abad Branch, Azad University, Tehran

Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: IEEE international conference on neural networks. Piscataway, pp 1942–1948

Khaji E. (2014) Soccer League Optimization: A heuristic Algorithm Inspired by the Football System in European Countries. arXiv preprint arXiv:1406.4462

Kiziloluk S, Alatas B (2012) Current social-based heuristic optimization algorithms. Cumhuriyet Univ J Econ Adm Sci 13(2):39–56

Kripka M, Kripka RML (2008) Big crunch optimization method. In: International conference on engineering optimization, Brazil, pp 1–5

Kronfeld M, Planatscher H, Zell A (2010) The EvA2 optimization framework. Lecture Notes in Computer Science, vol 6073. Springer, Berlin, pp 247–250

Labbi Y, Attous DB, Gabbar HA, Mahdad B, Zidan A (2016) A new rooted tree optimization algorithm for economic dispatch with valve-point effect. Int J Electr Power Energy Syst 79:298–311

Lam AY, Li VO (2010) Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans Evol Comput 14(3):381–399

Lenin K, Reddy BR, Kalavathi MS (2013) League championship algorithm (LCA) for solving optimal reactive power dispatch problem. Int J Comput Inf Technol 1(3):254–272

Lukasiewycz M, Glab FR, Helwig S (2009) Opt4: optimization framework for java. http://www.opt4j.org

Maniezzo V, Stützle T, Voss S (2009) Matheuristics: hybridizing metaheuristics and mathematical programming, vol 10. Springer, New York

Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inform 1:355–366

Meng X, Liu Y, Gao X, Zhang H (2014) A new bio-inspired algorithm: chicken swarm optimization. In: International conference in swarm intelligence. Springer, pp 86–94

Merrikh-Bayat F (2015) The runner-root algorithm: a metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. Appl Soft Comput 33:292–303

Mirjalili S (2016) SCA: a Sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

Moosavian N (2015) Soccer league competition algorithm for solving knapsack problems. Swarm Evol Comput 20:14–22

Moosavian N, Roodsari BK (2013) Soccer league competition algorithm, a new method for solving systems of nonlinear equations. Int J Intell Sci 4(1):7

Moosavian N, Roodsari BK (2014) Soccer league competition algorithm: a novel meta-heuristic algorithm for optimal design of water distribution networks. Swarm Evol Comput 17:14–24

Mora-Gutiérrez RA, Ramírez-Rodríguez J, Rincón-García EA (2014) An optimization algorithm inspired by musical composition. Artif Intell Rev 41(3):301–315

Murase H (2000) Finite element inverse analysis using a photosynthetic algorithm. Comput Electr Agr 29:115–123

Nedaie A, Khoshalhan F (2016) A new play-off approach in league championship algorithm for solving large-scale support vector machine problems. Int J Ind Eng Prod Res 27(1):61–68

Osaba E, Carballedo R, López-García P, Diaz F (2016) Comparison between Golden Ball Meta-heuristic, Evolutionary Simulated Annealing and Tabu Search for the Traveling Salesman Problem. In: Proceedings of the 2016 on genetic and evolutionary computation conference companion, ACM, pp 1469–1470

Osaba E, Diaz F, Onieva E (2014a) Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. Appl Intell 41(1):145–166
Osaba E, Diaz F, Carballedo R, Onieva E, Perallos A (2014b) Focusing on the golden ball metaheuristic: an extended study on a wider set of problems. Sci World J
Osaba E, Diaz F, Onieva E (2013) A novel meta-heuristic based on soccer concepts to solve routing problems. In Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. ACM, pp 1743–1744
Ozbay FA, Alatas B (2015) Review of social-based artificial intelligence optimization algorithms for social network analysis. Int J Pure Appl Sci 1:33–52
Ozbay FA, Alatas B (2016a) A simple and global physics based metaheuristic method: water evaporation optimization. In: 2nd international conference on engineering and natural sciences, pp 660–665
Ozbay FA, Alatas B (2016b) Review of computational intelligence method inspired from behavior of water. Afyon Kocatepe Univ J Sci Eng Spec Issue 137–147
Ozbay FA, Alatas B (2016c) Review of music based computational intelligence methods. 1st international conference on engineering technology and applied sciences (ICETAS), pp 663–669
Parejo J. A, Racero J, Guerrero F, Kwok T, Smith K (2003) Fom: a framework for metaheuristic optimization. In: Lecture Notes in Computer Science, vol 2660, Springer, pp 886–895
Pourali Z, Aminnayeri M (2011) A novel discrete league championship algorithm for minimizing earliness/tardiness penalties with distinct due dates and batch delivery consideration. In: International Conference on Intelligent Computing. Springer, Berlin Heidelberg, pp 139–146
Premaratne U, Samarabandu J, Sidhu T (2009) A new biologically inspired optimization algorithm. In: international conference on industrial and information systems (ICIIS), pp 279–284
Purnomo HD, Wee HM (2013) Soccer game optimization: an innovative integration of evolutionary algorithm and swarm intelligence algorithm. Meta-Heuristics optimization algorithms in engineering, business, economics, and finance. IGI Global, Pennsylvania
Purnomo HD (2014a) Soccer game optimization for continuous and discrete problems. Jurnal Metris 15(2):65–76
Purnomo HD (2014b) Soccer game optimization: fundamental concept. Jurnal Sistem Komputer 4(1):25–36
Purnomo HD, Wee HM (2015) Soccer game optimization with substitute players. J Comput Appl Math 283:79–90
Qi X, Zhu Y, Chen H, Zhang D, Niu B (2013) An idea based on plant root growth for numerical optimization. In: Intelligent computing theories and technology. Lecture Notes in Computer Science, vol 7996. Springer, pp 571–578
Rabanal P, Rodríguez I, Rubio F (2007) Using river formation dynamics to design heuristic algorithms. In: International conference on unconventional computation. Springer, Berlin, Heidelberg, pp 163–177
Ramezani F, Lotfi S (2013) Social-based algorithm (SBA). Appl Soft Comput 13(5):2837–2856
Rao RV, Savsani VJ, Vakharia DP (2012) Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183(1):1–15
Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
Razmjooy N, Khalilpour M, Ramezani M (2016) A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system. J Control Autom Electr Syst 1–22
Razmjooy N, Ramezani M (2016) Model Order Reduction based on meta-heuristic optimization methods. In: 1st international conference on new research achievements in electrical and computer engineering
Rezoug A, Boughaci D (2016) A self-adaptive harmony search combined with a stochastic local search for the 0–1 multidimensional knapsack problem. Int J Bio-Inspired Comput 8(4):234–239
Ruttanateerawichien K, Kurutach W, Pichpibul T (2014) An improved golden ball algorithm for the capacitated vehicle routing problem. Bio-Inspired Comput-Theor Appl. Springer, Berlin Heidelberg, pp 341–356
Ruttanateerawichien K, Kurutach W, Pichpibul T (2016) A new efficient and effective golden-ball-based technique for the capacitated vehicle routing problem. In: IEEE 15th international conference on computer and information science (ICIS), IEEE/ACIS, pp 1–5
Sacco WF, De Oliveira CR (2005) A new stochastic optimization algorithm based on a particle collision metaheuristic. In: Proceedings of 6th WCSMO
Sadollah A, Eskandar H, Bahreininejad A, Kim JH (2015) Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems. Appl Soft Comput 30:58–71
Sajadi SM, Kashan AH, Khaledan S (2014) A new approach for permutation flow-shop scheduling problem using league championship algorithm. In: Proceedings of CIE44 and IMSS, vol 14
Salcedo-Sanz S (2016) Modern meta-heuristics based on nonlinear physics processes: a review of models and design procedures. Phys Rep 655:1–70

Salem SA (2012) BOA: a novel optimization algorithm. In: IEEE 2012 international conference on engineering and technology (ICET), pp 1–5
Salhi A, Fraga ES (2011) Nature-inspired optimisation approaches and the new plant propagation algorithm. In: The international conference on numerical analysis and optimization (ICeMATH '11). Yogyakarta, Indonesia
Saraswathi D, Srinivasan E (2017) Mammogram analysis using league championship algorithm optimized ensembled FCRN classifier. Indones J Electr Eng Comput Sci 5(2):451–461
Sayoti F, Ri ME (2016) Golden ball algorithm for solving flow shop scheduling problem. Int J Artif Intell Interact Multim 4(1):15–18
Sayoti F, Riffi ME (2015) Random-keys golden ball algorithm for solving traveling salesman problem. Int Rev Model Simul (IREMOS) 8(1):84–89
Seyedhosseini SM, Badkoobehi H, Noktehdan A (2015) Machine-part cell formation problem using a group based league championship algorithm. J Promot Manag 21:55–63
Shah-Hosseini H (2009) The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. Int J Bio-Inspir Comput 1(1–2):71–79
Shah-Hosseini H (2011) Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. Int J Comput Sci Eng 6:132–140
Shahrezaee M (2017) Image segmentation based on world cup optimization algorithm. Majlesi J Electr Eng 11(2):39–45
Shi Y (2011) Brain storm optimization algorithm. In: International conference in swarm intelligence. Springer, Berlin, Heidelberg, pp 303–309
Stephen MJ, PV PR (2013) Simple league championship algorithm. Int J Comput Appl 75(6):28–32
Sun J, Wang X, Li K, Wu C, Huang M, Wang X (2013) An auction and league championship algorithm based resource allocation mechanism for distributed cloud. Int Workshop Adv Parall Process Technol. Springer, Berlin Heidelberg, pp 334–346
Surjanovic S, Bingham D (2013) Virtual Library of Simulation Experiments: Test Functions and Datasets. Retrieved May 11, 2017, from http://www.sfu.ca/ssurjano
Tayarani-N MH, Akbarzadeh-T MR (2008) Magnetic optimization algorithms a new synthesis. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) 2659-2664
Thammano A, Moolwong J (2010) A new computational intelligence technique based on human group formation. Expert Syst Appl 37(2):1628–1634
Ventura S, Romero C, Zafra A, Delgado J, Hervás C (2008) JCLC: a java framework for evolutionary computation. Soft Comput 2(4):381–392
Wagner S (2009) Heuristic optimization software systems modeling of heuristic optimization algorithms in the heuristic lab software environment (Ph.D. thesis), Johannes Kepler University, Linz
Wang H, Wang W, Sun H, Rahnamayan S (2016) Firefly algorithm with random attraction. Int J Bio-Inspir Comput 8(1):33–41
Xie L, Tan Y, Zeng J, Cui Z (2010) Artificial physics optimisation: a brief survey. Int J Bio-Inspir Comput 2(5):291–302
Xing B, Gao WJ (2014) Central force optimization algorithm. In: Innovative computational intelligence: a rough guide to 134 clever algorithms. Springer, pp 333–337
Xing B, Gao WJ (2014) Charged system search algorithm. In: Innovative computational intelligence: a rough guide to 134 clever algorithms. Springer, pp 339–346
Xu Y, Cui Z, Zeng J (2010) Social emotional optimization algorithm for nonlinear constrained optimization problems. In: International Conference on Swarm, Evolutionary, and Memetic Computing. Springer, Berlin Heidelberg, pp 583–590
Xu W, Wang R, Yang J (2015b) An improved league championship algorithm with free search and its application on production scheduling, Journal of Intelligent Manufacturing
Xu W, Yang J, Wang R (2015a) An Intelligent Method for Evaluation of Production Scheduling Performance. International Conference on Intelligent Systems Research and Mechatronics Engineering (ISRME 2015), 1121-1126
Yang X-S (2012) Flower Pollination Algorithm for global optimization. In: Unconventional computation and natural computation. Springer. 240–249
Yang FC, Wang YP (2007) Water flow-like algorithm for object grouping problems. J Chin Inst Ind Eng 24:475–488
Zhang H, Zhu Y, Chen H (2014) Root growth model: a novel approach to numerical function optimization and simulation of plant root system. Soft Comput 18:521–537

Zhao Z, Cui Z, Zeng J, Yue X (2011) Artificial plant optimization algorithm for constrained optimization problems. In: 2011 Second international conference on innovations in bio-inspired computing and applications (IBICA), 120–123

Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55:1–11

Zhou Y, Wang Y, Chen X, Zhang L, Wu K (2016) A Novel path planning algorithm based on plant growth mechanism. Soft Comput 1–11