International Journal of Intelligence Science, **2014, 4, 7-16**
Published Online January 2014 (http://www.scirp.org/journal/ijis)
http://dx.doi.org/10.4236/ijis.2014.41002

Soccer League Competition Algorithm, a New Method for Solving Systems of Nonlinear Equations

Naser Moosavian, Babak Kasaee Roodsari
Department of Civil Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
Email: naser.moosavian@yahoo.com

Received October 9, 2013; revised November 9, 2013; accepted November 15, 2013

ABSTRACT

This paper introduces Soccer League Competition (SLC) algorithm as a new optimization technique for solving nonlinear systems of equations. Fundamental ideas of the method are inspired from soccer leagues and based on the competitions among teams and players. Like other meta-heuristic methods, the proposed technique starts with an initial population. Population individuals called players are in two types: fixed players and substitutes that all together form some teams. The competition among teams to take the possession of the top ranked positions in the league table and the internal competitions between players in each team for personal improvements results in the convergence of population individuals to the global optimum. Results of applying the proposed algorithm in solving nonlinear systems of equations demonstrate that SLC converges to the answer more accurately and rapidly in comparison with other Meta-heuristic and Newton-type methods.

KEYWORDS

Soccer League Competition; Nonlinear Equations; Meta-Heuristic Algorithm

1. Introduction

Solving systems of nonlinear equations is one of the main concerns in a diverse range of engineering applications such as computational mechanics, weather forecast, hydraulic analysis of water distribution systems, aircraft control and petroleum geological prospecting. Many previous efforts have been made to find a solution for systems of nonlinear equations. Results of these studies comprise some theories and algorithms [1-4]. Among such approaches, Newton's method is one of the most powerful numerical methods and an important basic method which has a quadratic convergence if the function F is continuously differentiable and if a good initial guess x_0 is provided [5]. Frontini and Sormani [6] proposed a third-order method based on a quadrature formula to solve systems of nonlinear equations. Cordero and Torregrosa [7] developed some variants of Newton's method based on trapezoidal and midpoint rules of quadrature. Also, Darvishi and Barati [8-10] presented some high order iterative methods and Babajee *et al.* [11] proposed a fourth-order iterative technique. Luo *et al.* [12] solved a system of nonlinear equations using a combination of chaos search and Newton-type methods. More recently, Mo *et al.* [5] presented a combination of the conjugate direction method (CD) and particle swarm optimization (PSO) for solving systems of nonlinear equations.

The convergence and performance characteristics of Newton-type methods are highly sensitive to the initial guess of the solution supplied to the methods and the algorithm would fail if the initial guess of the solution is improper. However, it is difficult to select a good initial guess for most systems of nonlinear equations [13]. The system of nonlinear equations is considered as follows:

$$\begin{cases} f_1(x_1,x_2,\cdots,x_n)=0 \\ f_2(x_1,x_2,\cdots,x_n)=0 \\ \quad\vdots \\ f_n(x_1,x_2,\cdots,x_n)=0. \end{cases} \tag{1}$$

Applying the global optimization methods, the system of Equation (1) is transformed to an optimization problem. This is achieved by using the auxiliary function:

$$\min F(\boldsymbol{x}) = \sum_{i=1}^{n} f_i^2(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, x_2, \cdots, x_n) \tag{2}$$

Global minimum of above formulation is zero and $\boldsymbol{x}^*$ is a root for the corresponding system of equations if $F(x^*) = 0$. This paper presents a new meta-heuristic algorithm, called Soccer League Competitions (SLC), for solving Equation (2).

In Section 2, the basic concepts of SLC are defined. In Section 3, the performance and effectiveness of SLC are validated by some examples. Finally, the conclusions are presented in Section 4.

2. Soccer League Competitions

Level one soccer league consists of teams (clubs) competing each other during a season. In this environment, some stronger teams aim to sit in the first positions of the league table while some weaker teams plan to survive in the level one league in order to prevent a crash out to the second level league. During the course of a season, each team plays the others twice, once at their home stadium and once at that of their opponents. Teams receive 3 points for a win and no point is awarded for a lost. Teams are weekly ranked by total points and the club with the most point is crowned champion at the end of each season. The number of matches in each season depends on the team numbers. For instance, in a league consisting of M teams the total number of matches is calculated as follows:

$$\text{Total Match} = \left(M \times (M-1)\right)/2 \tag{3}$$

In this league, each team participates in $M-1$ independent matches, and totally, $\left(M \times (M-1)\right)/2$ competitions are being held during a season.

There is always an intense competition between the teams at the bottom of the league table. As a rule, the two bottom table teams are crashed out to the second level soccer league (relegations spots) at the end of the season. In return, two first table teams of the second level league (promotions spots) replaced with the relegated teams. Generally, promotions spots import new players to the league which may have potential of being a future star.

Each team consists of 11 fixed players (FP) and some substitutes (S). A team's power depends on the power of its players. Moreover, powerful teams have a higher chance of winning their matches. However, it is not possible to predict the exact winner of a specified match before the game ends.

As well as the league competitions among teams, there is an internal competition in each team. Players compete with each other to attract the head coach's attention by improving their performance. This internal competition leads to a growth in the quality and power of a team.

In each team, there is a key player which is called Star Player (SP). SP has the best performance among other players in the team. Moreover, there is a unique player in each league which is called the Super Star Player (SSP). SSP is defined as the most powerful player in the league.

After every match, players included in winner and loser teams of each match adopt different strategies for improving their future performance. When a team wins a match, fix players try to imitate the team's SP, and the SSP of the league (this strategy is simulated by Imitation Operator in this study). They aim to experience a promotion to the SP or, optimistically, occupy the place of SSP in the league. But, the main provocation of winner's substitutes is being a fixed player in the team. For this purpose, they try to have a performance approximately equal to the average level of fixed players in the team (this tendency is described by the Provocation Operator in this study). In other words, higher provocation for advancement gives them more chance of being a fixed player in the future.

On the other hand, 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. This revision may include a change in some aspects of their older habits (this strategy is defined as the Mutation Operator in this study). In addition, head coach usually considers new combinations of substitutes in order to stop the failures in the future (this change is performed by the Substitution Operator in this study).

Above mentioned strategies improve the overall performance of the teams after each match. Therefore, team's powers progressively increase while all teams play much better at the end of the season. Obviously, players with a noticeable progression increase the winning chance of their team.

As the first rank teams of each league have better financial affordance, they are able to recruit powerful players of other teams. This intensifies their power for future seasons. In the next section, the solving style of an optimization problem using the Soccer League Competition (SLC) algorithm is discussed.

2.1. Soccer League Competition (SLC) Algorithm

Competitions between teams in a soccer league for reaching success, and among players for being a SP or SSP can be simulated for solving optimization problems. Similar to a soccer league in which every player desires to be the best (SSP), in an optimization problem each solution vector seeks for the global optimum position. Therefore, each player in a league, Star Player (SP) in each team, and the Super Star Player (SSP) can be assumed as a solution vector, a local optimum, and the global optimum, respectively.

Each team consists of 11 fixed players (defined by principal solution vectors in the SLC algorithm) and some substitutes (described by reserved solution vectors in SLC algorithm). For each player, an objective function is calculated which stands for the power of its corresponding player. In a minimization problem smaller values of objective functions (cost function) illustrate powerful players (*PP*). The total power of a team is defined as the average power value of its players including fixed and substitute. The following formula shows how a Team's Power (*TP*) is calculated.

$$TP(i) = (1/\text{nPlayer}) \sum_{j=1}^{\text{nPlayer}} PP(i,j) \tag{4}$$

nPlayer is the total number of players in the *i*th team. $PP(i,j)$ is the power of *j*th player in the *i*th team $(PP(i,j) = 1/\text{cost}(i,j))$. In each match, the team with more power has a higher chance of winning. The probability of victory for each team in a match is given by:

$$Pv(k) = TP(k)/(TP(i)+TP(k)) \tag{5}$$

$$Pv(i) = TP(i)/(TP(i)+TP(k)) \tag{6}$$

Pv stands for the probability of victory. It should be noted that the sum of *Pv(k)* and *Pv(i)* equals 1.

After each match, the winner and the loser are noticed and some players (solution vectors), including fixed and substitute, experience changes. These changes, which are aimed to improve performance of both players and teams, are simulated with the following operators:

-Imitation Operator

-Provocation Operator

-Mutation Operator

-Substitution Operator

In the next part, detailed description of operators is defined.

2.1.1. Imitation Operator

Fixed players (FP) of the winner team, imitate both the Star Player (SP) in their own team and the Super Star Player (SSP) in the league to improve their future activities. Similarly, 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. In the SLC algorithm, Imitation is performed by the following formulas:

$$FP(i,j) = \mu_1 FP(i,j) + \tau_1 (SSP - FP(i,j)) + \tau_2 (SP(i) - FP(i,j)) \tag{7}$$

$$FP(i,j) = \mu_2 FP(i,j) + \tau_1 (SSP - FP(i,j)) + \tau_2 (SP(i) - FP(i,j)) \tag{8}$$

where $\mu_1 \sim U(\theta,\beta)$, $\mu_2 \sim U(0,\theta)$, $\tau_1 \sim U(0,2)$, and $\tau_2 \sim U(0,2)$ are random numbers with uniform distribution. $FP(i,j)$ stands for the *j*th fixed player of the ith team, and $SP(i)$ is the star player of the *i*th team. It is also proposed that: $1 \le \beta \le 2$, $0 \le \theta \le 1$.

First, solution vector of fixed players (FP) in the winner team experiences a big move toward the resultant vector direction of SP and SSP (Equation (7)). If the newly generated solution vector at this new position was better than the older solution vector, it is replaced with the old one. Otherwise, the solution vector experiences a medium move toward the resultant vector (Equation (8)). If this solution was better than the older one, it is replaced with the old vector. In the case that none of the discussed movements gave a better solution vector, the player is kept in its position with no change.

2.1.2. Provocation Operator

Substitutes of a winner team (*S*) have to prove a performance equal to the average performance level value of the fixed players in their team in order to be a fixed player. This process, which is performed by the Provocation Operator in SLC algorithm, is described by

$$S(i,j) = C(i) + \chi_1 (C(i) - S(i,j)) \tag{9}$$

$$S(i,j) = C(i) + \chi_2 (S(i,j) - C(i)) \tag{10}$$

where $\chi_1 \sim U(0.9,1)$, $\chi_2 \sim U(0.4,0.6)$ are random numbers with uniform distribution, and $C(i)$ is the average value of fixed player's solution vectors in the ith team. *S*(*i*,*j*) is the *j*th substitute of the *i*th team.

Firstly, solution vector of the weakest substitute player in the winner team experiences a forward move toward the gravity center of fixed players (Equation (9)). If the newly generated solution vector relating to this new position was better than the last one, it is replaced by its old vector. Otherwise, mentioned player will experience a backward movement toward the gravity center (Equation (10)). If this solution was better than the weakest solution, this vector is replaced with the old one. In the situation that none of the discussed movements gave a better solution vector for improving the weakest solution, a new vector is generated randomly and replaced with the old one. In an overall view, provocation operator acts on the weakest substitutes of winners. If advancement was evident in their performance, they are kept in the team. Otherwise, they are exported from the team while new random players (solution vectors) are entered for future games.

2.1.3. Mutation Operator

Fixed players of loser team in a match should revise their activity in order to prevent failure in future games. To perform this operation, the positions of some players are randomly changed. This mechanism is similar to mutation process in Genetic Algorithm (GA) for creating diversification in solutions.

2.1.4. Substitution Operator

The head coach usually considers new combinations of substitutes for future games. Similarly, a random-based approach is applied to reflect the head coach impact in this algorithm. To do this, a pair of new substitute vectors is being tested. If a suitable answer was obtained, this effective pair is entered to the team. This process, which is performed by the Substitution Operator in SLC algorithm, is described by

$$S_{NEW}(i,j) = \alpha \times S(i,j) + (1-\alpha) \times S(i,k) \quad (11)$$

$$S_{NEW}(i,k) = \alpha \times S(i,k) + (1-\alpha) \times S(i,j) \quad (12)$$

$\alpha \sim U(0,1)$ is a random vector with uniform distribution. The number of new examined pairs is proposed to be equal to the number of team substitutes.

In an overall view, 4 described operators have the following effects in the algorithm:

The Imitation Operator expedites the searching capability of the algorithm.

The Provocation Operator provides high accurate solutions to the complex optimizations problems.

The Mutation and Substitution Operators help the proposed algorithm to escape from local minimums and plateaus.

After each game, 4 discussed operators act on the players (solution vectors) and team's powers are updated according to the new solutions. Obviously, powerful teams are more likely to be successful in their future matches. This process continues to the end of the season and the Super Star Player (SSP) of the league yields the Global Optimum (best solution) for the optimization problem. After each season, players are arranged taking into account their updated power. Before commencing a new league, top players are devoted to the best teams, medium players are allocated to the teams with an average performance, and the weakest players are transferred to the bottom teams in the league table.

In the next section, the steps and flowchart of SLC algorithm are presented.

2.2. Steps and Flowchart of SLC Algorithm

Step 1. Initialize the problem and algorithm parameters

In this step, the optimization problem is specified as follows:

$$\text{Min}F(\boldsymbol{x}) = \sum_{i=1}^{n} f_i^2(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, x_2, \cdots, x_n) \quad (9)$$

where $F(x)$ is an objective function; x is a set of each decision variable x_i; n is the number of decision variables; and X_i is the set of possible range of values for each decision variable. Then, the number of seasons (nSeason), the number of teams included in the league (nTeam), the number of fixed players (nFixedPlayer), and the number of substitutes (nSubstitute) are determined.

Step 2. Generate samples

The total number of players in a league is calculated by the following formula:

$$\text{nPlayers} = \text{nTeam} \times (\text{nFixedPlayer} + \text{nSubstitute})$$

In most problems, it is suggested that

$$3 \le \text{nTeam} \le 5,$$

$$\text{nFixedPlayer} = 11,$$

$$\text{nSubstitute} = 11$$

In this step, randomly solution vectors are generated as many as the number of players in the league and each vector is devoted to a specified player. Hence the matrix TEAM which is generated randomly is given as:

$$\text{TEAM} = \begin{bmatrix} FP_1 \\ FP_2 \\ \vdots \\ FP_{\text{nFixedPlayer}} \\ RP_1 \\ RP_2 \\ \vdots \\ RP_{\text{nSubstitute}} \end{bmatrix} = \begin{bmatrix} xF_1^1 & xF_2^1 & \cdots & xF_N^1 \\ xF_1^2 & xF_2^2 & \cdots & xF_N^2 \\ \vdots & \vdots & & \vdots \\ xF_1^{\text{nFixedPlayer}} & xF_2^{\text{nFixedPlayer}} & \cdots & xF_N^{\text{nFixedPlayer}} \\ xR_1^1 & xR_2^1 & \cdots & xR_N^1 \\ xR_1^2 & xR_2^2 & \cdots & xR_N^2 \\ \vdots & \vdots & & \vdots \\ xR_1^{\text{nSubstitute}} & xR_2^{\text{nSubstitute}} & \cdots & xR_N^{\text{nSubstitute}} \end{bmatrix} \quad (10)$$

Next, an objective function relating to each solution vector (player's power) is calculated.

Step 3. Teams assessment

In this step, all players are arranged according to their calculated power and are devoted to teams. Each team's power is equal to the average power of its players.

Step 4. Start the league

In this step, competitions are started between all possible pairs of teams in the league, the winner and the loser of every match are determined, the Imitation Operator acts on fixed players in winner teams, the Provocation Operator acts on substitutes of winner teams, the Mutation Operator acts on 3 out of 11 from fixed players of loser teams, and the Substitution Operator acts on reserved players in the loser teams. Then, player's powers and the team's powers are updated. This process is continued by the end of the season.

Step 5. Relegation and promotion

In this step, the worst team (relegation spot) is exported from the first level league, and in return, a new team (promotion spot) is imported to this league. It should be noted that this step is only applied for complex optimization problems.

Step 6. Check the stopping criterion

In this section, Steps 3, 4, and 5 are repeated until the termination criterion (nSeason) is satisfied.

Figure 1 illustrates the flowchart of SLC procedure for solving optimization problems.

3. Numerical Results

In this section, the solutions for some systems of nonlinear equations are described. All of computations were executed in MATLAB programming language environment using five independent nonlinear systems with an Intel(R) Core(TM) 2Duo CPU P8700 @ 2.53 GHz and 4.00 GB RAM. For the examples 3 to 5, the stopping criterion is considered to be $F(\boldsymbol{x}_n) < 10^{-3}$.

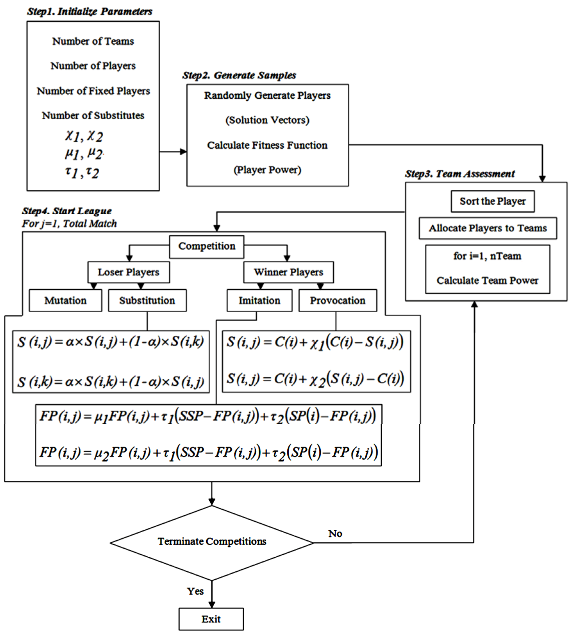
In all problems it is considered that $\beta = 1$ and $\theta = 0.7$.

Case study 1. Geometry size of thin wall rectangle girder section:

$$f_1(x) = bh - (b-2t)(h-2t) = 165$$

$$f_2(x) = \frac{bh^3}{12} - \frac{(b-2t)(h-2t)^3}{12} = 9369$$

$$f_3(x) = \frac{2(h-t)^2(b-t)^2}{h+b-2t} = 6835,$$



Step1. Initialize Parameters
Number of Teams
Number of Players
Number of Fixed Players
Number of Substitutes
χ_1, χ_2
μ_1, μ_2
τ_1, τ_2
Step2. Generate Samples
Randomly Generate Players
(Solution Vectors)
Calculate Fitness Function
(Player Power)
Step3. Team Assessment
Sort the Player
Allocate Players to Teams
for i=1, nTeam
Calculate Team Power
Step4. Start League
For j=1, Total Match
Competition
Loser Players
Winner Players
Mutation
Substitution
Imitation
Provocation
$S(i,j) = \alpha \times S(i,j) + (1-\alpha) \times S(i,k)$
$S(i,k) = \alpha \times S(i,k) + (1-\alpha) \times S(i,j)$
$S(i,j) = C(i) + \chi_1(C(i) - S(i,j))$
$S(i,j) = C(i) + \chi_2(S(i,j) - C(i))$
$FP(i,j) = \mu_1 FP(i,j) + \tau_1(SSP - FP(i,j)) + \tau_2(SP(i) - FP(i,j))$
$FP(i,j) = \mu_2 FP(i,j) + \tau_1(SSP - FP(i,j)) + \tau_2(SP(i) - FP(i,j))$
Terminate Competitions
No
Yes
Exit

Figure 1. SLC procedure for solving optimization problems.

where h is the height, b is the width and t is the thickness of the section. Mo *et al.* [5] solved this system using conjugate direction particle swarm optimization. In another study, Luo *et al.* [12] presented a solution for mentioned system using a hybridization of chaos search and Newton-type methods. Also, Jaberipour *et al.* [13] used a new version of particle swarm optimization (PPSO) for solving this system. In this research, the SLC algorithm is applied to solve above-mentioned problem. The bound variables were set between 0 and 30 m. As shown in Table 1, different intervals of χ_1, χ_2 are examined to find the best performance of SLC algorithm. As it can be seen in this problem, if $\chi_1 = 1$ and $\chi_2 = 0.5$, SLC reaches to the average accuracy level of 10^{-24} after 100000 function evaluations and 20 independent runs. Therefore, we set $\chi_1 = 1$ and $\chi_2 = 0.5$. According to the Table 2, the best solution equals zero that is the global optimum of this problem. In Table 2, values of the best, worst, mean, standard deviation and number of function evaluations are considered after 20 different runs. It is assumed that number of teams, fixed players, and substitutes equal 3, 6 and 6, respectively. In case A, all operators are taken into account in SLC algorithm while in case B mutation and substitution operators are exempted from the algorithm. As can be seen in case A, the best solution equals zero which is the exact value of the global optimum, but in case B, the best solution equals 1.29e^{-26}.

To verify the performance of the proposed algorithm, Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithms are applied to solve this system. As it can be seen in Table 3, the best solution of DE after 100,000 function evaluations equals 0.639, and PSO reaches to 267.594 and 1.89e^{-25} after 10,000 and 100,000 function evaluations, respectively. It should be noted that PSO algorithm can converge to the global optimum in one out of 20 runs. The initial population in DE and PSO are considered to be 20, and 300, respectively.

Table 4 presents the solution obtained from SLC and previous studies. According to the results, the SLC pro-

Table 1. Comparison results (Objective Functions) of SLC for different values of χ_1 and χ_2 (Number of function evaluations = 100,000).

χ_1	χ_2	best	worst	mean	Std[a]
0.5 - 1.5	0 - 1	0.1781	48.6241	9.9851	11.5017
0.7 - 1.2	0.3 - 0.7	4.567E−09[b]	17614[b]	916.7857[b]	3931.8
0.8 - 1.1	0.4 - 0.6	1.26E−21	4.63E−08	2.48E−09	1.03E−08
0.9 - 1	0.45 - 0.55	4.35E−24	1.19E−11	5.94E−13	2.66E−12
1	0.5	0	7.91E−24	1.38E−24	1.87E−24

[a]Standard Deviation. [b]Value of objective function.

Table 2. **Reviewing effects of different parameters for SLC in cases A and B.**

SLC	best	worst	mean	std	FCN
Case A					
3[a]					
6[b]	0[d]	7.91E−24	1.38E−24	1.87E−24	10,000
6[c]					
Case A					
	best	worst	mean	std	FCN
3					
6	0	1.87E−24	4.76E−25	4.91E−25	100,000
6					
Case B					
	best	worst	mean	std	FCN
3					
6	1.29E−26	9.25E−25	4.86E−25	3.18E−25	10,000
6					
Case B					
	best	worst	mean	std	FCN
3					
6	5.17E−26	3.32E−24	6.02E−25	7.28E−25	100,000
6					

[a]Number of teams. [b]Number of fixed players. [c]Number of substitutes. [d]Value of objective function.

Table 3. **Results of PSO and DE for case study 1.**

	best	worst	mean	std	FCN
DE	0.639[a]	3023.6[a]	663.8763[a]	949.29	100000
PSO	267.594	10000	9513.4	2176.2	10000
PSO	1.89E−25	10000	8500	3663.5	100000

[a]Value of objective function.

Table 4. **Comparison of SLC solutions with other methods in case study 1.**

Methods	B	h	t	$f_1(x)$	$f_2(x)$	$f_3(x)$
SLC	22.057	20.294	2.1705	165	9369	6835
PPSO [8]	43.156	10.129	12.944	709.24	9369	528.04
Mo *et al.* [11]	8.9431	23.271	12.913	165	9369	529.32
Luo *et al.* [10]	12.566	22.895	2.7898	166.72	9544.3	2585.5

vides exact solution and outperforms other discussed methods.

Case study 2. Consider

$$F(\boldsymbol{x}) = \left(f_1(x), f_2(x), \cdots, f_m(x)\right) \text{ with}$$

$$f_1(x) = (3-5x_1)x_1 + 1 - 2x_2 = 0$$
$$f_i(x) = (3-5x_i)x_i + 1 - x_{i-1} - 2x_{i+1} = 0,\ i = 2,\cdots,9$$
$$f_{10}(x) = (3-5x_{10})x_{10} + 1 - x_9 = 0,$$

The above system has ten unknown variables and ten equations.

SLC was used for solving this system. Maximum and minimum of decision variables were set between −1 and 0. In Tables 5 and 6, performance of SLC is verified for different number of players and teams. Values of χ_1, χ_2 are assumed based on explanations of section 2.1.2 of this paper. Due to the results of both case A and B, the best performance is reached when the number of teams, fixed players, and substitutes are assumed to be 5, 10,

Table 5. **Reviewing effects of different parameters for SLC in case A.**

SLC	Best	worst	mean	std	FCN
Case A[a]					
3					
10	3.63E−25	1.89E−05	1.49E−06	4.51E−06	10,000
10					
	Best	worst	mean	std	FCN
5					
10	9.40E−13	1.06E−06	5.29E−07	2.36E−07	10,000
10					

[a]SLC with all operators.

Table 6. **Reviewing effects of different parameters for SLC in case B.**

SLC	best	worst	mean	std	FCN
Case B[a]					
3					
10	7.64E−31	1.66E+00	1.67E−01	5.25E−01	10000
10					
	best	worst	mean	std	FCN
5					
10	4.70E−20	4.76E−12	5.08E−13	1.21E−12	10000
10					

[a]SLC without mutation and substitution operators.

and 10, respectively. In other words, better solutions are reached when the number of fixed players and substitutes equals the number of decision variables. PSO and DE algorithms are also used to solve this system. As shown in Table 7, the mean solution value of DE (considering initial population = 20) equals 7.12e−12, while PSO reaches to 2.72e−4 and 1.22e−7 when the initial population equals 100 and 1000, respectively. It should be mentioned that the mean solution of SLC equals 5.08e−13 (Table 6).

Table 8 demonstrates the best solution obtained from SLC algorithm, and compares this solution with results of Differential Equation (DE) and Particle Swarm Algorithm (PSO). It should be noted that the number of function evaluation in all methods is equal 10,000. It is obvious from Table 8 that SLC finds the exact solution more accurately comparing with other discussed methods.

Case study 3. Consider $F(\boldsymbol{x}) = \left(f_1(x), f_2(x), \cdots, f_m(x)\right)$ with

$$f_i(x) = x_i x_{i+1} - 1, \quad i = 1, 2, \cdots, m-1$$
$$f_m(x) = x_m x_1 - 1,$$

When m is odd, the exact zeros of $F(\boldsymbol{x})$ are (1, 1, ⋯,1) and (−1, −1, ⋯,−1). Shin *et al.* [14] solved the above system for various values of m using Newton-Like methods. They set the initial guess to be (0.5, 0.5, ⋯,0.5) for all methods.

Table 7. **Results of DE and PSO for case study 2.**

	best	worst	mean	std	FCN
DE	2.89E−13	4.13E−11	7.12E−12	1.23E−11	10,000
PSO[a]	2.76E−08	0.0025	2.72E−04	6.16E−04	10,000
PSO[b]	6.61E−08	7.27E−07	1.22E−07	1.54E−07	10,000

[a]Population = 100. [b]Population = 1000.

Table 8. **Comparison of SLC solutions with other methods in case study 2.**

Methods	Mo *et al.* [11]	DE	PSO	SLC
x_1	0.91555	−0.382084413	−0.382101391	−0.382084304
x_2	−0.22226	−0.438097433	−0.438106147	−0.438097493
x_3	−0.41465	−0.445927406	−0.445938190	−0.445927622
x_4	−0.43925	−0.446971289	−0.446966223	−0.446971297
x_5	0.42089	−0.446951503	−0.446961182	−0.446951485
x_6	−0.35459	−0.446355699	−0.446377379	−0.446355653
x_7	−0.13577	−0.444141223	−0.444154249	−0.444141159
x_8	0.42756	−0.436187399	−0.436180807	−0.436187334
x_9	0.75220	−0.407859019	−0.407836044	−0.407858897
x_{10}	−0.44070	−0.309566750	−0.309547753	−0.309566879
$f_1(x)$	−3.17E−06	−8.6491E−07	−9.9243E−05	2.2204E−16
$f_2(x)$	3.52E−07	1.2254E−07	−2.5652E−05	0.0000E+00
$f_3(x)$	−1.70E−06	1.5398E−06	−8.0323E−05	8.8818E−16
$f_4(x)$	1.77E−06	−1.2364E−07	6.7860E−05	1.2212E−15
$f_5(x)$	−1.68E+00	−5.1561E−08	−3.4056E−05	−3.2196E−15
$f_6(x)$	2.53E+00	−1.9788E−07	−1.2628E−04	−1.2212E−15
$f_7(x)$	−8.42E−01	−3.0441E−07	−8.8738E−05	−2.7756E−15
$f_8(x)$	−3.91E−07	−1.6868E−07	1.5435E−05	−2.2204E−16
$f_9(x)$	6.81E−07	−1.0542E−06	1.1699E−04	0.0000E+00
$f_{10}(x)$	2.34E−07	9.0578E−07	9.3726E−05	−6.8834E−15

To verify the performance of SLC algorithm abovementioned problem will be analyzed for 13, 71, 151, and 201 dimensions. We assume zero and one as the upper and lower bounds of unknown variables in SLC algorithm $(0.5 \le x_i \le 1.5)$. Values of χ_1, χ_2 are assumed based on explanations of section 2.1.2 of this paper. For each considered dimension value (m), the number of substitute players equal the number of decision variables while the number of fixed players in 3 cases equals the substitute players and in one case equals half of this value. Calculation results for 20 independent runs are presented in **Tables 9-12**. Similar to the previous examples, analysis is performed for 2 cases of A and B. In case A, all operators are taken into account in SLC algorithm while in case B mutation and substitution operators are exempted from the algorithm. According to **Tables 9** and **10**, the performance of SLC algorithm in case B is better than case A when the number of decision variables are $m = 13$ (number of function evaluations = 1057) and $m = 71$ (number of function evaluations = 10,602). In contrast, SLC algorithm in case A has a better performance when the number of decision variables are $m = 151$ (number of function evaluations = 75,137) and $m = 201$(number of function evaluations = 104,370).

Table 9. Review of parameter variation effect in SLC algorithm for m = 13.

	Case A	Case B		Case A	Case B
3			5		
13	1631[a]	**1057**[a]	13	3294[a]	2226[a]
13			13		

[a]Number of function evaluations.

Table 10. Review of parameter variation effect in SLC algorithm for m = 71.

	Case A	Case B		Case A	Case B
9			3		
71	70,024	63,377	35	23,827	**10,602**
71			71		

[a]Number of function evaluations.

Table 11. Review of parameter variation effect in SLC algorithm for m = 151.

	Case A	Case B		Case A	Case B
9			3		
151	191,370	253,330	75	**75,137**	100,460
151			151		

Table 12. Review of parameter variation effect in SLC algorithm for m = 201.

	Case A	Case B		Case A	Case B
9			5		
201	278,920	404,265	100	**104,370**	151,860
201			201		

To verify the performance of SLC algorithm, PSO and DE algorithms are applied to solve this system of nonlinear equations. As shown in **Table 13**, SLC has better convergence accuracy in comparison with PSO considering all different problem dimensions. It is also found that DE is not a good rival for SLC and PSO at all. For instance, this algorithm has not yet converged to the solution after more than 1 million function evaluations for m = 151 and m = 201. To reach the best performance for PSO and DE, the initial population in DE algorithm equals half of its number of decision variables and in PSO algorithm this value is assumed to be 100* number of decision variables. The best solution of DE after 100,000 function evaluations equals 0.639, and PSO reaches to 267.594 and 1.89e−25 after 10,000 and 100,000 function evaluations, respectively. It should be noted that PSO algorithm can converge to the global optimum in one out of 20 runs. The initial population in DE and PSO are considered to be 20, and 300, respectively.

The CPU time results are provided in **Table 14**. As shown in **Table 14**, the convergence rate in Newton-like methods dramatically increases as the problem's dimension (number of unknown variables) rises. In contrast, the convergence time in SLC has no dependency with the dimension parameter.

It should be mentioned that each linear system of equations should be solved in alliterations of the solving process [14]. The main reasons in less convergence time in SLC method is that it does not require solving linear system of equations and it only calls the optimization function during each season.

Considering the discussed examples in this article, SLC can properly solve huge system of nonlinear equations. To sum up, the following suggestions are worth to mention after detail preview of the problems:

1) Choose the values of χ_1 and χ_2 close to 1 and 0.5, respectively.
2) The number of teams should be selected between 3 and 5 for problems with dimension of lower than 200.
3) The number of fixed players should be selected between the number of decision variables and half of this value.
4) The number of substitute players should be equal to the number of decision variables.
5) To decrease the number of function evaluation in problems with small dimensions, Mutation and Subs-

Table 13. Results of DE and PSO for case study 3.

DE	$m = 13$	$m = 71$	$m = 151$	$m = 201$
FCN	70,400	269,600	1,000,000	1,000,000
$F(x)$	0.001	0.001	0.79828	8.2763
PSO	$m = 13$	$m = 71$	$m = 151$	$m = 201$
FCN	4290	58,220	135,900	261,300
$F(x)$	0.001	0.001	0.001	0.001

Table 14. Comparison results of CPU time for SLC with other methods in case study 3.

Method	$m = 13$	$m = 71$	$m = 151$	$m = 201$
	CPU time(s)	CPU time(s)	CPU time(s)	CPU time(s)
N-K [14]	0.484375	7.828125	75.03125	272.546875
Ned1 [14]	0.5	11.890625	121.578125	461.8125
Ned2 [14]	0.53125	14.46875	172.046875	588.90625
Ned3 [14]	0.375	10.28125	166.71875	431.984375
mNm [14]	0.546875	12.109375	138.28125	790.21875
FM [14]	0.640625	13.75	199.046875	802.640625
CL [14]	0.421875	8.609375	109.09375	501.546875
SLC	0. 3121	1.9216	9.0952	16.9021

titution operators can be neglected.

To find a solution, SLC algorithm rapidly reaches the local optimums and considers the best of them as the global optimum solution using Imitation operator. Next, the local optimum solution is precisely approximated by the provocation operator. During the above-mentioned operations, both Mutation and Substitution operators check the skewed points to prevent the ignorance of any other possible local or global optimum points in the domain. Combination of 4 discussed operators together with the team ranking procedures in leagues, make SLC algorithm as an incomparable optimizer among many other meta-heuristic and mathematical methods.

4. Conclusion

In this article, a new meta-heuristic algorithm was introduced, entitling Soccer League Competitions (SLC), to solve nonlinear systems of equations. This algorithm seeks for the answer using 4 independent operators and rapidly converges to the results. Due to the comparison results between the proposed algorithm with other meta-heuristic and Newton-like methods, SLC provides more accurate answers in a considerably smaller time. Furthermore, SLC has the lowest sensitivity to the problem's dimensions. In conclusion, the proposed algorithm is recommended for huge and complex optimization problems and nonlinear systems of equations specifically when the running time is considered as an important factor.

REFERENCES

[1] W. R. J. Ortega, "Iterative Solution of Nonlinear Equation in Several Variable," Academic Press, New York, 1970.

[2] S. Krzyworzcka, "Extension of the Lanczos and CGS Methods to Systems of Nonlinear Equations," *Journal of Computational and Applied Mathematics*, Vol. 69, No. 1, 1996, pp. 181-190. http://dx.doi.org/10.1016/0377-0427(95)00032-1

[3] S. J. J. Nocedal, "Numerical Optimization," Spring Science + Business Media Inc., Berlin, 1999.

[4] D. G. Huang and Y. Ma, "Nonlinear Numerical Analysis," Wuhan University Press, Wuhan, 2000.

[5] Y. Mo, H. Liu and Q. Wang, "Conjugate Direction Particle Swarm Optimization Solving Systems of Nonlinear Equations," *Computers & Mathematics with Applications*, Vol. 57, No. 11-12, 2009, pp. 1877-1882. http://dx.doi.org/10.1016/j.camwa.2008.10.005

[6] M. Frontini and E. Sormani, "Third-Order Methods from Quadrature Formulae for Solving Systems of Nonlinear Equations," *Applied Mathematics and Computation*, Vol. 149, No. 3, 2004, pp. 771-782. http://dx.doi.org/10.1016/S0096-3003(03)00178-4

[7] A. Cordero and J. R. Torregrosa, "Variants of Newton's

Method for Functions of Several Variables," *Applied Mathematics and Computation*, Vol. 183, No. 1, 2006, pp. 199-208. http://dx.doi.org/10.1016/j.amc.2006.05.062

[8] M. T. Darvishi and A. Barati, "Super Cubic Iterative Methods to Solve Systems of Nonlinear Equations," *Applied Mathematics and Computation*, Vol. 188, No. 2, 2007, pp. 1678-1685. http://dx.doi.org/10.1016/j.amc.2006.11.022

[9] M. T. Darvishi and A. Barati, "A Fourth-Order Method from Quadrature Formulae to Solve Systems of Nonlinear Equations," *Applied Mathematics and Computation*, Vol. 188, No. 1, 2007, pp. 257-261. http://dx.doi.org/10.1016/j.amc.2006.09.115

[10] M. T. Darvishi and A. Barati, "A Third-Order Newton-Type Method to Solve Systems of Nonlinear Equations," *Applied Mathematics and Computation*, Vol. 187, No. 2, 2007, pp. 630-635. http://dx.doi.org/10.1016/j.amc.2006.08.080

[11] D. K. R. Babajee, *et al.*, "A Note on the Local Convergence of Iterative Methods Based on Adomian Decomposition Method and 3-Node Quadrature Rule," *Applied Mathematics and Computation*, Vol. 200, No. 1, 2008, pp. 452-458. http://dx.doi.org/10.1016/j.amc.2007.11.009

[12] Y.-Z. Luo, G.-J. Tang and L.-N. Zhou, "Hybrid Approach for Solving Systems of Nonlinear Equationsusing Chaos Optimization and Quasi-Newton Method," *Applied Soft Computing*, Vol. 8, No. 2, 2008, pp. 1068-1073. http://dx.doi.org/10.1016/j.asoc.2007.05.013

[13] M. Jaberipour, E. Khorram and B. Karimi, "Particle Swarm Algorithm for Solving Systems of Nonlinear Equations," *Computers & Mathematics with Applications*, Vol. 62, No. 2, 2011, pp. 566-576. http://dx.doi.org/10.1016/j.camwa.2011.05.031

[14] B. C. Shin, M. T. Darvishi and C.-H. Kim, "A Comparison of the Newton-Krylov Method with High Order Newton-Like Methods to Solve Nonlinear Systems," *Applied Mathematics and Computation*, Vol. 217, No. 7, 2010, pp. 3190-3198. http://dx.doi.org/10.1016/j.amc.2010.08.051