ITC 3/48
Journal of Information Technology and Control
Vol. 48 / No. 3 / 2019
pp. 373-388
DOI 10.5755/j01.itc.48.3.20627

A Physics-Based Novel Approach for Travelling Tournament Problem: Optics Inspired Optimization

Received 2018/04/22 | Accepted after revision 2019/07/16

http://dx.doi.org/10.5755/j01.itc.48.3.20627

A Physics-Based Novel Approach for Travelling Tournament Problem: Optics Inspired Optimization

B. Alatas, H. Bingol
Department of Software Engineering; Firat University; Elazig, Turkey
phone: +9042423700000; e-mails: balatas@firat.edu.tr, harun_bingol@hotmail.com

Corresponding author: balatas@firat.edu.tr

Computational intelligence search and optimization algorithms have been efficiently adopted and used for many types of complex problems. Optics Inspired Optimization (OIO) is one of the most recent physics inspired computational intelligence methods which treats the search space of the problem to be optimized as a wavy mirror in which each peak is assumed to reflect as a convex mirror and each valley to reflect as a concave one. Each candidate solution is treated as an artificial light point that its glittered ray is reflected back by the search space of the problem and the artificial image is formed based on mirror equations adopted from Optics, as a new candidate solution. In this study, OIO for the first time has been designed as a solution search strategy for travelling tournament problem which is one of the current sport problems and aids to minimize transportation and total movement of teams. Furthermore, this problem has been firstly solved by League Championship Algorithm and obtained results from both synthetic and real datasets have been compared. By this study, new application areas for OIO and LCA have been introduced. Obtained results show the superiority of OIO which is a novel algorithm and seems to efficiently solve many complex problems.

KEYWORDS: Computational intelligence optimization, optics inspired optimization, travelling tournament problem, artificial intelligence.

1. Introduction

Most of the search and optimization methods require mathematical models of the system. Establishing a mathematical model for complex systems is often difficult. Even if the model is established, the solution time cannot be used due to the huge cost. Classical search and optimization algorithms are insufficient for complex large scale combinatorial and nonlinear search and optimization problems.

Such algorithms are not effective in adapting them to interested problems. This, in many cases, requires some assumptions that may be difficult to validate. Often due to the natural solution mechanisms of classical search methods, the problem concerned is modeled such that the method will manage it. The solution strategy of classical optimization methods is usually dependent on the type of objectives and constraints and on the type of variables. Their effectiveness is also highly dependent on the solution space in the problem model, the number of constraints, and number of decision variables. Another important shortcoming is that they cannot present general solution purpose strategies that can be utilized in the case of different types of variables, objectives, and constraints. In other words, classical methods solve models that have a specific type of objective function or constraint functions. However, many optimization problems in management, sports, engineering, economy, computer, etc., require different types of variables, objective functions, and constraints in formulations simultaneously. Therefore, computational intelligence optimization methods are purposed and efficiently adapted. These methods have become very popular in recent years because they are computationally powerful and their transformations are easy [1].

General-purpose computational intelligence search algorithms are divided into various groups such as biology-based, social-based, chemical-based, physics-based, music-based, mathematics-based, sports-based, swarm-based, plant-based, and water-based. Their combinations can also be considered as hybrid category. Genetic algorithm, ant colony algorithms, and differential evolution algorithm are biologically based; simulated annealing algorithm and charged system search algorithm are physics based; human mental search is social based; artificial chemical reaction optimization algorithm is chemistry-based and musical composition method is music-based algorithm and models [3].

Optics Inspired Optimization (OIO) is one of the most recent population-based physics inspired algorithms proposed by Kashan [23, 24]. OIO is inspired by the optical characteristics of concave and convex mirrors that can be utilized to solve different types of complex large scaled problems. When the light rays strike on the concave mirror, they reflect towards the principal axis and converge. When the light rays fall on the convex mirror, they reflect away from the principal axis and diverge. Exploration and exploitation capabilities of OIO are controlled by these concave and convex mirror phenomena.

In this study, OIO has been used as a solution search strategy for the first time in the Travelling Tournament Problem (TTP), which aims to minimize the moving problems of the current sports problems and the total movement of the sports teams. The obtained results are compared with the League Championship Algorithm (LCA) which is one of the most recent sports inspired artificial intelligence optimization algorithms [25].

The organization of this work is as follows. In Section 2, information about computational intelligence optimization algorithms and their advantages are discussed. Physics-based OIO is briefly introduced in Section 3. In Section 4, TTP is explained in detail with examples. In Section 5, designing of OIO for the TTP that confronts the league is explained. The performance of the OIO optimized TTP is experimentally investigated in the real and synthetic dataset and the performance comparisons with the sports-based LCA are presented for the first time. Section 6 contains comments on what kind of problems OIO can be efficiently used for and what can be done in future works.

2. Computational Intelligence Search and Optimization

In most real-life problems, the solution space of the problem is infinite or so large that all solutions cannot be evaluated. For this to be acceptable, it is necessary to create and evaluate the candidate solutions and find a good solution within an acceptable time. The evaluation of solutions in such a way that they are acceptable for such problems actually means the evaluation of "some solutions" in the entire solution space. The way in which some solutions are chosen and how they are selected varies according to the computational intelligence technique [2, 3, 5].

Computational intelligence algorithms provide general solution strategies when the optimization problem may have a structure in which the exact solution finding process cannot be identified. When a mathematical model cannot be constructed or when constructed model has different types of variables, objective functions, and constraint functions, they can also be efficiently utilized. Their computing power is good and their transformations are easy. They are adaptable for different types of complex problems.

For clarity, computational intelligence algorithms can be much simpler in terms of decision makers. They can be used as part of learning and precise solution finding. Definitions made by mathematical formulas often ignore the most difficult parts of real-world search and optimization problems (what objectives and what constraints should be used, which alternatives should be tested, how to gather problem data). In the case that the data used in determining the model parameters have noise, worse solutions may be obtained than the suboptimal solution that the computational intelligence search techniques can produce. Due to many advantages, computational intelligence algorithms are densely and efficiently being used as search strategy in many search and optimization problems.

General-purpose computational metaheuristic search and optimization algorithms can be divided to ten categories according to different inspiration fields: as sociology, music, physics, biology, swarm, sports, chemistry, water, plant, and mathematics. Categorization is depicted in Table 1.

Table 1

Computational intelligence search and optimization methods

Computational Intelligence Algorithms	
Physics-based	Optics inspired optimization, Ions motion optimization
Music-based	Harmony search, Musical composition algorithm
Swarm-based	Particle swarm optimization, Ant colony algorithm
Biology-based	Genetic algorithm, Clonal selection algorithm
Math-based	Base optimization algorithm, Golden sine algorithm
Plant-based	Plant growth optimization
Social based	Parliamentary optimization algorithm, Social based algorithm
Water-based	Water drops algorithm
Sports based	League championship algorithm
Chemistry based	Artificial chemical reaction optimization algorithm

Physics-based computational intelligence optimization methods mimic physical rules [9]. The most popular physics-based methods are magnetic optimization algorithm [46], ions motion optimization [22], central force optimization algorithm [47], and OIO [23, 24]. Social based optimization algorithms are inspired by behaviors of people, human learning mechanism, and many features associated with the social situation of the people [36, 28]. Some of the popular social based optimization algorithms are named as Parliamentary optimization algorithm [8], teaching-learning based optimization [39], and social based algorithm [38].

Harmony search [19] and musical composition algorithm [33] are music based methods. Artificial chemical reaction optimization algorithm [4] is a chemistry- based method. Genetic algorithm [21] and clonal selection algorithm [14] are well-known biological based algorithms. Base optimization algorithm [42] and golden sine algorithm [45] are mathematics based methods.

Particle swarm optimization [26], chicken swarm optimization [32], and ant colony algorithm [15] are some of the popular swarm inspired search and optimization algorithms proposed inspiring from swarm intelligence systems in nature.

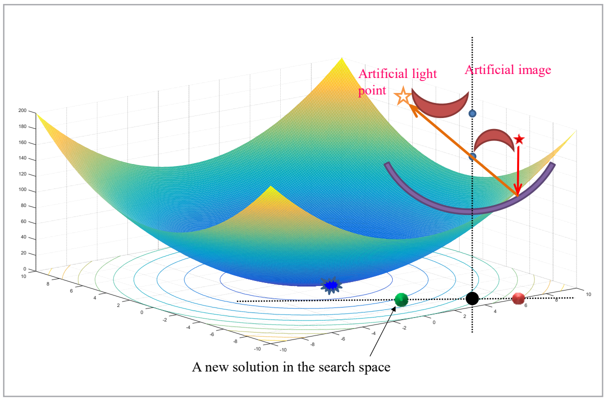
Plant-based algorithms have been proposed by inspiration from plant intelligence [3]. Water-based algorithms have been proposed inspiring from the process in hydrology [35].

3. Optics Inspired Optimization

OIO is one of the most recent population-based physics inspired algorithms proposed by Kashan [23-25]. OIO is inspired by the optical characteristics of concave and convex mirrors that can be utilized to solve different types of complex search problems. When the light rays strike on the concave mirror, they reflect towards the principal axis and converge. When the light rays fall on the convex mirror, they reflect away from the principal axis and diverge. Exploration and exploitation capabilities of OIO are controlled by these concave and convex mirror phenomena. The pictorial representation of an artificial image formation (new candidate solution generation) is shown in Figure 1

Figure 1

Demonstration of the law of reflection when light rays fall on convex mirror [23]



where each valley represents the concave mirror and peek represents a convex mirror [23, 29].

The candidate solutions in OIO are represented by artificial light points (ALP). Their rays strike at the mirror and reflect back to form artificial images (new candidate solutions) with the help of artificial mirror (AM). The reflected surface can be convex or concave. The variables used in OIO equations are described in Table 2.

A number of artificial light points (points in R_{n+1} whose mapping in R_n are candidate solutions for the interested problem) are assumed to be sitting in front of an artificial wavy mirror (function surface) reflecting their images. OIO treats the surface of the function to be minimized or maximized as the reflecting mirror composed of peaks and valleys. Each valley is considered as a concave reflective surface and each peak is considered as a convex reflective surface [24].

Distances and radius have been computed as in Equations (1) and (2). After computing these values, Equation (3) is applied to obtain q^t_{j,i_k}.

$$p^t_{j,i_k}=s^t_{j,i_k}-f\left(\vec{F}^t_{i_k}\right) \quad (1)$$

$$r^t_{i_k}=m^t_{i_k}-f\left(\vec{F}^t_{i_k}\right) \quad (2)$$

$$\frac{2}{r^t_{i_k}}=\frac{1}{p^t_{j,i_k}}+\frac{1}{q^t_{j,i_k}} \Rightarrow q^t_{j,i_k}=\frac{r^t_{i_k}p^t_{j,i_k}}{2p^t_{j,i_k}-r^t_{i_k}}. \quad (3)$$

Given the fact that

$$HO^t_{j,i_k}=\left\|\vec{o}^t_j-\vec{F}^t_{i_k}\right\|, \quad (4)$$

the image height of the ALP j can be calculated as in Equation (5):

$$HI^t_{j,i_k}=-HO^t_{j,i_k}\frac{q^t_{j,i_k}}{p^t_{j,i_k}}. \quad (5)$$

Table 2

Meaning of variables and formulations

Formulations and Variables	Meaning
$\vec{O}^t_j=[O^t_{j1}O^t_{j2}\ldots O^t_{jn}]_{1xn}$	Position of artificial light point (ALP) j in the n-dimensional search space in iteration t (the jth solution in the population).
$\vec{F}^t_{i_k}=[f^t_{ik^1}f^t_{ik^2}\ldots f^t_{ik^n}]_{1xn}$	Individual in the population which passes the artificial principal axis through itself
$\vec{I}^t_j=[I^t_{j1}I^t_{j2}\ldots I^t_{jn}]_{1xn}$	Image position of the ALP j in iteration t.
S^t_{j,i_k}	Position of the ALP j on the function/objective axis in iteration t.
p^t_{j,i_k}	Distance between the position of ALP j on the function/objective axis and the position of artificial mirror (AM)
q^t_{j,i_k}	Distance between the image position of the ALP j on the function/objective axis and the position of AM vertex on the function/objective axis at iteration t.
r^t_{j,i_k}	Radius of curvature of the AM
$m^t_{i_k}$	Position of the center of curvature on the function/objective axis
HO^t_{j,i_k}	Height of the ALP j from artificial principal axis in iteration t.
HI^t_{j,i_k}	Image height of the ALP j from artificial principal axis in iteration t.
K^t_{j,i_k}	Value of lateral aberration relevant to the AM which is reflecting the image of the ALP j in iteration t.

Image position of the ALP j in iteration t can be computed as in Equation (6):

$$\vec{I}^{t}_{j,i_k}=\vec{F}^{t}_{i_k}+HI^{t}_{j,i_k}\frac{(\vec{o}^{t}_{j}-\vec{F}^{t}_{i_k})}{\left\|\vec{o}^{t}_{j}-\vec{F}^{t}_{i_k}\right\|}=\vec{F}^{t}_{i_k}-HO^{t}_{j,i_k}\frac{q^{t}_{j,i_k}}{p^{t}_{j,i_k}}\frac{(\vec{o}^{t}_{j}-\vec{F}^{t}_{i_k})}{\left\|\vec{o}^{t}_{j}-\vec{F}^{t}_{i_k}\right\|}$$
$$\Rightarrow \vec{I}^{t}_{j}=\vec{F}^{t}_{i_k}-\frac{q^{t}_{j,i_k}}{p^{t}_{j,i_k}}(\vec{o}^{t}_{j}-\vec{F}^{t}_{i_k})=\vec{F}^{t}_{i_k}-\frac{r^{t}_{i_k}}{2p^{t}_{j,i_k}-r^{t}_{i_k}}(\vec{o}^{t}_{j}-\vec{F}^{t}_{i_k}). \quad (6)$$

A new consequent image position (candidate solution) is calculated according to Equations (7) and (8):

$$\vec{I}^{t}_{j}=\sum_{k=1}^{K} w^{t}_{k}\,\vec{I}^{t}_{j,i_k} \quad (7)$$

$$\sum_{k=1}^{K} w^{t}_{k}=1,\ 0<w^{t}_{k}\leq 1. \quad (8)$$

Equation (9) is used to calculate the value of lateral aberration relevant to an AM which is reflecting the image of the ALP j in iteration t:

$$\vec{K}^{t}_{j,i_k}=\frac{(r^{t}_{i_k})^2}{2\sqrt{(r^{t}_{i_k})^2-(HO^{t}_{j,i_k})^2}}-\frac{\left|r^{t}_{i_k}\right|}{2}. \quad (9)$$

The flow-chart of OIO is demonstrated in Figure 2.

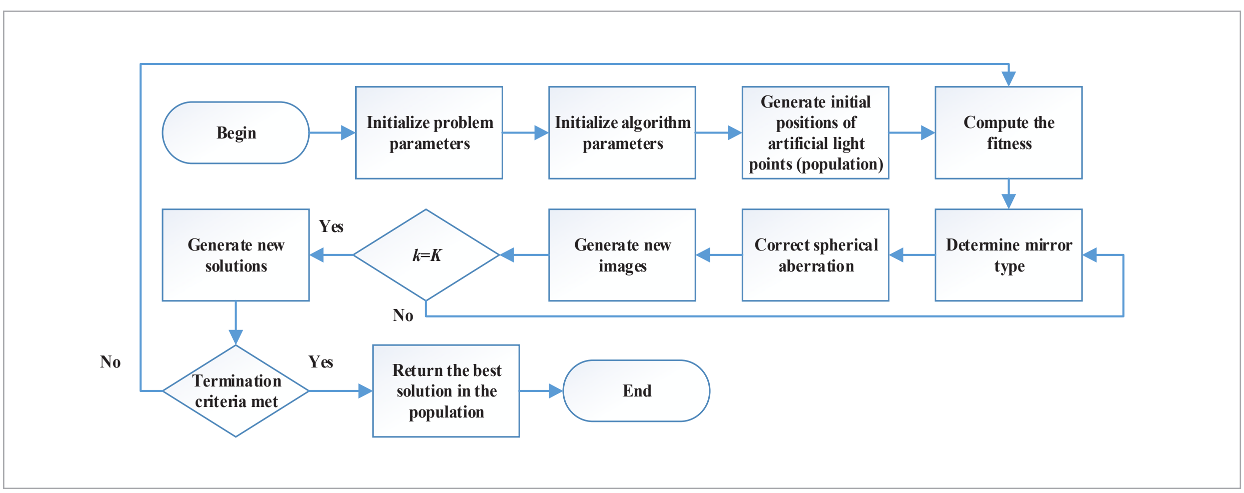
4. Travelling Tournament Problem

Travelling Tournament Problem (TTP) is one of the most popular scheduling problems in any sport. Where the team travels when traveling charts are created is an important issue. It is a problem that has arisen by imitating the traveling salesman problem [44, 20, 37].

There are professional leagues all over the world. They have great economic hedge because of the massive revenue generated by broadcast rights and ticket sales of popular matches. Therefore, the planning of these leagues is of great importance. Another important aspect is the creation of the time table for the tournament. During the season, which teams will match the other teams and the venues of the matches must be specified. Considering the distance between team numbers and locations, the time table of the Double Round Robin Tournament is used to reduce the total number of travel distances by teams [27, 12, 41].

The TTP is a problem of optimizing tournament schedule. Given N teams with N even, Double Round Robin Tournament is a game series in which every team plays every other team exactly once at home and once away. The game is designated as an ordered pair of competitors. $2\times(N\text{-}1)$ slots or time periods are required to play Double Round Robin Tournament. The distance between tool positions is given as $N\times N$ distance matrix. Each team starts in their own home and goes on a trip to play in selected locations. At the end

Figure 2
Flow-chart of OIO



of the program, each team will return to their home if necessary. Thus, the problem is to decide on the optimal plan and to make the average travel fee less.

Input: N (Number of teams), D (N×N distance matrix)

Output: Teams travel in the final tournament schedule with all restrictions and least total distance

The problem can be formalized according to the following rules:

1. N teams participate on the tour.
2. Every team has its own stadium.
3. Distances between stadiums are known.

The following constraints should be taken into account:

1. Each pair of teams (we can call them A and B) makes 2 matches. One of them is at A's and the other is at B's home. Thus, 2×(N-1) rounds are made and N/2 matches are performed in each round (N is the number of teams) (Double Round Robin restriction).
2. No team will be able to compete successive four times in their home or away on the road (Successive restriction).
3. For any pair of teams, for example, A and B, the match played at A's/B's home and the match played at B's/A's home cannot be on consecutive tours (non-repeating restriction). Thus, if the match is made at A's home this week, it cannot be done at B's home; matches should be played with other teams [44].

Table 3

The distance matrix for 6 × 6 TTP [11]

	ATL	NYM	PHI	MON	FLA	PIT
ATL	0	745	665	929	605	521
NYM	745	0	80	337	1090	315
PHI	665	80	0	380	1020	257
MON	929	337	380	0	1380	408
FLA	605	1090	1020	1380	0	1010
PIT	521	315	257	408	1010	0

ATL team will play against FLA (home), NYM (home), PIT (home), PHI (away), MON (away), PIT (away), PHI (home), MON (home), NYM (away), and FLA (away) based on the program in Table 4 [35]. Looking at the distance between the cities where the teams in Table 3 are located, it will be found how much the ATL team will move through the league [11]:

$$d_{ATL,PHI} + d_{PHI,MON} + d_{MON,PIT} + d_{PIT,ATL} + d_{ATL,NYM} + d_{NYM,FLA} + d_{FLA,ATL} = 665 + 380 + 408 + 521 + 745 + 1090 + 605 = 4414$$

The total distance in the league will be obtained by taking the sum of the distances according to the league program for each team. This total distance gives us the value of the objective function. As this distance decreases, the solution of the problem will also be considered as successful. Therefore, TTP is exactly the problem of minimizing the total distance in the league. The NL shown in Table 5 means nation-

Table 4

Sample league program for TTP (The @ sign represents a team playing away) [11]

Slot	ATL	NYM	PHI	MON	FLA	PIT
0	FLA	@PIT	@MON	PHI	@ATL	NYM
1	NYM	@ATL	FLA	@PIT	@PHI	MON
2	PIT	@FLA	MON	@PHI	NYM	@ATL
3	@ PHI	MON	ATL	@NYM	PIT	@FLA
4	@MON	FLA	@PIT	ATL	@NYM	PHI
5	@PIT	@PHI	NYM	FLA	@MON	ATL
6	PHI	@MON	@ATL	NYM	@PIT	FLA
7	MON	PIT	@FLA	@ATL	PHI	@NYM
8	@NYM	ATL	PIT	@FLA	MON	@PHI
9	@FLA	PHI	@NYM	PIT	ATL	@MON

Table 5
Solution comparison of TTP with the existing algorithms (The best values are expressed in boldface) [11]

Work	Method	NL4	NL6	NL8	NL10	NL12	NL14	NL16
[16, 40, 34, 16]	Linear Programming	**8276**	**23916**	41113				312623
[7]	A combination of constraint programming and lanrange relaxation	**8276**	**23916**	42517	68691	143655	301113	437273
[10]	Tabu Search	**8276**	**23916**	40416	66037	125803	205894	308413
[30]	Unknown (data from TTP Website)	**8276**	**24073**	39947	61608	119012	207075	293175
[43]	“Greedy big step” Meta-Heuristic			39776	61679	117888	206274	281660
[31]	Simulated Annealing and Hill-Climbing	**8276**	**23916**	**39721**	59821	115089	196363	274673
[30]	Unknown (data from TTP Website)				**59436**	112298	190056	272902
[13]	Ant Colony Optimization with Local Improvement	**8276**	**23916**	40797	67640	128909	238507	346530
[6]	Simulated Annealing	**8276**	**23916**	**39721**	59583	**111248**	**188728**	**263772**
[18]	Composite-Neighborhood Tabu Search Approach				59583	111483	190174	270063
[11]	Ant Algorithm Hyper-Heuristic	**8276**	**23916**	40361	65168	123752	225169	321037

al league and the *N* parameter indicates the number of teams. NL8 shows that the league consists of 8 teams. Several algorithms have been searched for TTP and the test results obtained are shown in Table 5.

5. Solution of the Travelling Tournament Problem with Optics Inspired Optimization

Assume that the number of teams is 8. By following the constraints defined in TTP, based on the distance matrix of the cities indicated in Table 6, the problem of minimizing the total movement of the teams through the league is achieved by OIO. League schedule for the first match is shown in Table 7. Table 9 shows the league fixture based on the weekly match schedule in Table 8. Twelve iterations are shown in Tables 10-21.

Table 6
Distances between stadiums (km)

Teams	1	2	3	4	5	6	7	8
1	0	50	60	80	120	182	160	220
2	50	0	75	92	90	137	170	196
3	60	75	0	100	110	89	150	77
4	80	92	100	0	25	94	152	86
5	120	90	110	25	0	75	215	69
6	182	137	89	94	75	0	50	60
7	160	170	150	152	215	50	0	40
8	220	196	77	86	69	60	40	0

Table 7
League schedule for the first match based on the number of teams

1. Week							
5	6	8	1	3	4	7	2

Table 8
Weekly league schedule showing the match of 8 teams for the first match

1. Week			
5	2	7	4
6	8	1	3

2. Week			
5	6	2	7
8	1	3	4

3. Week			
5	8	6	2
1	3	4	7

4. Week			
5	1	8	6
3	4	7	2

5. Week			
5	3	1	8
4	7	2	6

6. Week			
5	4	3	1
7	2	6	8

6. Week			
5	7	4	3
2	6	8	1

Table 9
League fixture

	Week	At home							At away						
		1	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Teams	1	7	6	5	4	2	8	3	7	6	5	4	2	8	3
	2	8	3	7	6	1	4	5	8	3	7	6	1	4	5
	3	4	2	8	5	7	6	1	4	2	8	5	7	6	1
	4	3	7	6	1	5	2	8	3	7	6	1	5	2	8
	5	**6**	**8**	**1**	**3**	**4**	**7**	**2**	**6**	**8**	**1**	**3**	**4**	**7**	**2**
	6	5	1	4	2	8	3	7	5	1	4	2	8	3	7
	7	1	4	2	8	3	5	6	1	4	2	8	3	5	6
	8	2	5	3	7	6	1	4	2	5	3	7	6	1	4

Movement path for 1st team: Team 1 will play with team 7 (home), 6 (home), 5 (home), 7 (away), 6 (away), 5 (away), 8 (home), 3 (home), 4 (away), 2 (away), 4 (home), 2 (home), 8 (away), 3 (away) according to the league fixture shown in Table 9.

Looking at the distances between the cities shown in Table 6, it can be found how far the 1st team will move through the league:

$d_{1,7} + d_{7,6} + d_{6,5} + d_{5,1} + d_{1,4} + d_{4,2} + d_{2,1} + d_{1,8} + d_{8,3} + d_{3,1} = 160 + 50 + 75 + 120 + 80 + 92 + 50 + 220 + 77 + 60 = 984.$

Similarly, for all other teams, the total mobility during the iteration is shown in Table 10. The league schedule and the total distance the teams have taken based on the first match shown in Table 7 are similarly calculated for the matches shown in Tables 11-21.

Table 10
Total distance the teams took in an iteration according to the first match

Total path taken by team 1: 984 Total: 984	Total path taken by team 2: 1169 Total: 2153
Total path taken by team 3: 1271 Total: 3424	Total path taken by team 4: 993 Total: 4417
Total path taken by team 5: 1185 Total: 5602	Total path taken by team 6: 1051 Total: 6653
Total path taken by team 7: 1109 Total: 7762	Total path taken by team 8: 1009 Total: 8771

Table 11
League schedule for the second match and the total distance the teams took in an iteration

2. Week							
4	1	7	6	2	3	5	8
Total path taken by team 1: 1003 Total: 1003				Total path taken by team 2: 938 Total: 1941			
Total path taken by team 3: 895 Total: 2836				Total path taken by team 4: 831 Total: 3667			
Total path taken by team 5 : 1010 Total: 4677				Total path taken by team 6: 952 Total: 5629			
Total path taken by team 7 : 1172 Total: 6801				Total path taken by team 8: 1094 Total: 7895			

Table 12
The league schedule for the third match and the total distance the teams took in an iteration

3. Week							
3	4	5	1	8	6	7	2
Total path taken by team 1: 1250 Total: 1250				Total path taken by team 2: 892 Total: 2142			
Total path taken by team 3: 926 Total: 3068				Total path taken by team 4: 921 Total: 3989			
Total path taken by team 5: 1080 Total: 5069				Total path taken by team 6: 979 Total: 6048			
Total path taken by team 7: 1375 Total: 7423				Total path taken by team 8: 1009 Total: 8432			

Table 13
The league schedule for the fourth match and the total distance the teams took in an iteration

4. Week							
4	3	6	1	8	7	5	2
Total path taken by team 1: 1252 Total: 1252				Total path taken by team 2: 955 Total: 2207			
Total path taken by team 3: 1047 Total: 3254				Total path taken by team 4: 936 Total: 4190			
Total path taken by team 5: 979 Total: 5169				Total path taken by team 6: 1041 Total: 6210			
Total path taken by team 7: 1118 Total: 7328				Total path taken by team 8: 944 Total: 8272			

Table 14
League schedule for the fifth match and the total distance the teams took in an iteration

5. Week							
5	6	8	1	3	4	7	2
Total path taken by team 1: 984 Total: 984				Total path taken by team 2: 1169 Total: 2153			
Total path taken by team 3: 1271 Total: 3424				Total path taken by team 4: 993 Total: 4417			
Total path taken by team 5: 1185 Total: 5602				Total path taken by team 6: 1051 Total: 6653			
Total path taken by team 7: 1109 Total: 7762				Total path taken by team 8: 1009 Total: 8771			

Table 15
League schedule for the sixth match and the total distance the teams took in an iteration

6. Week							
4	1	7	5	2	3	6	8
Total path taken by team 1: 1064 Total: 1064				Total path taken by team 2: 945 Total: 2009			
Total path taken by team 3: 956 Total: 2965				Total path taken by team 4: 987 Total: 3952			
Total path taken by team 5: 1093 Total: 5045				Total path taken by team 6: 1035 Total: 6080			
Total path taken by team 7: 1255 Total: 7335				Total path taken by team 8: 1079 Total: 8414			

The total distance (global minimum) of the teams in the 11th week was calculated as 7955 km, based on the results obtained in 1 iteration, i.e. 12 weeks, of the algorithm. When the algorithm is run for 100 iterations, the total distance obtained is 7372 km.

Generally, performances of computational intelligence optimization algorithms in terms of many metrics for a specific complex problem are compared within the same conditions. Due to the stochastic characteristics of the computational intelligence opti-

Table 16

League schedule for the seventh match and the total distance the teams took in an iteration

7. Week							
3	4	6	1	8	5	7	2
Total path taken by team 1: 1194 Total: 1194				Total path taken by team 2: 904 Total: 2098			
Total path taken by team 3: 1087 Total: 3185				Total path taken by team 4: 906 Total: 4091			
Total path taken by team 5: 878 Total: 4969				Total path taken by team 6: 1022 Total: 5991			
Total path taken by team 7: 1092 Total: 7083				Total path taken by team 8: 1009 Total: 8092			

Table 17

League schedule for the eighth match and the total distance the teams took in an iteration

8. Week							
4	3	6	1	8	7	5	2
Total path taken by team 1: 1252 Total: 1252				Total path taken by team 2: 955 Total: 2207			
Total path taken by team 3: 1047 Total: 3254				Total path taken by team 4: 936 Total: 4190			
Total path taken by team 5: 979 Total: 5169				Total path taken by team 6: 1041 Total: 6210			
Total path taken by team 7: 1118 Total: 7328				Total path taken by team 8: 944 Total: 8272			

Table 18

League schedule for the ninth match and the total distance the teams took in an iteration

9. Week							
4	6	8	1	3	5	7	2
Total path taken by team 1: 1001 Total: 1001				Total path taken by team 2: 1169 Total: 2170			
Total path taken by team 3: 1206 Total: 3376				Total path taken by team 4: 1103 Total: 4479			
Total path taken by team 5: 965 Total: 5444				Total path taken by team 6: 1051 Total: 6495			
Total path taken by team 7: 1103 Total: 7598				Total path taken by team 8: 1024 Total: 8622			

Table 19

The league schedule for the tenth match and the total distance the teams took in an iteration

10. Week							
5	1	4	7	2	3	6	8
Total path taken by team 1: 1300 Total: 1300				Total path taken by team 2: 1012 Total: 2312			
Total path taken by team 3: 1192 Total: 3504				Total path taken by team 4: 1095 Total: 4599			
Total path taken by team 5: 1046 Total: 5645				Total path taken by team 6: 1037 Total: 6682			
Total path taken by team 7: 1248 Total: 7930				Total path taken by team 8: 1051 Total: 8981			

mization algorithms, the performance of the algorithm can be understood by interpreting the results after at least 30 runs. In this study, League Championship Algorithm (LCA) has been selected for performance comparisons. OIO and LCA were run 30 times for both 1 (iteration, season) and 100 (iterations, seasons). For statistical analysis, t-test method has been performed.

In order to compare the performance of the OIO algorithm with the LCA, the same iteration parameters have been used. *LightPoint* value is selected as 12 in OIO algorithm. In LCA, the number of teams (*Team*) is 4. *LightPoint* is set to this value because there will be a total of 12 matches during the season, as it is 4 teams in the league and 3 weeks a season. *N* value

Table 20

The league schedule for the eleventh match and the total distance the teams took in an iteration

11. Week							
3	6	5	1	8	4	7	2
Total path taken by team 1: 990 Total: 990				Total path taken by team 2: 1005 Total: 1995			
Total path taken by team 3: 1002 Total: 2997				Total path taken by team 4: 717 Total: 3714			
Total path taken by team 5: 876 Total: 4590				Total path taken by team 6: 775 Total: 5365			
Total path taken by team 7: 1115 Total: 6480				Total path taken by team 8: 1066 Total: 7546			

Table 21

The league schedule for the twelfth match and the total distance the teams took in an iteration

12. Week							
4	3	5	1	8	7	6	2
Total path taken by team 1: 1240 Total: 1240				Total path taken by team 2: 955 Total: 2195			
Total path taken by team 3: 991 Total: 3186				Total path taken by team 4: 1011 Total: 4197			
Total path taken by team 5: 1106 Total: 5303				Total path taken by team 6: 1136 Total: 6439			
Total path taken by team 7: 1206 Total: 7645				Total path taken by team 8: 966 Total: 8611			

refers to the problem dimension and N was selected as 8 in both algorithms. In contrast to the number of seasons in LCA, the maximum iteration value in OIO algorithm is 100. LCA and OIO for 1 season or iteration and 100 seasons or iterations used in the total distance optimization in the TTP have been run and the performances of both optimization algorithms on this problem have been measured. The total minimum distance obtained is shown in Table 22.

Table 22

Comparison of OIO and LCA (km)

Number of Run	OIO (1 iteration)	LCA (1 season)	OIO (100 iterations)	LCA (100 seasons)
1	7955	8417	7372	7362
2	7897	8074	7409	7625
3	7886	8393	7415	7546
4	8024	8115	7480	7511
5	8288	8123	7406	7362
6	7682	8038	7429	7569
7	7597	8078	7394	7415
8	7704	8062	7459	7519
9	7730	7902	7362	7405
10	8169	7744	7385	7530
11	7985	7575	7516	7495
12	8042	8024	7478	7453
13	7716	7777	7505	7362
14	7937	7980	7338	7512
15	7966	8496	7295	7511
16	7581	8237	7449	7400
17	8040	8274	7530	7563
18	7858	8130	7496	7554
19	8177	8649	7394	7575
20	7973	8310	7540	7600
21	7849	7845	7453	7347
22	7516	7861	7386	7295
23	7850	7793	7400	7340
24	7933	7945	7340	7338
25	7878	8019	7482	7347
26	7507	7809	7358	7470
27	7842	7876	7491	7539
28	8036	8152	7360	7372
29	7794	8373	7603	7629
30	7899	7693	7347	7519

The t-test results obtained by running LCA and OIO algorithms for 1 season or iteration are shown in Table 23, and the t-test results obtained by running 100 seasons or iterations are shown in Table 24.

Table 23
t-test results obtained by 1 season-iteration operation of LCA and OIO algorithm

	LCA	*OIO*
Mean	8058.8	7877.033333
Variance	65023.06207	36074.65402
Observations	30	30
Hypothesized Mean	0	
t Stat	3.59930922	
P(T<=t) one-tail	0.000586497	
t Critical one-tail	1.699127027	
P(T<=t) two-tail	0.001172994	
t Critical two-tail	2.045229642	

Table 24
t-test results obtained by 100 season-iteration operations of LCA and OIO algorithm

	LCA	*OIO*
Mean	7482.233333	7429.066667
Variance	16078.25402	5199.512644
Observations	30	30
Hypothesized Mean	0	
t Stat	2.170686074	
P(T<=t) one-tail	0.019141499	
t Critical one-tail	1.699127027	
P(T<=t) two-tail	0.038282999	
t Critical two-tail	2.045229642	

In the t-test;

H_0: It is argued that there is no difference between the means.

H_a: It is argued that there is a meaningful difference between the means.

P: Probability value

Observations: Number of experiments

t Stat: *t* statistic value

Pearson Correlation: The correlation coefficient between LCA and OIO samples

t Critical one-tail: Single-sided *t* critical value

t Critical two-tail: Double-sided *t* critical value

alfa: Significant level

In the t-test, there are two hypotheses, H_0 and H_a. When the *P* value is less than 0.05, the H_0 hypothesis is rejected and H_a is accepted. When the *P* value is greater than or equal to 0.05, the H_0 hypothesis is accepted and H_a is rejected. H_0 hypothesis will be “The convergence value of OIO to global minimum is not lower than the convergence value of LCA to global minimum, there is no difference between the mean values of the obtained minimum values”. H_a hypothesis will be “The convergence value of OIO to the global minimum is lower than the convergence value of LCA to global minimum, meaning there is a significant difference between the means”.

According to the *P* values shown in Table 27, it is seen that single-ended 0.0005 and double-ended 0.001, both values being less than 0.05. The H_0 hypothesis is rejected and H_a is accepted. OIO algorithm has shown better results than LCA for 1 season-iteration. According to t-test results, OIO’s success has been found to be statistically better than LCA in 30 experiments. According to the *P* values shown in Table 28, it is seen that single-ended 0.01 and double-ended 0.03, both values being less than 0.05. The H_0 hypothesis is rejected and H_a is accepted. OIO algorithm has shown better results than LCA for 100 season-iteration. According to t-test results, OIO’s success has been found to be statistically better than LCA in 30 experiments.

Another experiment has been performed within real dataset NL8 obtained from [30] as shown in Table 25. The same parameters have been selected as used in synthetic dataset. Two algorithms have been executed 30 times. The obtained distances are listed in Table 26. OIO seems to perform better than LCA, however, the differences between mean values are not statistically important as evidenced by the t-test results shown in Table 27.

Table 25
Distances between stadiums (km)

		Teams							
		1	2	3	4	5	6	7	8
Teams	1	0	745	665	929	605	521	370	587
	2	745	0	80	337	1090	315	567	712
	3	665	80	0	380	1020	257	501	664
	4	929	337	380	0	1380	408	622	646
	5	605	1090	1020	1380	0	1010	957	1190
	6	521	315	257	408	1010	0	253	410
	7	370	567	501	622	957	253	0	250
	8	587	712	664	646	1190	410	250	0

Table 26
Comparison of LCA and OIO algorithms within NL8 dataset

Number of Run	OIO (100 iterations)	LCA (100 seasons)	Number of Run	OIO (100 iterations)	LCA (100 seasons)
1	45242	45357	16	45815	45460
2	45752	45242	17	45376	46452
3	45452	45242	18	45374	45242
4	45438	45399	19	45438	45564
5	45433	45815	20	45706	45452
6	45821	45242	21	45492	46009
7	45460	46002	22	45433	45242
8	45357	45677	23	45242	45857
9	45438	45357	24	45242	46002
10	45357	45752	25	45242	45522
11	45997	45633	26	45357	46673
12	45242	45564	27	45564	45564
13	45581	46275	28	45438	45433
14	45941	46002	29	45564	45242
15	45857	45492	30	45581	45374
			Mean	45507.73	45637.97

Table 27
t-test results of LCA and OIO within real NL8 dataset

	LCA	OIO
Mean	45507.73333	45637.96667
Variance	46565.71954	143230.723
Observations	30	30
Hypothesized Mean	0	
t Stat	-1.530079783	
P(T<=t) one-tail	0.068416633	
t Critical one-tail	1.699127027	
P(T<=t) two-tail	0.136833266	
t Critical two-tail	2.045229642	

6. Conclusions

Problems encountered in sports sciences such as traveling tournament problem, referee appointment problem, tournament planning, qualification and elimination problems, minimization of moving problems are difficult to be efficiently solved. Novel and computationally efficient methods should be searched for these search and optimization problems. OIO is one of the most recent physics-based algorithm that can be effectively used to solve such problems.

In this study, LCA and OIO were used for the first time to solve the Traveling Tournament Problem for which the total movement of the teams of the games has been aimed to be minimized considering the constraints. The results obtained from OIO are compared with the results obtained from LCA which is one of the most recent artificial intelligence based optimization algorithms.

In the formed artificial league, the total movement of all the teams obtained from OIO in the 1st season was 8417 km. When the algorithm's season number is set to 100, the total mobility has decreased to 7362 km. Similarly, the total movement of all the teams obtained from LCA in the 1st iteration was 7955 km. When the algorithm's iteration number is set to 100, the total mobility has decreased to 7372 km. These two new algorithms have been also used for a real dataset and OIO seems to perform better than LCA. Although OIO is very new, the obtained results are promising and OIO seems to be an alternative method for the complex search and optimization problems for which mathematical model cannot be created or takes too long for computing even if it is created. Distributed and parallel versions of OIO and LCA with optimized parameters can be proposed for many different complex problems.

Acknowledgments

We would like to present our thanks to anonymous reviewers and the esteemed editors for their helpful suggestions.

References

1. Akyol, S., Alatas, B. Automatic Mining of Accurate and Comprehensible Numerical Classification Rules with Cat Swarm Optimization Algorithm. Journal of the Faculty of Engineering and Architecture of Gazi University, 2016, 31(4), 839-857. https://doi.org/10.17341/gazimmfd.278440
2. Akyol, S., Alatas, B. Güncel Sürü Zekası Optimizasyon Algoritmaları. Nevsehir Journal of Science and Technology, 2012, 1(1) 36-50.
3. Akyol, S., Alatas, B. Plant Intelligence Based Metaheuristic Optimization Algorithms. Artificial Intelligence Review, 2017, 47(4), 417-462. https://doi.org/10.1007/s10462-016-9486-6
4. Alatas, B. ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization. Expert Systems with Applications, 2011, 38(10), 13170-13180. https://doi.org/10.1016/j.eswa.2011.04.126
5. Altay E. V., Alatas, B., Bird Swarm Algorithms with Chaotic Mapping. Artificial Intelligence Review, 2019. https://doi.org/10.1007/s10462-019-09704-9
6. Anagnostopoulos, A., Michel, L., Hentenryck, P., Vergados, Y. A Simulated Annealing Approach to the Travelling Tournament Problem. Journal of Scheduling, 2006, 9(2), 177-193. https://doi.org/10.1007/s10951-006-7187-8
7. Benoist, T., Laburthe, F., Rottembourg B. Lagrange Relaxation and Constraint Programming Collaborative Schemes for Travelling Tournament Problems. CP-AI-OR'2001,Wye College (Imperial College), Ashford, Kent UK, 2001.

8. Borji, A., Hamidi, M. A New Approach to Global Optimization Motivated by Parliamentary Political Competitions. International Journal of Innovative Computing. Information and Control, 2009, 5(6), 643-1653.
9. Can, U., Alatas, B. Physics Based Metaheuristic Algorithms for Global Optimization. American Journal of Information Science and Computer Engineering, 2015, 1(3), 94-106.
10. Cardemil, A. Optimizacion de Fixtures Deportivos: Estado del Arte y Un Algoritmo Tabu Search Para el Travelling Tournament Problem. Master thesis, Departamento de Computaci Ón Facultad de Ciencias Exactas y Naturales Universidad de Buenos Aires, 2002.
11. Chen, P., Kendall, G., Berghe, G. V. An Ant Based Hyper-Heuristic for the Travelling Tournament Troblem. Proceedings of IEEE Symposium of Computational Intelligence in Scheduling, 2007, 19-26. https://doi.org/10.1109/SCIS.2007.367665
12. Choubey, N. S. A Novel Encoding Scheme for Traveling Tournament Problem using Genetic Algorithm. IJCA Special Issue on Evolutionary Computation for Optimization Techniques, 2010, 79-82. https://doi.org/10.5120/1536-139
13. Crauwels, H., Oudheusden, D. V. Ant Colony Optimization and Local Improvement. The Third Workshop on Real-Life Applications of Metaheuristics, Antwerp, Belgium, 2003.
14. De Castro, L. N., Von Zuben, F. J. Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, 2002, 6(3), 239-251. https://doi.org/10.1109/TEVC.2002.1011539
15. Dorigo, M., Maniezzo, V., Colorni, A. The Ant System: An Autocatalytic Optimizing Process. Technical Report No. 91- 016, Dipartimento di Elettronica, Politecnico di Milano, 1991, Italy.
16. Easton, K., Nemhauser, G. L., Trick, M. A. Solving The Travelling Tournament Problem: A Combined Integer Programming and Constraint Programming Approach. 4th International Conference on the Practice and Theory of Automated Timetabling. Gent, Belgium, 2002, 319-330.
17. Easton, K., Nemhauser, G. L., Trick, M. A. The Travelling Tournament Problem: Description and Benchmarks. Principles and Practice of Constraint Programming. Springer, LNCS 2239, 2001, 580-585. https://doi.org/10.1007/3-540-45578-7_43
18. Gaspero, L. D., Schaerf, A. A Composite-Neighborhood Tabu Search Approach to The Travelling Tournament Problem. Journal of Heuristics, 2007, 13(2), 189-207. https://doi.org/10.1007/s10732-006-9007-x
19. Geem, Z. W., Kim, J. H., Loganathan, G. V. A New Heuristic Optimization Algorithm: Harmony Search. Simulation. 2001, 76, 60-68. https://doi.org/10.1177/003754970107600201
20. Gupta, D., Goel, L., Chopra, A. Enhanced Heuristic Approach for Travelling Tournament Problem Based on Extended Species Abundance Models of Biogeography. International Conference on Advances in Computing, Communications and Informatics, 2014, 1118-1124. https://doi.org/10.1109/ICACCI.2014.6968336
21. Holland, J. H. Adaption in Natural and Artificial Systems. University of Michigan Pres., 1975. Ann Arbor, MI.
22. Javidy, B., Hatamlou, A., Mirjalili, S. Ions Motion Algorithm for Solving Optimization Problems. Applied Soft Computing, 2015, 32, 72-79. https://doi.org/10.1016/j.asoc.2015.03.035
23. Kashan, A. H. A New Metaheuristic for Optimization: Optics Inspired Optimization (OIO). Computers & Operations Research, 2015, 55, 99-125. https://doi.org/10.1016/j.cor.2014.10.011
24. Kashan, A. H. An Effective Algorithm for Constrained Optimization based on Optics Inspired Optimization (OIO). Computer-Aided Design, 2015, 63, 52-71. https://doi.org/10.1016/j.cad.2014.12.007
25. Kashan, A. H. League Championship Algorithm (LCA): An Algorithm for Global Optimization inspired by Sport Championships. Applied Soft Computing, 2014, 16, 171-200. https://doi.org/10.1016/j.asoc.2013.12.005
26. Kennedy, J., Eberhart, R. C. Particle Swarm Optimization. IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, 1942-1948.
27. Kim, B. M. Iterated Local Search for the Traveling Tournament Problem. Master Thesis, 2012, Austria.
28. Kiziloluk, S., Alatas, B. Current Social-Based Heuristic Optimization Algorithms. Cumhuriyet University Journal of Economics and Administrative Sciences, 2012, 13(2), 39-56.
29. Lalwani, P., Banka, H., Kumar, C. CRWO: Clustering and Routing in Wireless Sensor Networks using Optics Inspired Optimization. Peer-to-Peer Networking and Applications, 2017, 10(3), 453-471. https://doi.org/10.1007/s12083-016-0531-7
30. Langford, Challenging Travelling Tournament Instances. http://mat.gsia.cmu.edu/TOURN/, Accessed on February 01, 2018.
31. Lim, A., Rodrigues, B., Zhang, X. A Simulated Annealing and Hill-Climbing Algorithm for the Traveling Tournament Problem. European Journal of Opera-

tional Research, 2006, 174(3), 1459-1478. https://doi.org/10.1016/j.ejor.2005.02.065

32. Meng, X., Liu, Y., Gao, X., Zhang, H. A New Bio-Inspired Algorithm: Chicken Swarm Optimization. International Conference in Swarm Intelligence, Springer International Publishing, 2014, 86-95. https://doi.org/10.1007/978-3-319-11857-4_10
33. Mora-Gutiérrez, R. A., Ramírez-Rodríguez, J., Rincón-García, E. A. An Optimization Algorithm inspired by Musical Composition. Artificial Intelligence Review, 2014, 41(3), 301-315. https://doi.org/10.1007/s10462-011-9309-8
34. Nemhauser, G. L., Trick, M. A. Scheduling a Major College Basketball Conference. Operations Research, 1998, 46(1), 1-8. https://doi.org/10.1287/opre.46.1.1
35. Ozbay, F. A., Alatas, B. Review of Computational Intelligence Method Inspired from Behavior of Water. Afyon Kocatepe University Journal of Science and Engineering, 2016, 16, 137-147.
36. Ozbay, F. A., Alatas, B. Review of Social-Based Artificial Intelligence Optimization Algorithms for Social Network Analysis. International Journal of Pure and Applied Sciences, 2015, 1, 33-52.
37. Pérez-Cáceres, L., Riff, M. C. Solving Scheduling Tournament Problems Using a New Version of CLONALG. Connection Science, 2015, 27(1), 5-21. https://doi.org/10.1080/09540091.2014.944099
38. Ramezani, F., Lotfi, S. Social-Based Algorithm (SBA). Applied Soft Computing, 2013, 13(5), 2837-2856. https://doi.org/10.1016/j.asoc.2012.05.018
39. Rao, R. V. Savsani, V. J., Vakharia, D. P. Teaching-Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large Scale Problems. Information Sciences, 2012, 183(1), 1-15. https://doi.org/10.1016/j.ins.2011.08.006
40. Russell, R. A., Leung J. M. Devising a Cost Effective Schedule for a Baseball League. Operations Research, 1994, 42(4), 614-625. https://doi.org/10.1287/opre.42.4.614
41. Ryckbosch, F., Berghe, G. V., Kendall, G. A Heuristic Approach for the Travelling Tournament Problem Using Optimal Travelling Salesman Tours. Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, 2008.
42. Salem, S. A. BOA: A Novel Optimization Algorithm. IEEE International Conference on Engineering and Technology, 2012, 1-5. https://doi.org/10.1109/ICEngTechnol.2012.6396156
43. Shen, H., Zhang, H. Greedy Big Steps As a Meta-Heuristic for Combinatorial Search. The University of Iowa AR Reading Group, Spring 2004 Readings.
44. Tamgave, P., Jain, S., Waghmode, G., Udagire, S., Umale, J. Optimization of Travelling Tournament Problem using Nature Based Algorithms. International Journal of Innovative Research in Science, Engineering and Technology, 2014, 3(2), 9395-9402.
45. Tanyildizi, E., Demir, G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm. Advances in Electrical and Computer Engineering, 2017, 17(2), 71-78. https://doi.org/10.4316/AECE.2017.02010
46. Tayarani-N, M. H., Akbarzadeh-T, M. R. Magnetic Optimization Algorithms a New Synthesis. IEEE Congress on Evolutionary Computation, 2008, 2659-2664. https://doi.org/10.1109/CEC.2008.4631155
47. Xing, B., Gao, W. J. Central Force Optimization Algorithm. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Springer International Publishing, 2014, 333-337. https://doi.org/10.1007/978-3-319-03404-1_19