Indonesian Journal of Electrical Engineering and Computer Science
Vol. 22, No. 1, April 2021, pp. 542~551
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v22.i1.pp542-551

Organizing sports matches with a hybrid monkey search algorithm

Ruqaya Zedan Shaban[1], Isra Natheer Alkallak[2]
[1]Computer unit, College of Medicine, University of Mosul, Mosul, Iraq
[2]Department Basic Sciences, College of Nursing, University of Mosul, Mosul, Iraq

Article Info

Article history:

Received Oct 15, 2020
Revised Jan 15, 2021
Accepted Jan 29, 2021

Keywords:

First keyword
Second keyword
Third keyword
Fourth keyword
Fifth keyword

ABSTRACT

This academic piece of the paper illustrates an approach of organizing sports matches to be as a tool of sports schedule maker via a proposed hybrid Monkey search algorithm by doubles elimination tournament. Input are a set of teams represented by monkeys. It states that the natural behavior of monkeys and its climbing to obtain (solution) through uphill (climbing), watch-jumping, and somersault process. The parameters in the climb were represented by whether or not sports team won. The distribution of the teams, whether in the upper or the lower half, represented in the watch-jumping by mathematical equations, in addition to representing the variables in somersault process. The consequences behind the suggested algorithm used have tackled steps to reach an optimal schedule for a variety of teams for the same championship. The team cannot play more than two consecutive matches with the condition of starting with a random population of feasible solutions. The performance of the presented algorithm was very well through providing a flexible and easy organization of sports matches without using classic or manual routine. Furthermore, the search space reduced so that can give a guarantee to achieve a globally optimum solution. The software platform was MATLAB m-files.

Corresponding Author:

Ruqaya Zedan Shaba
Computer unit, College of Medicine
University of Mosul, Mosul, Iraq.
Email: rzs@uomosul.edu.iq

1. INTRODUCTION

The term "monkey search algorithm" term was first coined via Mucherino and Seref in 2007. In detail, it is a metaheuristic for global optimization and type of swarm intelligence algorithms. This route was inspired by the simulation of mountain-climbing processes in its search for food. When the monkey starts climbing a different new mountain for the first time, it can only arbitrarily select the mountain's paths, as it has no previous mountain experience. Monkey search incorporates principles and techniques from other meta-heuristic approaches such as the genetic algorithm approach and the colony of ant. The mechanisms of the Monkey hunt algorithm are summarized in find solution, fitness function identification, initialization process, climbing procedure, watch-jump practice, somersault process, and finalization or termination [1-7]. The initial games to be played and the rules for pairing the winners in the subsequent games define tournaments. The number of rounds in a tournament is the maximum number of matches that must be played by each of the teams to win the tournament. A bye occurs around when a team does not play. A classic tournament is a tournament where a power of two is the number of teams and no byes are given. There are two brackets in double-elimination tournaments the winning bracket where the competition continues similarly as in a bracket and the losing bracket where the bracket losers are mapped into a new knockout tournament. The loser's bracket tournament is held after (or concurrently) with the end of the winners' bracket. The losers 'bracket champion faces (possibly again) the winners' bracket champion (the bracket winner) the winner of this final match is the overall tournament winner in a deterministic setting. One of the ways to organize sports matches is a double loss of the team. The double loss of the team is characterized by the fact that the joint team is forced to play until it loses twice [8-10]. Figure 1 illustrated the tournament for 4 teams.

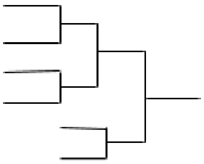


Figure 1. Example of tournament for 4 teams

The intention behind this research was to recommend a Monkey fusion search algorithm to organize sports matches. This hybridization algorithm relies on the search algorithm for Monkey with the suggested steps to arrange sports matches. Hybridization is known as a phase of development in optimization techniques in which machinists from a certain algorithm are shared with other proposed steps to efficiently produce the optimal solution and achieve better results. Our search space in the proposed algorithm is n! (factorial). The number n represents the number of sports teams or the number of monkeys. This will result in a large search space because each team has several positions in series with other teams. This gives a broad possibility of one team position’s odds relative to other team’s positions.

This work is structured as follows: Section 2 introduces artificial intelligence with meta-heuristic, monkey search algorithms, and sports matches. The experimental set-up, solution representation, and overview of the proposed algorithm are shown in Section 3. Section 4 addressed the proposed algorithm with the flow chart and the numerical experimental results. Section 5 outlines the results of these conclusions.

2. ARTIFICIAL INTELLIGENCE WITH META-HEURISTIC

In the first place, Fred Glover [11] proposed the word ' meta-heuristic.' It is called the overall strategic approach at the highest level that can direct other heuristics in searching for reasonable solutions in domains, particularly when the task is classified as complex work. One of the best methods for "good enough/fast enough/cheap enough" solutions is meta-heuristic [12].

Meta-heuristic algorithms are methods that are used to solve combination problems to achieve optimal/sub-optimal solutions that decrease errors. Given these types of algorithms (meta-heuristic) being easy to solve combinatorial optimization problems and finding local/global solutions, in some cases there is no guarantee of achieving an optimal solution. One of these algorithms is the Tabu search process that has the advantage of avoiding a return to the previous solution in each iteration. Alkallak and Sha'ban highlighted a structure or methodology to solve the Traveling salesman problem with the help of the Tabu search algorithm [13, 14]. One of the meta-heuristic algorithms is an optimization of the Ant colony [15]. Alkallak [16] proposed a hybrid algorithm to interfere with assignment problems of the Hungarian method with Ant colony optimization to solve the assignment problem. The simulated annealing is a global heuristic technique, that aims to avoid falling into local optimal by accepting unwanted results when specific function conditions are satisfied. Alkallak & Sha'ban succeeded in producing several possible solutions through the Spanning tree problem algorithm with simulated annealing in an undirected tree graph [17]. However, the Cuckoo search algorithm is inspired by meta-heuristic to solve other problems of combinatorial optimization as the N-Queens problem via Alkallak's proposed hybridization algorithm [18, 19]. Mosquito swarm algorithm is another meta-heuristics algorithm. To solve one of the problems of combinatorial optimization, it is a bio-inspired algorithm. Mosquito swarms algorithm was a fortunate knapsack problem solving by Sha'ban & Alkallak [20]. They adapted steps of the Mosquito swarms algorithm depend on the behavior of the mosquito. The role of the hybrid is clear in solving optimization problems such as the Knight tour problem by the Sudoku grid [21]. Simultaneously, the researcher Alkallak pointed to the role of the Magic square, especially of odd order as three to construct the Sudoku grid [22]. Certain meta-heuristic forms are a genetic algorithm to solve problems of optimization as a problem of scheduling. Alkallak & Sha'ban addressed the genetic algorithm in scheduling independent tasks for multiprocessors by proposed heuristic Genetic algorithm [23] and with dependent tasks for multiprocessors [24].

2.1. Monkey search algorithm

The monkey search algorithm is a nature-inspired optimization scheme. It can be considered as one of the recent swarm intellect algorithms which simulate the behavior of monkeys. It impersonators the process in which monkeys climb mountains to reach the uppermost point. The climbing method consists of three processes: climbing route, watch-jump +practice, and somersault process. What occurs in the climbing process, the monkey explores the search field or space, this means that the monkey searches about the local group's finest solution. While the second process which is watch-jump can allow the monkey to search for new different solutions with goal values that will be higher than the present ones. Then, the somersault process begins with the monkey search for new points in other search domains to discover and reach better solutions and to avoid getting trapped at local optimal. Monkeys can somersault across a certain pivot, the pivot is created according to a center of mass of present places coordinates of all monkeys around. Finally, termination condition getting started, the Monkey search algorithm continues running till there is no further development in the cost of the fitness function is acquired [1-7].

2.2. Sport matches

A tournament is a rule defining how to compare the teams to determine a champion. To win a double-elimination tournament, a team must either win all of its games in the winning tournament or win all of its remaining games after losing one game. The double-elimination tournament appeared by winning the winner's tournament and beating the loser's tournament champion, or winning the winner's tournament, losing to the loser's tournament champion, and then beating the loser's tournament champion, or winning the loser's tournament and beating the winner's tournament champion twice [8-10]. The method of double loss of the team is characterized by the team is allowed to play until losing twice. Some teams play and others wait. The total number of matches equals twice the number of teams minus one. The number of teams that will be waiting for matches equals to next power of two of the number of teams minus the number of teams. The number of teams that will be playing matches equals the number of teams minus waiting matches. If the number of waiting teams is even, then they shall be distributed equally. If the number of waiting teams is odd, then the teams distributed as follows:

a) In the Upper half, $(the\ number\ of\ the\ team\ waiting\ for\ +\ 1)\ /2$.
b) In the lower half, $(the\ number\ team\ waiting\ -\ 1)\ /2$ [25].

3. EXPERIMENTAL SETUP

The search has some properties or attributes matching the monkey search algorithm similar to the climbing process, the team discovers the search space, which means the team searches the local optimum solution and each team has a level. This level interprets the position of the team. In the watch-jump process, the team looks for alternative solutions with a new objective value higher than the obtained solutions through two candidate teams randomly selected to play the match by using random function. In the somersault process, the team attempt to find new points in other domains to achieve healthier solutions and avoid being trapped at local optimal. Teams can somersault around a pivot which is generated a based number of waiting matches of all teams. In the termination condition, the proposed algorithm rules will be continuously applied without stopping till no obtained improvement in the value of the cost of fitness function this means that we have the winning team. The research initializes the parameters as follows:

a) The number of monkeys (the number of teams) denoted by M, (number of teams arbitrary).
b) Population size and the number of positions for all (teams) denoted by M!
c) the number total matches or number of mountain equivalent number climb denoted by NC as in the calculation:$NC = 2 * (M - 1)$
d) At the most number of matches for each team equivalent step length by:
e) $1 \leq a \leq (NC/2)$
f) Number of positions for each team denoted by M!/M as a vector labeled $x_i = (x_{i1}\ , x_{i2}, \dots \dots \dots x_{iM})$
   - The position will be employed to represent a solution to the problem.
g) The number of teams that will be waiting matches denoted by Wt, as Wt = next power of two of the number of teams - M.
h) The number of teams that will be playing matches denoted by Pt= M- Wt.
i) Record the local solution for the winning team with the expected teams. Record the losing team with the losers.

3.1. Representation of solution and description the proposed algorithm

In this research, to represent the Monkey search algorithm to organize sports matches by describing the components of the proposed algorithm in details as follows:

3.1.1. Initial population

Initialization of the proposed algorithm possible solutions are generated randomly. The number of monkeys is (number of teams). The number of mountains is the total number of sports matches. However, sports match problem, have $M!$ cases of possibilities. Each team has $M!\ /\ M$ cases of possibilities because each team has several positions, where each team does not know where the position within the teams. So, for the team, we randomly choose the samples. These positions are different. Also, at this initialization, for each step, two innovative solutions are produced that one team selects to own a sports match or not.

3.1.2. Climb process

Climbing is running gradually to vary the monkey's position (team's position) differs from the current to new ones to improve the objective function. In this process, we used a vector to assign the level or position for each team. In the beginning, the level is zero for each team as [0, 0, .0]. Furthermore, when one team wins the sports match, the level of the winning team will increase by one. Otherwise, the level of the losing team will be reduced. Resulting in this will be an adjustment in the objective function as well as the team's position will be changed. To choose any team to play a random vector will be generated. To determine Δx_{ij} by:

$$\Delta x_{ij} = \begin{Bmatrix} +1 & ; If\ a\ team\ wins \\ -1 & ; otherwise \end{Bmatrix}$$

In the vector $x_i = (x_{i1}\ , x_{i2}, \dots \dots . . x_{iM})$ the level contained positive value, or negative value, or zero. The step length of the climbing procedure is calculated by the number of matches for each team, at most, step length or parameter a = (number of matches for each team) /2.

3.1.3. Watch-jump process

The winner looks at whether there is another team for a sports match. If the answer yes, it will be jumping there from the position that monkey being there and then repeat the climbing process until it grasps the end of the sports matches. The new solution progresses in the objective function. If we have the number of waiting teams, then the teams are distributed depending on the eyesight or parameter b of the watch-jump process. If the number of waiting teams is even, then the number of teams in the upper half and lower half equaled. Otherwise, if the number of waiting teams is odd, Therefore, the value of eyesight is determined by +1 or -1 in the below equations that belong to locate the upper or lower team then as follows:

a) Number of teams in upper half $= (Wt + 1)\ /\ 2$

b) Number of teams in lower half $= (Wt - 1)\ /\ 2$

3.1.4. Somersault process

As a result of repetitions of both climbing and the watch-jumping processes, each team can find its local solution. In somersault to find out new search domain. In this research, sports matches with double dotting as the loser being eliminated twice. We have the losing teams, those teams will do a somersaulting process along around the pivot direction, which is equally as follows:

$$Pivot\ =\ a$$

In the general Monkey, the search algorithm contains the somersault interval as [c..d]. In this research, if c equaled 2, then the team is canceled because the team loosed twice. Moreover, the somersault interval will be valued as [1..d]. The interval [1..d] represented for each losing team. If the team had a level as 2, then the team out of the challenge, because sports matches with double dotting as the loser is eliminated twice. Here, the parameter d value equals the number of waiting teams plus the number of teams to be playing. We represent it as follows:

$c\ =\ 2$; For each team lose twice.

$d = wt + Pivot$

3.1.5. Termination condition

In the termination state, the proposed algorithm continues running until the winning team is obtained.

4. PROPOSED ALGORITHM

Below, the hybrid proposed Monkey search algorithm to organize sports matches as follows:

```
Part I: Initialization
Step1: Initialization for the number of monkeys (the number of teams) denoted by M.
       Calculate population size is M!, by $M! = M * (M-1) * (M-2) * \ldots..1$.
       the number of teams arbitrary
Step2: Compute the total number of sports matches or $NC = 2 * (M-1)$.
       $Wt$ = the number multiplier two closest to the number of teams - M.
       $Pt = M - Wt$; the number of teams to be playing.
Step3: Assign level to each team. At first, the level is zero, through vector
       $x_{ij} = (x_{i1}, x_{i2}, \ldots x_{in})$.
Part II: Climb process
Step4: While $(NC)$ DO
Step5: $Parameter\ a = Pt/2$
       Here, at the beginning we have a matrix called flag, to represent the status of each
       team of the number of times lost whose values are zeros. Matrix called x contains
       winning teams in each climb. Matrix called y contain losing teams in each climb.
Step6: Generate a new sports match randomly from two teams.
Step7: Update the level of the winning team by +1 in Vector $x_{ij} = (x_{i1}, x_{i2}, \ldots x_{iNC})$.
       Also, update the level of losing team by -1 in Vector $_{ij} = (x_{i1}, x_{i2}, \ldots x_{iNC})$. Record the local
       solution for the winning team in x and record the losing team with the losers in y with an update flag.
Part III: Watch-Jump Process
Step8: IF $Wt >= 1$ Then calculate eyesight (parameter b) as follows:
       IF $Wt\ is\ odd$ then b has
           upper half $= (Wt + 1) / 2$
           lower half $= (Wt - 1) / 2$
             Else   $b = Wt/2$
             End
           Else
         End.
Part IV: Climb process
Part V: Somersault Process
Step9:  Calculate Pivot⟶ $Pivot = a$
Step10: IF $Wt <> 0$ then $d = Wt + Pivot$
Step11: For i=1:d
Step12:   $c =$ Assign the level of losing team in Vector $x_{ij} = (x_{i1}, x_{i2}, \ldots x_{iPivot})$.
Step13: IF $c = 2$ Then the team is bye.
Step14: Else
Step15: Generate a new sports match by randomly of two teams,
Step16: $M$ =the number of teams in matrix $x$, update $Wt, Pt$.
Step17: GoTo Part II
Step18: End. End. End
Step19: Print the winning team.
```

4.1. Flow chart for proposed algorithm

Flowchart of hybrid proposed monkey search algorithm for organizing sports matches as shown in Figure 2.

4.2. Experimental Result

Below example illustrated hybrid proposed Monkey search algorithm to organize sports matches with 7 monkeys (7 teams) as follows:

Part I Initialization

Number of monkeys (number of teams) $M = 7$. Population size is *=(5040). NC =12* . Step length or parameter $a = 6$. For Monkey1 (team1) the positions may be one of these positions $(x11, x12, x13, x14, x15, x16, x17)$. each monkey (team) had 7 positions. Below, Figure 3 contains positions for each team. We generated randomly the positions. In this Figure, We assumed sequence number 1 to represent the position of each team as $M1\ M2\ M3\ M4\ M5\ M6\ M7$, from Figure 3 to Figure 11 illustrated the steps of the solution graphically. $Wt = 1, Pt = 6$, Eyesight (b) as follows:

The number of teams in the $upper\ half = 1$, Number of teams in the $lower\ half = 0$.

Part II Climb Process

Figure 4 illustrated the climbing process by $a = 6, M2$ with $M3$ will be played, $M4$ with $M5$ will be played, $M6$ with $M7$ will be played while $M1$ is waiting.

Part III Watch–jump Process

Counter of winner =3, Counter of losing =3. The number of matches is 3. Here, $if\ 3 < 12$ then execute Watch–jump Process. While the losing team, $M = 3, Wt = 1$, and $Pt = 2$. As for Eyesight (b) an equal number of teams in the $upper\ half\ \ = 1$, number of teams in $lower\ half\ \ 0$.

Generate new sports match randomly of two teams.

Part IV: Climb process

The climb repeated to prepare the somersault. Figure 5 illustrated the climbing process through $Wt = 1, Pivot = 2$, and the total number of matched = 4.

Part V: Somersault Process

Figure 6 illustrated the somersault process through assign the level of losing team in Vector $x_{ij} = (x_{i1}, x_{i2}, \ldots\ x_{iPivot})$ to c. Generate new sports matches by randomly of two teams in [1..d]. Counter of winner =1, Counter of losing =1, and a total number of teams played=2. After that execute somersault process the M4 and M7 are (bye) because the counter of monkey loser = 2, the total number of matched=5. Here, the number of matches is 5. If 5 <12 then execute from the beginning. Figure 7 illustrated the steps to be completed through M=4, Wt=0, Pt=4, and a=2. After M1 with M6 are the winner. The total number of matched =7. After that execute somersault process the M2 and M5 are (bye) because the counter of monkey loser = 2, the total number of matched = 9. Here, the number of matches is 9. If 9 <12 then <12 then execute from the beginning. Also, Figure 8 illustrated the steps to be completed through M=2, Wt=0, Pt=2, and a=1. After M1 is the winner the total number of matched=10. While Figure 9 illustrated the steps after that execute somersault process then M6 is (bye) because the counter of monkey loser = 2, the total number of matched =11 this can be illustrated by Figure 10. Here, the number of matches is 11. If 11 <12 then <12 then execute from the beginning. To continue we note that Figure 11 showed that M=2, Wt=0, Pt=2, and a=1. After that execute the somersault process then M3 is (bye) because the counter of monkey loser = 2, the total number of matched=12 this can be illustrated by Figure 11. Finally, the number of matched NC=12 then stop the algorithm, and the winning Monkey or winning team is M1.

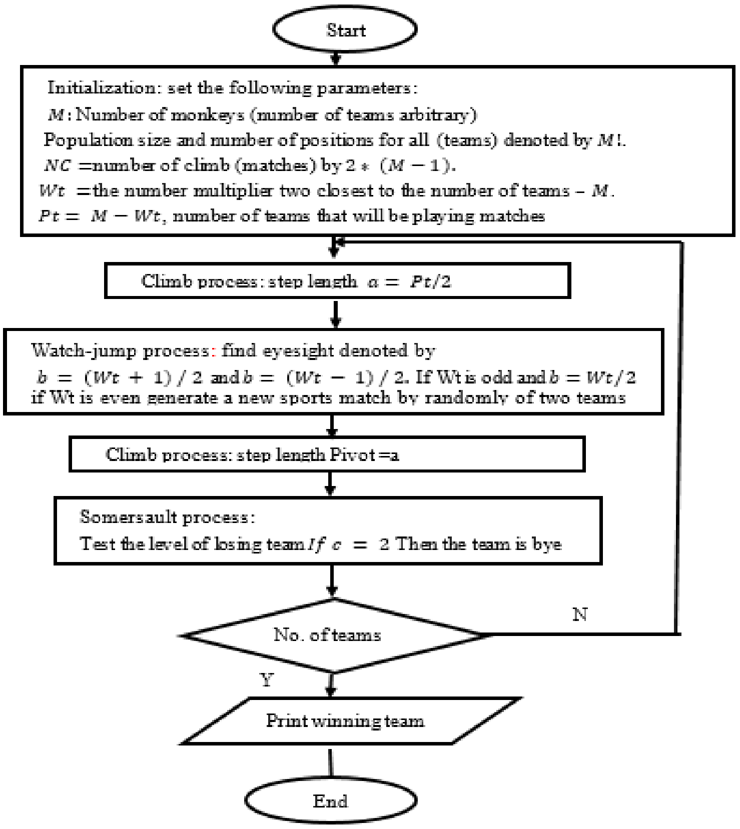
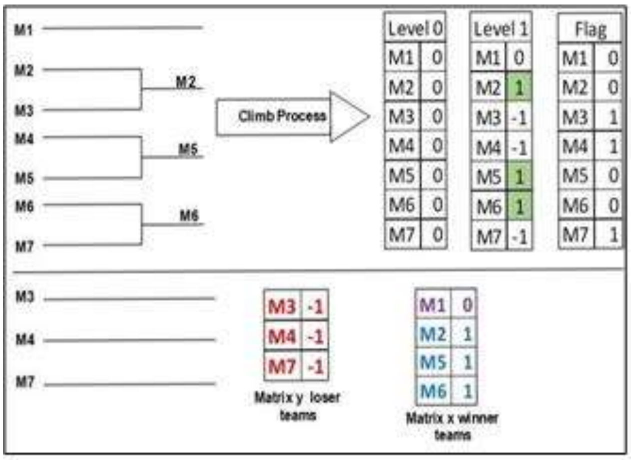

Start
Initialization: set the following parameters:
M: Number of monkeys (number of teams arbitrary)
Population size and number of positions for all (teams) denoted by $M!$.
NC =number of climb (matches) by $2 * (M - 1)$.
Wt =the number multiplier two closest to the number of teams – M.
$Pt = M - Wt$, number of teams that will be playing matches
Climb process: step length $a = Pt/2$
Watch-jump process: find eyesight denoted by
$b = (Wt + 1) / 2$ and $b = (Wt - 1) / 2$. If Wt is odd and $b = Wt/2$
if Wt is even generate a new sports match by randomly of two teams
Climb process: step length Pivot =a
Somersault process:
Test the level of losing team $If\ c = 2$ Then the team is bye
No. of teams
N
Y
Print winning team
End

Figure 2. Flowchart of hybrid proposed monkey search algorithm for organizing sports matches

	population						
Seq.	X1	X2	X3	X4	X5	X6	X7
1	M1	M2	M3	M4	M5	M6	M7
2	M1	M2	M3	M4	M5	M7	M6
3	M1	M2	M3	M4	M6	M5	M7
4	M1	M2	M3	M4	M6	M7	M5
5	M1	M2	M3	M5	M4	M6	M7
6	M1	M2	M3	M5	M4	M7	M6
7	M1	M2	M3	M5	M6	M4	M7
8	M1	M2	M3	M5	M6	M7	M4
9	M1	M2	M3	M5	M7	M4	M6
:	:	:	:	:	:	:	:
:	;	;	;	;	;	;	;
:	:	:	:	:	:	:	:
5040	M7	M6	M5	M4	M3	M2	M1

Figure 3. The population



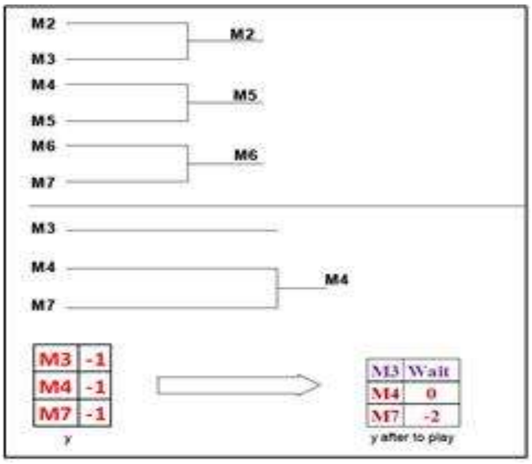
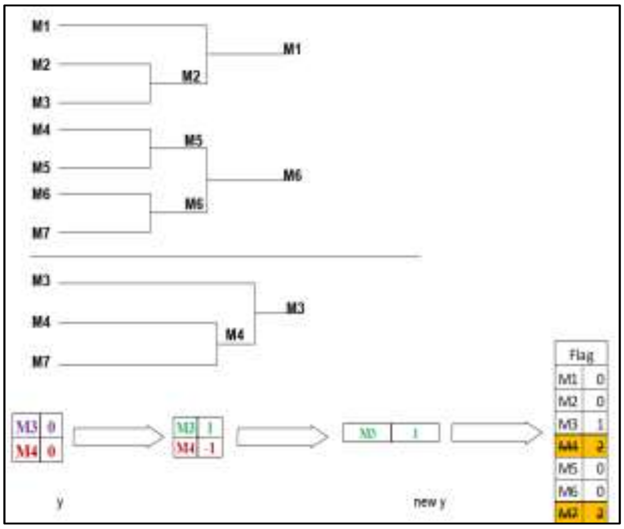


Figure 4. Illustrated the climbing process by $a = 6, M2$ with $M3$ will be played, $M4$ with $M5$ will be played, $M6$ with $M7$ will be played while $M1$ is waiting

Figure 5. Illustrated the climbing process through $Wt = 1, Pivot = 2$, and the total number of matched = 4



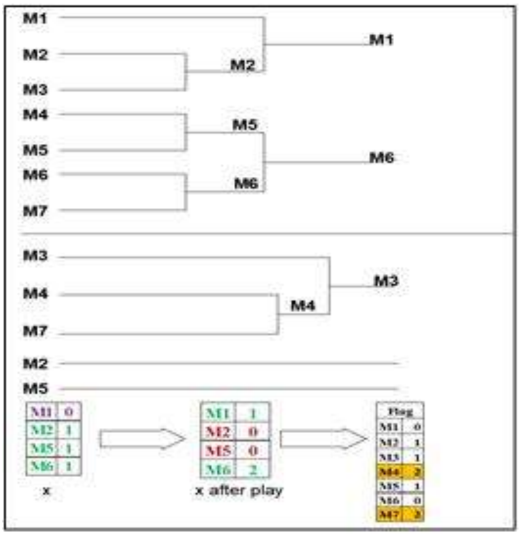


Figure 6. illustrated the somersault process through assign the level of losing team

Figure 7. illustrated the steps to be completed

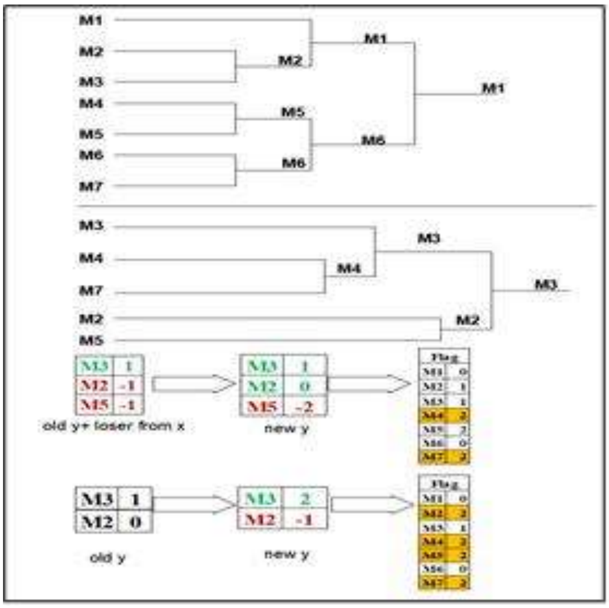


Figure 8. illustrated the steps to be completed through M=2, Wt=0, Pt=2, and a=1

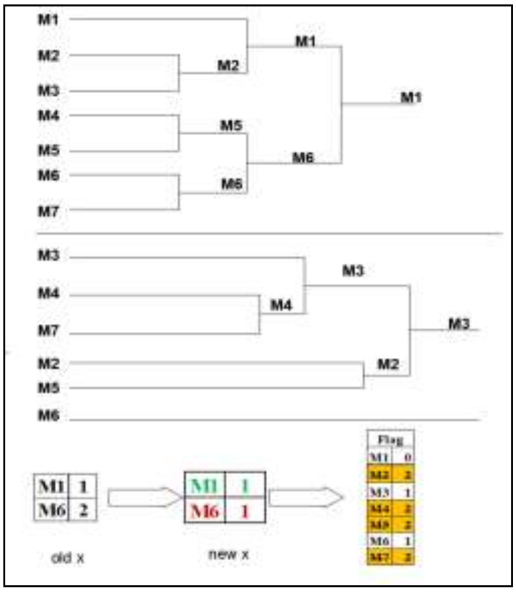


Figure 9. illustrated the steps after that execute somersault process

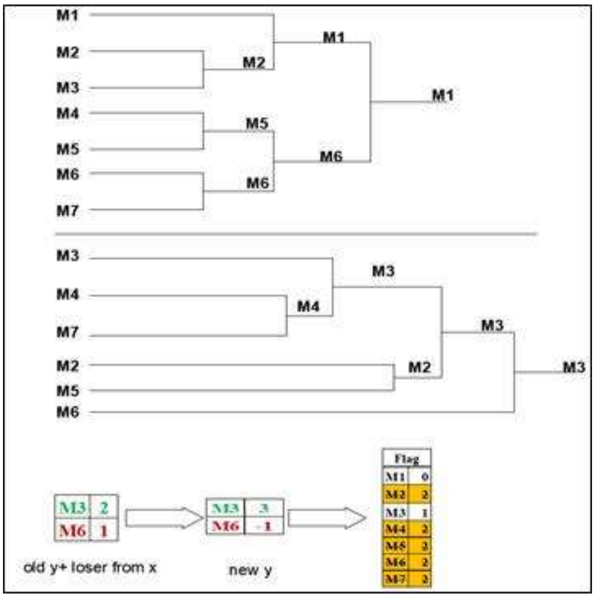


Figure 10. Illustrated total number of matched=11

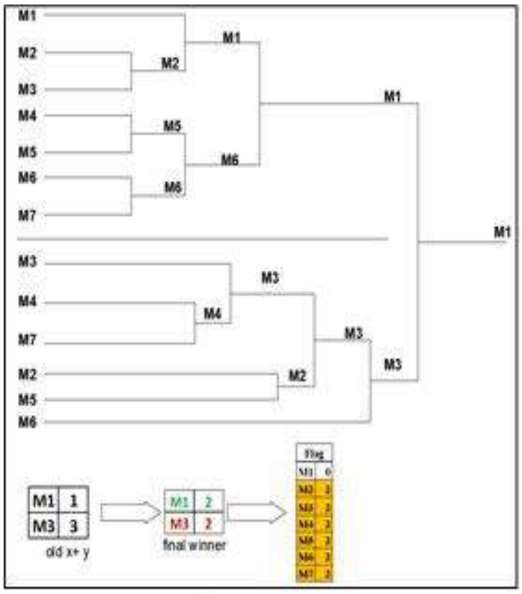


Figure 11. Illustrated M3 is (bye) because the counter of monkey loser = 2, the total number of matched=12

The best Monkey search algorithm for some algorithms for easy implementation of its steps. The variables of the Monkey search algorithm are easily applied to any problem to be solved, while other algorithms have many complications. We chose an algorithm for the monkey to search and climb trees or mountains similar to the issue of sports matches, including the position of the team and calculating whether or not it won. In our research, we chose the monkey search algorithm and its ease in the matter of sports matches compared to the classic lengthy manual methods.

5. CONCLUSION

This research tackled organizing sports matches with an optimization algorithm to schedule a double-elimination tournament via a proposed hybrid Monkey search algorithm and preparing schedules for sports teams through a hybrid monkey search algorithm. The input data for the proposed algorithm is a set of teams represented by monkeys. The algorithm states that the natural behavior of monkeys and its climbing to obtain (solution) through uphill (climbing), watch-jumping, and somersault process. The parameters in the climbing process were represented by whether or not the sports team won. The distribution of the teams, whether in the upper or the lower half, was represented in the watch-jumping process by mathematical equations, in addition to representing the variables in the process of somersault. The consequences behind the suggested procedure have tackled several steps to reach an optimal schedule for a variety of teams for the same championship by software that is created to simulate the monkey hybrid algorithm to obtain the solution for easy. The performance of the presented algorithm was very well through providing a flexible and easy organization of sports matches without using classic or manual routine. Furthermore, the search space has been reduced so that can give a guarantee to achieve a globally optimum solution. It can be said that through this research we conclude obtaining solutions for scheduling sports matches using artificial intelligence algorithms according to software that simulates the problem and dispensing with the use of boring manual methods. The proposed algorithm is most useful when the number of teams is too large, in this case, the classical methods are not used for scheduling the teams. In this algorithm, the teams are played until all but one team has been eliminated, this team is the winner.

ACKNOWLEDGEMENTS

The authors are very grateful to the University of Mosul/ College of Medicine and College of Nursing for their provided facilities, which helped to improve the quality of this work.

REFERENCES

[1] M. Aghababaei and M. Farsangi, "Coordinated Control of Low Frequency Oscillations Using Improved Monkey Algorithm," *International Journal on Technical and Physical Problems of Engineering,* vol. 4, no. 2, pp. 13-17, 2012.

[2] A. Mucherino, C. Lavor and N. Maculan, "Comparisons between an Exact and a Meta-Heuristic Algorithm for the Molecular Distance Geometry Problem," in *Genetic and Evolutionary Computation Conference*, Canada, Montreal, 2009.

[3] A. Mucherino and O. Seref, "Monkey Search: A Noval Meta-heuristic Search for Global Optimization, Data Mining, System Analysis and Optimization in Biomedicine," in *AIP Conference Proceedings*, 2007.

[4] N. J. Navimipour, S. H. Shabestari and V. S. Samaei, "Minimize Energy Consumption and Improve the lifetime of Heterogeneous Wireless Sensor Network by Using Monkey Search Algorithm," in *International Conference on Information and Knowledge Management*, Singapore, 2012.

[5] R. Vasundhare Devi and S. Siva Sathya, "Monkey Behavior Based Algorithms- A Survey," *International Journal Intelligent Systems and Applications,* vol. 12, no. 12, pp. 67-86, 2017, doi:10.5815/ijisa.2017.12.07.

[6] C. M. Ituarte-Villarreal, N. Lopez and J. F. Espiritu, "Using the Monkey Algorithm for Hybrid Power Systems Optimization," in *Procedia Computer Science*, Washington, 2012, doi:10.1016/j.procs.2012.09.082.

[7] R. Zhao and W. Tang, "Monkey Algorithm for Global Numerical Optimization," *Journal of Uncertain Systems, ACADEMIC,* vol. 2, no. 3, pp. 165-176, 2008.

[8] D. Annis and S. Wu, "A Comparison of Potential Playoff Systems for NCAA I-A Football," *The American Statistician,* vol. 60, no. 2, pp. 151-157, 2006.

[9] C. T. Edwards, "Double-Elimination Tournaments:Counting and Calculating," *The American Statistician,* vol. 50, no. 1, pp. 27-33, 1996, doi:10.1080/00031305.1996.10473538.

[10] M. E. Glickman, "Bayesian Locally Optimal Design of Knockout Tournaments," Journal of Statistical Planning and Inference, vol. 138, no. 1, pp. 2117-2127, 2008, doi:10.1016/j.jspi.2007.09.007.

[11] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Journal Computers and Operation Research,* vol. 13, no. 5, pp. 533-549, 1986, doi:10.1016/0305-0548(86)90048-1.

[12] Z. Z. Abidin, M. R. Arshad and U. K. Ngah, "A Simulation Based Fly Optimization Algorithm for Swarms of Mini Autonomous Surface Vehicles Application," *Indian Journal of Geo-Marine Sciences,* vol. 40, no. 2, pp. 250-266, 2011.

[13] I. N. Alkallak and R. Z. Sha'ban, "Tabu Search Method for Solving the Traveling Salesman Problem," *Al-Rafidain Journal of Computer Sciences and Mathematics,* vol. 5, no. 2, pp. 141-153, 2008.

[14] M. Gendreau, "An Introduction to Tabu Search," in *Handbook of Metaheuristics*, vol. 57, Boston, International Series in Operation Research and Management Science. Springer, 2003, pp. 37-54, doi:10.1007/0-306-48056-5_2

[15] M. Dorigo, M. Birattari and T. Stutzle, Artists, *Ant Colony Optimization: Artificial Ants as a C Computational Intelligence Technique.* [Art]. IEEE, 2006.

[16] I. N. Alkallak, "A Hybrid Ant Colony Optimization Algorithm to Solve Assignment Problem by Hungarian Method," *Al-Rafidain Journal of Computer Sciences and Mathematics,* vol. 6, no. 2, pp. 159-175, 2009, doi:10.33899/csmj.2009.163805.

[17] I. N. Alkallak and R. Z. Sha'ban, "A Survey of Two Optimization Methods to Solve A Modified Minimal Spanning Tree Problem in Undirected Tree Graph," *Al-Rafidain Journal of Computer Sciences and Mathematics,* vol. 8, no. 1, pp. 117-134, 2001.

[18] F. A. Ali and M. A. Tawhid, "A Hybrid Cuckoo Search Algorithm with Nelder Mead Method for Solving Global Optimization Problems," *Springer International Publishing,* vol. 5, no. 473, pp. 1-22, 2016, doi:10.1186/s40064-016-2064-1.

[19] I. N. Alkallak, "A Hybrid Algorithm from Cuckoo Search Method with N-Queens Problem," *Al-Rafidain Journal of Computer Sciences and Mathematics,* vol. 9, no. 2, pp. 183-192, 2012, doi:10.33899/csmj.2012.163710.

[20] R. Z. Sha'ban and I. N. Alkallak, "Adaptation of a Mosquito Swarm Algorithm to Solve a Binary Knapsack Problem," *Al-Rafidain Journal of Computer Sciences and Mathematics,* vol. 12, no. 2, pp. 13-24, 2018, doi:10.33899/csmj.2020.164663.

[21] I. N. Alkallak, Y. H. Alnema and R. Z. Sha'ban, "A Proposed Hybrid Algorithm for Constructing Knight Tour Problem by Sudoku Grid," *Journal of Advanced Research in Dynamical and Control Systems,* vol. 10, no. 10-special, pp. 2333-2342, 2018.

[22] I. N. Alkallak, "Using Magic Square of Order 3 to Solve Sudoku Grid Problem," *Tikrit Journal of Pure Science,* vol. 17, no. 4, pp. 244-248, 2012.

[23] I. N. Alkallak and R. Z. Sha'ban, "A Heuristic Genetic Algorithm for Independent Task Scheduling," *Dirasat Pure Sciences,* vol. 35, no. 2, pp. 180-193, 2008.

[24] I. N. Alkallak and R. Z. Sha'ban, "A Genetic Algorithm in Scheduling Processes with Non-Permeability," *Tanmiat Al-Rafidain,* vol. 30, no. 89, pp. 245-256, 2008, doi:10.33899/tanra.2008.161777.

[25] R. A. Ismail and W. K. Hamam, “Organization and Management in the Sports Field,” 1 ed., Mosul, Books and Document in Baghdad 168: Ibn Atheer Published/ University of Mosul, 2010, pp. 1-196.