❒   538

Lion optimization algorithm for team orienteering problem with time window

Esam Taha Yassen, Alaa Abdulkhar Jihad, Sudad H. Abed
Computer Center, University of Anbar, Al_Anbar, Iraq

Article Info	ABSTRACT
	Over the last decade, many nature-inspired algorithms have been received considerable attention among practitioners and researchers to handle several optimization problems. Lion optimization algorithm (LA) is inspired by a distinctive lifestyle of lions and their collective behavior in their social groups. LA has been presented as a powerful optimization algorithm to solve various optimization problems. In this paper, the LA is proposed to investigate its performance in solving one of the most popular and widespread real-life optimization problems called team orienteering problem with time windows (TOPTW). However, as any population-based metaheuristic, the LA is very efficient in exploring the search space, but inefficient in exploiting it. So, this paper proposes enhancing LA to tackle the TOPTW by utilizing its strong ability to explore the search space and improving its exploitation ability. This enhancement is achieved via improving a process of territorial defense to generate a trespass strong nomadic lion to prevail a pride by fighting its males. As a result of this improving process, an enhanced LA (ILA) emerged. The obtained solutions have been compared with the best known and standard results obtained in the former studies. The conducted experimental test verifies the effectiveness of the ILA in solving the TOPTW as it obtained a very competitive results compared to the LA and the state-of-the-art methods across all tested instances.

Corresponding Author:

Esam Taha Yassen
Computer Center
University of Anbar
Al_Anbar, Iraq
Email: co.esamtaha@uoanbar.edu.iq

1. INTRODUCTION

It is often very difficult for tourists who plan to visit a place for few days to visit all the sites they are fascinated with. Consequently, tourists need to determine what they consider to be the most valued attractions. The task of making a practicable plan to visit the most exciting places within the limited available time is surely difficult [1].

As described above, the problem was first introduced as the Orienteering Problem (OP) by Tsiligrides [2] (also, the problem is sometimes named as the Selective Traveling Salesman Problem, see Laporte and Martello [3]). Orienteering is a sport in which a player (competitor) has to select a path from a starting point (all competitors start out form the same point) to a final destination, going through control points alongside the path. A score is associated with every control point. A travel cost is associated with each pair of control points. The competitor has to select a group of control points to visit; hence the overall score is maximized, while the total travel cost does not exceed a given threshold [4].

The Team Orienteering Problem (TOP) describes as a generalization of the OP in which the competitors are organized in teams [5]. The TOP is known as a model of numerous real life applications, like the multi-vehicle version of the home fuel delivery problem [6, 7], selecting college football players [8], the sport game of team orienteering [9], several applications related to pick up and delivery services including public carriers and private fleets [10], and the routing technicians scheduling service [11].

Several variants of the TOP exist, and of them is the one that considers Time Window constraints (TOPTW). These constraints are inspired by several practical conditions and usually appear in routing problems in which every location must be visited within a previously identified time interval (the earliest and the latest time) at which the service should begin [5]. A list of places (points) is assigned in the TOPTW and every place is given a score, a service time and a time window (open and close). The aim is to maximize the total number of gained scores by with fixed number. These routes should be with a limited length and should also permit visiting locations at the precise time allocating a group of routes [5].

For TOPTW, an Ant Colony System is suggested by Montemanni and Gambardella [5]. Further improve Ant Colony System is proposed by Montemanni et. al. [4]. The algorithm includes two extra operations to enhance the ability of Ant Colony System. These operations focus on utilizing the best obtained solutions during the construction phase and applying the local search procedure on those solutions [4]. Two different versions of Simulated Annealing are introduced by Lin and Yu [12]; FastSA and SlowSA. FastSA is mainly applied when the quick responses are needed, while SlowSA is concerned about the solutions' qualities in exchange for more computational time [12]. The hybridization of the Greedy Randomized Adaptive Search Procedure (GRASP) and Evolutionary Local Search is proposed by Labadie et al [13] for the TOPTW. Hu and Lim [14] introduced a hybrid algorithm based on local search, Simulated Annealing and Route Combination. Within a particular number of iterations, this hybridization is iteratively incorporated those three components [14]. Cura [15] proposes hybridization based on an Artificial Bee Colony and simulated annealing in order to improve the TOPTW solution quality [15]. In order to achieve the balance between diversification and intensification, Gunawan et. al. [16] suggest an Iterated Local Search based on swap, 2-opt, insert and replace neighborhood structures. Additionally, they implemented the combination between acceptance criterion and perturbation mechanisms. A hybridization between Simulated Annealing and Iterated Local Search is proposed by Gunawan et. al. [17, 18] to solve the TOPTW. A hybrid artificial bee colony algorithm is introduced by Yu et. al. [19] to obtain an optimal solutions for small instances and near-optimal solutions for larger instances.

Based on the above and to the best knowledge of the researchers, the need to utilize a new algorithm is still necessary. Lion Optimization Algorithm (LA) which is inspired by special lifestyle of lions and their collective behavior in their social groups [20]. LA has been presented as a powerful optimization algorithm to solve various optimization problems. This stimulates the current study to propose LA for solving the TOPTW.

2.   LION ALGORITHM

Natural computation is an effective computing area that aims at developing search, optimization and machine learning algorithms by analyzing natural phenomena and model it to problem solving conditions [21]. Lion's algorithm is proposed based on the lions' social behavior, which is to be the strongest in every generation. LA searches an optimal solution based on two unique lions' behaviors, namely territorial defense and territorial takeover. Territorial defense happens between the occupant males and wandering males while the territorial takeover happens between the old resident male and new mature resident male [20, 22].

Definition 1: A Lion represents a solution to be identified, while the cubs are solutions produced from current solutions [20].

Definition 2: Territorial Defense is meant to be the procedure of assessing the current solution (territorial lion) and produced solution (nomadic lion). More specifically; the current solution is replaced by the new solution whenever the new solution sounds better than the current one. Hence, all the obtained solutions of old solution are vanished [20].

Definition 3: Territorial Takeover is defined as the act of retaining the best derived male and female solutions. These solutions are rather more efficient than new solutions. Thus, current solutions are vanished in the pride [20].

```
Algirithm 1. Lion Algorithm [20]
Step 1: Pride Generation
        Step 1.1: Generate territorial male and female subject to solution constraints
Step 2: Mating
        Step 2.1: Crossover
        Step 2.2: Mutation
        Step 2.3: Gender grouping
        Step 2.4: Kill sick/ weak cubs
        Step 2.5: Update Pride
```

```
Step 3: Territorial Defense
        Step 3.1: Keep the record of cubs' age
        Step 3.2: Do
                Step 3.2.1: Generate and Trespass Nomadic lion
                Until stronger Nomadic lion trespasses
        Until Cubs get matured
Step 4: Territorial Takeover
        Step 4.1: Selection of best lion and lionesses
        Step 4.2: Go to Step 2 until termination criteria is met
```

LA can be commonly classified into four major steps on the basis of the nature of the functions they perform. These steps are, (i) Pride Generation, which is in charge of producing solutions, (ii) Mating, which denotes the process of generating new solutions, (iii) Territorial Defense and (iv) Territorial Takeover, which both aim to substitute the new most excellent quality solution with the identified poorest quality solution. This process keeps iterating to allow heuristic search meeting the nearer solution to the preferable one [20].

Definition 4: A constantly altering solution pool and as well as altering size is representing pride. It starts with a couple of random solutions, one of these solutions refers to the male while the other one refers to the female. In that pool, the action of updating the generated solutions and the vanishing of undesirable ones take place [20].

Definition 5: Mating refers to the process of generating the new finest solutions from the existing solutions that involve crossover and mutation for generating new solutions, gathering genders that help to discover divergence within the solutions. Assuring that the newly generated solutions are the best by eliminating sick/ weak cubs [20].

The LA's search procedure primarily concentrates on detecting the best solutions, which solve the objective function i.e. minimize or maximize the objective function.

Definition 6: Gender grouping is the action that splits the obtained solution pool into a pair of collections, where there is a collection that includes the male cubs and the other includes the females [20].

3.   IMPROVED LION ALGORITHM

Despite the positive characteristics of LA and its successful application to address various problems, it still suffers from slow convergence problem when addressing constraint problems especially TOPTW. As any population-based metaheuristic [23-25], this problem is attributed to the fact that LA is effective in exploring the search space but ineffective in exploiting it. In order to improve this issue, an Improved Lion Algorithm (ILA) is proposed. In ILA, a territorial defense (algorithm 1, step 3) is improved through replacing the traditional process of generate and trespass nomadic lion with a new one. More precisely, instead of randomly generate a nomadic lion; a new process aims to introduce a strong nomadic lion to take over a pride by fighting its males. This aim is achieved via utilizing different criteria during the search to select which solution is the best to handle the pride. According to TOPTW, four criteria have a significant effect on the performance of the generated solution; closing time window (C_time), opening time window (O_time), profit of the location (Prof) and distance between two vertexes (Dist). As a result of these criteria, five generating heuristics are emerged to select the most appropriate one which is adopted to generate the strong nomadic lion. These heuristics are: Heu1, Heu2, Heu3, Heu4 and Heu5. The generation process is the same for all heuristics, only the adopting criteria change from one heuristic to another.
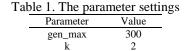
a)   *Heu1*: This generating heuristic deals with two criteria out of four which are the closing time window *C_time* and distance *Dist.* In this method, the goal is to search for a location with the nearest distance *Dist* and close time *C_time* to be adopted in the generated solution.

b)   *Heu2:* In this method three criteria have been taking into consideration. Those three criteria are open time window *O_time,* close time window *C_time* and distance *Dist.* this method tries to select a location with the largest open time window *O_time* and the nearest distance *Dist* and close time *C_time.*

c)   *Heu3:* This method deals with three criteria which are *Prof*, *C_time,* and *Dist.* This method tries to select a location based on a threshold number that can be obtained by the multiply the *Dist* by the result of dividing the *Prof* on C_time.

d)   *Heu4:* This method deals with two criteria which are *Prof* and *Dist.* trying to find a threshold number by dividing the **Prof** by **Dist** to decide whether to select a location or not.

e)   *Heu5:* This method deals with only two criteria which are the **Prof** and Dist. In this method, the aim is to select a location with the largest profit and nearest distance.

The following algorithm illustrates the process of generating a nomadic lion (Solution) (Algorithm 2).

```
Algoritm 2. Generating a nomadic lion
        Repate
           for each player (i)
              select_vertex = null
              threshold_select_vertex = 0
              for each unvisited Vertex (uver)
                  get the criteria values (C_time_uver, O_time_uver, Dist_uver, Prof_uver)
                  Normalize these values
                  calculate the threshold_uver according to the selected heuristic (Heu1,
                  Heu2, Heu3, Heu4 and Heu5)
                  randomly generate ratio number
                   if   ratio  <   threshold_uver  and  threshold_uver  better  than
                  threshold_select_vertex
                      select_vertex = uver
              Add this select_vertex to the path of player_i :
              Path=Path ∪ select_vertex
        Until no vertex can be visited
        Return Solution
```

4.   EXPERIMENTAL SETUP

The proposed algorithms (LA and ILA) are programmed and executed with C# which was running on a PC equipped with windows7, Intel processor core i7 CPU 2.10GHz, RAM 8.00GB. In this work, the appropriate values for the parameters of the proposed algorithms are determined based on a preliminary test or as suggested by previous studies. The proposed algorithms, during the preliminary test, are executed 10 runs on some instances. Finally, the parameter settings that have been adopted in this work are summarized in Table 1.

Table 1. The parameter settings

Parameter	Value
gen_max	300
k	2
Age_{mat}	5

4.1. The toptw benchmark

In order to evaluate the performance of LA and ILA, a larger benchmark is adopted. This benchmark is provided by Vansteenwegen et. al. [1] based on Solomon's and Cordeau et al.'s benchmarks; the instances of these benchmarks can be publicly found at (https://www.mech.kuleuven.be/en/cib/op). It should be taken into consideration that the benchmarks used in this work contain instances with different situations related to nodes' positions, scores, service times and the density of time windows. Those benchmarks were designed for the Vehicle Routing Problem with Time Windows (VRPTW) and the Multi Depot Periodic VRPTW respectively. Hence, these benchmarks are very realistic. Additionally, these benchmarks have been adopted by most studies to evaluate the effectiveness of their heuristics [26].

The Solomon's benchmark involves 56 instances each of them has 100 nodes. According to the distribution of nodes, the Solomon's instances are categorized into three classes (R, C and RC). Class C contains clustered nodes in which the distances between nodes are short. Class R contains randomly distributed customers in which the distances between nodes are comparatively longer than class C. Class RC contains mixed distribution nodes (clustered and randomized). The Cordeau et al.'s benchmark involves 20 instances each of them has different number of nodes; varying from 48 to 288 nodes (pr01–pr20) [1].

4.2. Results and discussion

In order to investigate the performance of the LA and ILA for TOPTW, we conduct three sets of experiments. In the first experiments, the efficiency of LA in solving VRPTW is evaluated. In the second experimental test, the effect of the process that generates a nomadic lion on the LA's performance is analyzed. Finally, the results of ILA are compared with those of standard LA. The proposed algorithms were executed for 11 independent runs. Then, the best (Max), average (Avr) and minimum (Min) are reported. The proposed algorithms are tested on Solomon's and Cordeau et al.'s benchmarks.

4.2.1. The Standard LA results

In this experiment, the effectiveness of LA in solving the TOPTW is investigated. Table 2 presents the *Max*, *Avr* and *Min* for LA over **11** runs.

Table 2. Results of standard LA

Instance	Max	Avr	Min	Instance	Max	Avr	Min
50_c101	850	750	650	pr01	297	252.5	208
50_c102	800	755	710	pr02	620	547.5	475
50_r101	146	115	84	r101	371	345	319
50_r102	655	594.5	534	r102	1331	1185.5	1040
50_rc101	640	485	330	r201	1309	1249.5	1190
50_rc102	960	850	740	r202	1202	1138	1074
c101	1570	1520	1470	rc101	916	845.5	775
c102	1570	1500	1430	rc102	1506	1224.5	943
c201	1580	1545	1510	rc201	1378	1345.5	1313
c202	1640	1585	1530	rc202	1222	1155	1088

4.2.2. The effect of generating a nomadic lion process

In this experiment, the effect of the process to generate a nomadic lion on the LA's performance is investigated. So, the effectiveness of different suggested heuristics based on different criteria are verified to select the most appropriate one which is adopted to generate a strong nomadic lion. The results of these methods are presented in Table 3. These results show that the *Heu1* gained the best results on 16 instances out of 20 tested instances. Consequently, due to the fact that the *Heu1* is more efficient than others as it obtained the based results, it will be adopted in *ILA*.

The results of the *ILA* are compared with these of the standard *LA*. Twenty instances are selected in this experiment. For each tested instance, the *Max, Avr, Min* values are reported in Table 4. In terms of the *Max* values, the *ILA* achieved better results than *LA* in 18 out of 20 instances. Based on the *Avr* and *Min* values, the *ILA* achieved the best *Avr* results on all tested instances.

Table 3. results of different suggested heuristics based on different criteria

Instance	Heu1	Heu2	Heu3	Heu4	Heu5
50_c101	860	860	860	860	860
50_c102	860	860	860	850	860
50_r101	721	577	686	665	685
50_r102	721	721	721	721	721
50_rc101	970	820	960	970	870
50_rc102	950	920	930	920	950
c101	1810	1810	1810	1810	1640
c102	1800	1800	1740	1560	1560
c201	1810	1810	1800	1810	1730
c202	1810	1810	1790	1740	1650
pr01	299	280	298	267	286
pr02	620	620	616	604	604
r101	1293	877	1258	1119	1171
r102	1397	1369	1369	1285	1281
r201	1445	1342	1419	1418	1357
r202	1387	1401	1370	1331	1267
rc101	1510	1324	1529	1409	1221
rc102	1480	1292	1512	1506	1275
rc201	1705	1564	1616	1621	1531
rc202	1569	1453	1572	1532	1346

Table 4. Comparison between standard LA and ILA

Instance	LA			ILA		
	Max	Avr	Min	Max	Avr	Min
50_c101	850	750	650	860	860	860
50_c102	800	755	710	860	855	850
50_r101	146	115	84	721	649	577
50_r102	655	594.5	534	721	721	721
50_rc101	640	485	330	970	895	820
50_rc102	960	850	740	950	935	920
c101	1570	1520	1470	1810	1725	1640
c102	1570	1500	1430	1800	1680	1560
c201	1580	1545	1510	1810	1770	1730
c202	1640	1585	1530	1810	1730	1650
pr01	297	252.5	208	299	283	267
pr02	620	547.5	475	620	612	604
r101	371	345	319	1293	1085	877
r102	1331	1185.5	1040	1397	1339	1281
r201	1309	1249.5	1190	1445	1393.5	1342
r202	1202	1138	1074	1387	1334	1267
rc101	916	845.5	775	1510	1375	1221
rc102	1506	1224.5	943	1480	1393.5	1275
rc201	1378	1345.5	1313	1705	1618	1531
rc202	1222	1155	1088	1569	1459	1346

These results demonstrated that giving a better chance for points with the nearest distance (*Dist*) and close time (*C_time*) to be selected and added into the new solution can effectively raise the performance of *ILA*. So, *Heu1* method has given the best result among the other four proposed methods. On the other hand, the results achieved using *Heu2* is competitively close to these achieved using *Heu1* but with the cost of dealing three criteria instead of two. As a result, we adopted *Heu1* to select the nomadic lion since it has given the best result with lost cost.

4.2.3. Comparison of ILA with the state of the art methods

To demonstrate the performance of the *ILA* in solving TOPTW, this experiment compares the results obtained by *ILA* with those found and presented by the other algorithms in literature and the best-known results obtained in the literature. Five of the recent studies which achieved the best known results were selected for comparison with the proposed method. These studies are as follows:

a) **I3CH:** iterative three-component heuristic [27].
b) **GVNS:** granular variable neighborhood search based on linear programming [28].
c) **SSA:** slow version of the simulated annealing heuristic [29].
d) **ABC:** artificial bee colony algorithm [30].
e) **ILS:** iterated local search algorithm [31].

In terms of *Max*, Table 5 reveals that *ILA* has attained competitive results over all instances and better results in some others. Finally, the obtained results have shown that the *ILA* outperformed the standard *LA*.

Table 5. Comparison among different heuristics

Instance	I3CH	GVNS	SSA	ABC	ILS	ILA
50_c101	-	-	-	-	-	860
50_c102	-	-	-	-	-	860
50_r101	-	-	-	-	-	721
50_r102	-	-	-	-	-	721
50_rc101	-	-	-	-	-	970
50_rc102	-	-	-	-	-	950
c101	1810	1754	1770	1786	1720	1810
c102	1810	1794	1810	1810	1790	1800
c201	1810	1810	1810	1810	1810	1810
c202	1810	1810	1810	1810	1810	1810
pr01	-	-	-	-	608	299
pr02	-	-	-	-	1180	620
r101	1458	1432.2	1455	1457.4	1441	1293
r102	1458	1441.2	1458	1455.4	1450	1397
r201	1458	1458	1458	1458	1458	1445
r202	1458	1450.2	1458	1458	1443	1387
rc101	1724	1690.2	1712	1713.6	1686	1510
rc102	1724	1685	1718	1705.2	1659	1480
rc201	1724	1724	1724	1451.2	1724	1705
rc202	1719	1702.8	1714	1724	1686	1569

"−" indicates that the results have not been reported.

5. CONCLUSION

In this paper, the LA was adopted to tackle the TOPTW. Through analyzing the LA results, it was observed that it succeeded in tackling the TOPTW via improving the pride quality. But, the ability of LA is gradually decreased. This is due to the fact that LA suffers from slow convergence which prevents it from improving the quality of obtained solutions. This problem often occurred because of the LA is satisfying at exploring the search space but not at exploiting it. In order to enhance the exploitation ability of LA, a new process for generating a trespassing nomadic lion was introduced and resulted in the improved LA denoted as (ILA). Instead of randomly generating a nomadic lion, this process works to generate a strong nomadic lion to take over a pride by fighting its males. The obtained results have shown that the performance of ILA is much better than the standard LA due to the fact that generating a strong nomadic lion can assist the exploitation ability of ILA. Thus, this work concludes that the proposed ILA successes in addressing the TOPTW, and it can be applied to solve other combinatorial optimization problems.

REFERENCES

[1] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1-10, 2011.
[2] T. Tsiligirides, "Heuristic methods applied to orienteering," *Journal of the Operational Research Society*, vol. 35, no. 9, pp. 797-809, 1984.
[3] G. Laporte and S. Martello, "The selective travelling salesman problem," *Discrete applied mathematics*, vol. 26, no. 2-3, pp. 193-207, 1990.
[4] R. Montemanni, D. Weyland, and L. Gambardella, "An enhanced ant colony system for the team orienteering problem with time windows," *2011 International Symposium on Computer Science and Society*, pp. 381-384, 2011.
[5] R. Montemanni and L. M. Gambardella, "An ant colony system for team orienteering problems with time windows," *Foundation Of Computing And Decision Sciences*, vol. 34, no. 4, pp. 287-306, 2009.
[6] B. L. Golden, L. Levy, and R. Vohra, "The orienteering problem," *Naval Research Logistics (NRL)*, vol. 34, no. 3, pp. 307-318, 1987.
[7] B. L. Golden, Q. Wang, and L. Liu, "A multifaceted heuristic for the orienteering problem," *Naval Research Logistics (NRL)*, vol. 35, no. 3, pp. 359-366, 1988.

[8] S. Butt and D. Ryan, "A heuristic for the multiple path maximum collection problem," *Computers and Operations Research*, vol. 26, pp. 427-441, 1999.

[9] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *European journal of operational research*, vol. 88, no. 3, pp. 464-474, 1996.

[10] R. W. Hall and M. Racer, "Transportation with common carrier and private fleets: system assignment and shipment frequency optimization," *IIE transactions*, vol. 27, no. 2, pp. 217-225, 1995.

[11] H. Tang and E. Miller-Hooks, "A tabu search heuristic for the team orienteering problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1379-1407, 2005.

[12] S.-W. Lin and F. Y. Vincent, "A simulated annealing heuristic for the team orienteering problem with time windows," *European Journal of Operational Research*, vol. 217, no. 1, pp. 94-107, 2012.

[13] N. Labadie, J. Melechovský, and R. W. Calvo, "Hybridized evolutionary local search algorithm for the team orienteering problem with time windows," *Journal of Heuristics*, vol. 17, no. 6, pp. 729-753, 2011.

[14] Q. Hu and A. Lim, "An iterative three-component heuristic for the team orienteering problem with time windows," *European Journal of Operational Research*, vol. 232, no. 2, pp. 276-286, 2014.

[15] T. Cura, "An artificial bee colony algorithm approach for the team orienteering problem with time windows," *Computers & Industrial Engineering*, vol. 74, pp. 270-290, 2014.

[16] A. Gunawan, H. C. Lau, and K. Lu, "An iterated local search algorithm for solving the orienteering problem with time windows," *European conference on evolutionary computation in combinatorial optimization*, pp. 61-73, 2015.

[17] A. Gunawan, *et al*, "SAILS: hybrid algorithm for the team orienteering problem with time windows," *MISTA 2015: Proceedings of the 7th Multidisciplinary International Scheduling Conference*, pp. 276-295, 2015.

[18] A. Gunawan, H. C. Lau, P. Vansteenwegen, and K. Lu, "Well-tuned algorithms for the team orienteering problem with time windows," *Journal of the Operational Research Society*, vol. 68, no. 8, pp. 861-876, 2017.

[19] F. Y. Vincent, P. Jewpanya, S.-W. Lin, and A. P. Redi, "Team orienteering problem with time windows and time-dependent scores," *Computers & Industrial Engineering*, vol. 127, pp. 213-224, 2019.

[20] B. Rajakumar, "The Lion's Algorithm: a new nature-inspired search algorithm," *Procedia Technology*, vol. 6, pp. 126-135, 2012.

[21] J. Liu and K. C. Tsui, "Toward nature-inspired computing," *Communications of the ACM*, vol. 49, no. 10, pp. 59-64, 2006.

[22] M. Yazdani and F. Jolai, "Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm," *Journal of computational design and engineering*, vol. 3, no. 1, pp. 24-36, 2016.

[23] Harikishore, S., & Sumalatha, V., "Ant colony optimization based energy efficiency for improving opportunistic routing in multimedia wireless mesh network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 3, pp. 1371-1378, 2019.

[24] Rasheed, M., Omar, R., Sulaiman, M., & Halim, W. A., "Particle swarm optimisation (PSO) algorithm with reduced numberof switches in multilevel inverter (MLI)," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1114-1124, 2019.

[25] Alhamdani, I. M., & Ibrahim, Y. I., "Swarm intelligent hyperdization biometric," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, pp. 385-395, 2020.

[26] E. T. Yassen, M. Ayob, M. Z. Ahmad Nazri, and N. R. Sabar, "A hybrid meta-heuristic algorithm for vehicle routing problem with time windows," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 6, pp. 1550021, 2015.

[27] H. Qian and L. Andrew, "An iterative three-component heuristic for the team orienteering problem with time windows," European Journal of Operational Research, vol. 232, pp. 276–286, 2014.

[28] N. Labadie, *et al*., "The team orienteering problem with time windows: An lp-based granular variable neighborhood search," *European Journal of Operational Research*, vol. 220, no. 1, pp. 15–27, 2012.

[29] S.-W. Lin and V. F. Yu, "A simulated annealing heuristic for the team orienteering problem with time windows," *European Journal of Operational Research*, vol. 217, no. 1, pp. 94–107, 2012.

[30] T. Cura, "An artificial bee colony algorithm approach for the team orienteering problem with time windows," *Computers & Industrial Engineering*, vol. 74, pp. 270–290, 2014.

[31] P. Vansteenwegen, *et al*, "Iterated local search for the team orienteering problem with time windows," *Computers & Operations Research*, vol. 36, no. 12, pp. 3281–3290, 2009.

BIOGRAPHIES OF AUTHORS

Esam Taha Yassen is a lecturer in the college of Computer and Information Technology at the University of Anbar, Iraq since 2002. He has obtained his PhD in Computer Science at The University Kebangsaan Malaysia (UKM) in 2015. His main research areas include meta-heuristics, hyper-heuristics and combinatorial Optimization problems especially, routing and scheduling. He has been served as a programme committee for four international conferences and reviewers for high impact journals. He is a researcher in Data Mining and Optimization Research Group (DMO), Centre for Artificial Intelligent (CAIT), UKM. Currently, he is the manager of Computers Centre in University of Anbar.

Alaa Abdalqahar Jihad was born in Anbar-Iraq in 1985. He received his B.Sc. from Faculty of Computer Science at Anbar University, Iraq in 2009. The MSc. degree Faculty of Computer Science at Anbar University, Iraq 2012. His research interests are, Data Warehouse, Data Mining, Artificial Intelligent, Machine Learning and Natural Language Processing.

Sudad H. Abed is a full-time instructor in the University of Anbar. He received his master degree in computer science from Ball State University in 2016. His research area of interest is machine learning, neural networks, deep learning, and data analytics. He has a solid experience in web application development and currently servers as the director of the software development department in the University of Anbar.