25 International Journal of Smart Electrical Engineering, Vol.13, No.1, Winter 2024 ISSN: 2251-9246
EISSN: 2345-6221
pp. 25:31

Swarm-Based Scheduling Algorithm for Lifetime Improvement of Visual Sensor Networks

Mir Gholamreza Mortazavi[1], Mirsaeid Hosseini Shirvani[*2], Arash Dana[1], Mahmood Fathy[3,4]

[1]Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[2]Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran, mirsaeid_hosseini@iausari.ac.ir
[3]School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[4]School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

Abstract

Visual sensor networks (*VSNs*) apply directional sensors that can be configured only in one direction and also can be set in one of the possible observing ranges. In this battery-resource-limited environment, battery management and network lifetime expansion are still important challenges. The target coverage problem in such networks, in which all of the specified targets must be continuously observed and monitored by administrators is formulated as an integer linear programming problem (ILP) that is an NP-Hard problem. Although several approaches have been presented in the literature to solve the aforementioned problem, the majority of them suffer from getting stuck in the local trap and low exploration in search space. To address the issue, a discrete cuckoo-search optimization algorithm (*DCSA*) is extended to solve this combinatorial problem. The discrete operator of the proposed algorithm is designed in such a way that explore search space efficiently and lead to balancing in the local and global search process. The proposed algorithm was examined in different conducted scenarios. The returned results of simulations of numerous scenarios show the dominance of the proposed algorithm in comparison with other existing approaches in terms of network lifetime maximization. In other words, the proposed *DCSA* has 19.75% and 13.75% improvement in terms of network average lifetime expansion against *HMNLAR* and *GA*-based approaches respectively in all scenarios.

Keywords: Visual Sensor Network (*VSN*). Directional Sensor Network (*DSN*). Discrete Cuckoo Search Optimization Algorithm (*DCSA*). Network Lifetime Expansion. Scheduling
Article history: Received 2023/12/12; Revised 2024/01/20; Accepted 2024/01/28, Article Type: Research paper

https://doi.org/ 10.30495/ijsee.2024.2003747.1297

1. Introduction

A typical directional sensor network (*DSN*) comprises a set of directional sensors that can be configured based on the situation. Each directional sensor is adjusted in one of its possible directions and is a configurable range based on the target distance. The ultrasonic, radar, and video camera sensors, and visual sensor networks (*VSNs*) are typical kinds of *DSN* [1-3]. Target coverage is a key issue in harsh or expensive industries in which the predetermined targets in the observing field should be permanently observed and monitored by admins to react carefully according to the events in observing environment [4]. In this environment, one of the most important challenges is to manage the battery consumption of sensors in such a way that the sensors are utilized uniformly. Although different technologies for energy harvesting are used, utilizing these technologies is impossible or expensive in some cases. However, battery usage management is still a challenge. Fig. 1 illustrates a common *DSN*. In this network, 5 different targets must be continuously observed and monitored by 5 available configurable directional sensors in the two-dimensional observing environment.

The uniformly engagement of sensors leads to distribute their battery usage and more round the targets can be observed. On the other hand, the naïve usage of sensors may lead into inefficiency in which the early battery depletion does not let the network observe all of the targets in the field. To solve the problem efficiently, the new concept of "*cover*" is defined. A *cover* is a set of configured sensors not all available sensors in which they collaboratively can observe all of the targets in the field.

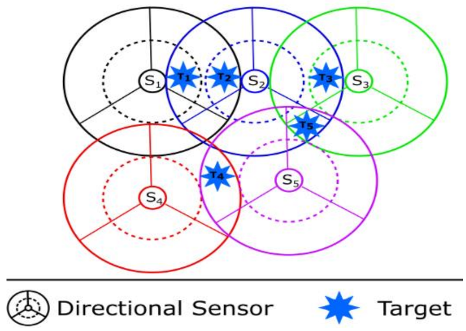

Fig. 1. A common DSN with 5 directional sensors observing 5 different targets [1].

Eq. (1) draws a valid *cover* by utilizing only 4 sensors out of 5 alive sensors. Note that, sensor S_4 is configured in sleep mode to save energy.

$$cover = \{S_1(1,2), S_2(2,2), S_3(3,1), S_5(3,2)\} \quad (1)$$

Note that, the notation $S_5(3,2)$ means that the sensor S_5 is activated in its 3rd direction by adjustment of its 2nd range observing target T_4. Once a *cover* is utilized for some time, the other sensors are set to sleep mode to save energy in that time. Then, another *cover* that utilizes other configured sensors is engaged. This procedure is iterated until the battery depletion of sensors does not let to creating another *cover*. The sum of the engagement time of utilized covers is named network lifetime. The objective of this paper is to maximize network lifetime. This is an optimization problem which needs an efficient algorithm. The need for developing an efficient algorithm is necessary to create ample covers and switch between them to utilize all sensors as uniformly as possible. In this way, network lifetime is promisingly increased. Therefore, the innovations of this paper are enlisted below:

- It formulates the network lifetime maximization into a discrete optimization model.
- It designs a discrete cuckoo search optimization scheduling algorithm (DCSA) to solve stated discrete optimization problem.
- The proposal is tested in different scenarios to be trustable for engagement in the real world.
- The rest of the paper is organized as follows.

Section 2 reviews related works. Section 3 presents the problem statement formally. The suggested DCSA is elaborated in section 4. The performance of this proposal is verified in Section 5. Section 6 concludes the paper along with getting a clue for future direction.

2. Related Works

The profound literature review in target coverage and lifetime maximization of DSNs, reveals that we can classify algorithms in two main categories. The categories are heuristic-based and meta-heuristic-based approaches. Target coverage problem is a class of k-coverage problem in DSNs once the parameter k is taken 1. In k-coverage problem, each target must be permanently observed by k different sensors to improve reliability of observation [3]. To this end, Zannat et al. suggested number of heuristics to figure out the k-coverage problem in VSN so that each video sensor is configured in right direction and minimum coverage range to optimal electricity usage [4]. Jinglan Jian et al. proposed a heuristic approach to take full-view target coverage in VSNs [5]. Similar heuristic has been suggested in literature to apply sensor nodes in WSNs to maximize target coverage subject to taking sensor directions and battery limitations at the same time [6]. Other heuristics have been presented by Tan and Jarvis to balance between detection and target coverage process [7]. Xu et al. proposed a hierarchical target-oriented multi-agent coordination framework (HiT-MAC) to improve target coverage issue in sensor networks [8]. They formulated the problem into an integer linear programming (ILP). To figure out this combinatorial ILP problem, the two-level heuristic was developed. In addition, a greedy target coverage-aware approach has been extended in literature to solve network lifetime maximization problem in battery-limited environments [9]. Ahmad and Kohil presented bi-direction sensor placement algorithm to solve k-coverage problem to reduce data transmission costs and also increase network lifetime [10]. Mohammadi et al. introduced couple of heuristics each of which considers one of the prominent network parameters to solve target coverage problem [11]. Finally, they proposed a comprehenisive heuristic by incorporating all weighted effective network parameters which lead promising solution. Al Zishan et al. in [12] formulated target coverage and k-target coverage problems into an integer quadratic programing problem (IQLPP). In this paper two different over and under provisioned networks were investigated. A heuristic optimization algorithm so-called maximum coverage with minimum sensors (MCMS) was extended to solve stated IQLPP. To this end, a greedy algorithm opts a sensor in every round of optimization process by taking a predetermined criterion. This criterion is to select sensor that can cover the most number of targets as possible. In the large-scale problems, the heuristics seldom produce the most efficient solutions because the heuristics explore search space according predetermined criteria in which they cannot compensate the inefficient passed paths. To mitigate the challenge, the meta-heuristics have more time to explore search space and change the passed directions. A cuckoo search optimization algorithm was utilized in wireless sensor networks (WSNs) for scheduling of sensors to cover targets [13]. A discrete grey wolf optimization algorithm (DGWA) was proposed by Ekhlas et al. to solve the *k*-coverage problem in DSNs [3]. They proposed numerous discrete exploration and exploitation operators each of which was tailored with the discrete search space. Finally, the most efficient permutations led to promising solutions. One of the most applicable meta-heuristic algorithms is genetic-based (*GA*) algorithm because it can simple adapt itself with all kinds of problems [14-16]. To this end, a GA algorithm was suggested in literature to solve the k-coverage problem in both over-provisioned and under-provisioned DSNs [17]. Investigation on meta-heuristic algorithm reveals that most of them have local search trend while others have global search trend. In addition, majority of them have evolutionary inclination which may be threatened with local optimum trap. One of the successful mat-heuristic algorithm is cuckoo search optimization algorithm (*CSA*) that engages superior and inferior solutions and gives the chance only to bad solutions [20]. This procedure lessens the getting stuck in local trap possibility. One of the most important operators of *CSA* is to utilize *lévy flight* process to keep randomness of searching. Since the stated problem is discrete, the discrete version of *CSA* is customized and presented.

The literature comparison reveals that further processing is needed to introduce operators. In addition, it needs to call them at the right time to lead balancing in local and global searches and to produce the promising solutions as well.

3. Problem Formulation

In this section the problem statement is done. It is necessary to create a *cover* at a time. This *cover* including limited number of configured sensors in which every *cover* can observe all of the targets in the observing environment itself. Note that, if every *cover* is continuously utilized to monitor all targets, this procedure has the lowest efficiency [1, 3]. Therefore, designing the intelligent procedure which finds different covers and switch between them leads to better performance in terms of network lifetime maximization. If a cover C_i is engaged for T_i unit of time in i -th round; then, the sum of all T_i for i=1,…,*M* is named network lifetime. The objective of this paper is to maximize network lifetime that Eq. (2) calculates. This is an integer linear programming (*ILP*) problem which needs an efficient solution.

$$\text{Max.} \sum_{i=1}^{M} T_i \tag{2}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{M}\sum_{L=1}^{NRng} X_{ijdL}\times\Delta^L\times T_i \le E \quad \text{where } d\in\{1,2,\ldots,DMax\},\; X_{ijdL}\in\{0,1\} \tag{3}$$

$$\sum_{i=1}^{n} Z_i = n \quad \text{, where } Z_i \in \{0,1\} \tag{4}$$

$$T_i \ge 0 \text{ , } \forall \; i=1,2,\ldots,M \tag{5}$$

Eq. (2) is the main objective that must be maximized subject to keeping some constraints. Eq. (3) shows that the sensors' full battery (E) can be shared between rounds once the sensors are utilized for observation. Otherwise, the sensors are configured to sleep-mode to save energy. Two decision binary variables X_{ijdL} are Z_i used for different goals. For instance, the former is used to indicate the sensor S_i is engaged in j–th round in its d-th direction and L–th range whereas the latter Z_i is used to show the target i–th is observed or not. Eq. (4) emphasizes that all of the targets in the field must be observed. Recall that the notation Δ^L is the battery usage pattern of each sensor once it is configured in its L–th range.

4. Proposed Meta-heuristic-based Methodology for Lifetime Improvement of DSN

There are several meta-heuristic-based optimization algorithms. The majority of them are suitable for continuous optimization search space. Also, some of them are evolutionary-based in which their operators conduct solutions toward better area. Ignoring not toward not to explore worst area increase the getting stuck in local trap. To bridge the gap, the cuckoo search-based algorithm (CSA) is extended which has three promising features. Firstly, it is a discrete version tailored with the stated discrete optimization problem. Secondly, it even examines worse solutions which may produce efficient solutions probability. Thirdly, it introduced discrete operator which permutes search space efficiently.

A) Principles: Encoding and Fitness Function

One of the most important concepts in meat-heuristic approaches is how to present a candidate solution. The encoding indicates how the solution is encoded. The used encoding approach is similar to papers in [18-19]. In regard to DSN of Fig. 1, the *cover* calculated in Eq. (1) is encoded in Fig. 2.

Target	t_1	t_2	t_3	t_4	t_5
Configured Sensor	[1,1,2]	[1,1,2]	[3,3,1]	[5,3,2]	[2,2,2]

Fig. 2. An encoded cover as a candidate solution

The notation [5,3,2] in t_4's column means that the sensor S_5 is activated in its 3rd direction by the 2nd adjusted range to monitor target t_4. To evaluate the competency of each candidate solution, a fitness function is determined. The fitness function is directly depends on the objective of the problem. To this end, the new fitness function is introduced in which a cover with lesser sum of adjustment ranges of utilized sensor is favorable. In this paper, Eq. (6) calculate a fitness value of each *cover*.

$$Fitness(Cover) = \sum_{j:\, S_j(i,L)\epsilon\, cover}^{n} \Delta^L \tag{6}$$

The proposed algorithm is iterated until the new cover cannot be made. If the battery level of alive sensors permit to create another cover, the new cover with the lowest sum of possible battery consumption is created. This is done by performing an enhanced CSA. Since the CSA algorithm is an endless approach similar to other meta-heuristic-based algorithms, it is iterated for some rounds.

B) Description of the proposed Algorithm for solving the sated problem

Algorithm 1 is dedicated to solve SDN's lifetime expansion. It is iterated until the remaining battery of alive sensors can constitute the new cover. The main loop is between lines 7 and 40. Then, Algorithm 1 returns the efficient scheduling solution which includes the cover, its engagement time, and sensors' configurations. As the proposed algorithm is CSA-based, in each round, it splits solutions into two different sets that are inferior and superior set of cuckoos. Before doing so, all cuckoos that are candidate solutions are evaluated by fitness function and then are sorted. The best solutions are placed in superior set while the worst solutions that have P_α probability are placed in the inferior solution set.

In inferior update, pair of cuckoos are selected to mate with each other; then, two newborn cuckoos are generated. The newborn solutions are generated by calling Shuffle-based-LevyFlight procedure. The newborn solutions are directly substituted with parents, but in superior update, pair of cuckoos as two solutions are selected. Similar to previous section, two newborn solutions are generated. Each of newborn solutions is constituted with relevant parent provided it dominates associated parent in terms of objective function. After executing in several rounds, the best so far cuckoo is returned as an efficient solution. Shuffle-based-LevyFlight procedure in Algorithm 2 is called to explore discrete search space. The details of new Shuffle-based-LevyFlight procedure is seen in Algorithm 2.

Algorithm 1. Discrete Cuckoo Search Algorithm for SDN lifetime extension:

Input:
DSN: A given directional sensor network specification
m : number of available sensors
n : number of determined targets;
DMax : All sensor's directions;
NRng : All adjustable ranges of each sensor;
E : initial battery level of sensors;
Iterations : maximum iterations;
CuckooSize : size of all individuals in cuckoo swarm;
P_α : probability of ***inferior*** cuckoos in the cuckoo swarm;
Output:
All efficient *covers* each of which is engaged for some time;

1: *SchedulingSolution* ← ϕ ;
2: *Lifetime* ← 0 ;
3: *NPR* ← 0 ; (* number of possible rounds *)
4: For i ← 1 To *m* Do
5: E_i ← 1 ; (* The initial battery level of every sensor *)
6: End-For
7: Until *A*live sensors can observe all targets Repeat
8: Create random initial cuckoos in the *CuckooSwarm*
9: Calculate the fitness function for each cuckoo according to Eq. (6).
10:Sort the *CuckooSwarm* based on fitness function in decreasing order.
11: For *Iter*← 1 to *Iterations* Do
12: Partition *CuckooSwarm* into *inferior* and *superior* sets based on fitness function by P_α and (1- P_α) probability respectively;
13: For random pairs of cuckoos ($Cuckoo_i$, $Cuckoo_j$) in *inferior* set Do
14:[$Cuckoo_p$, $Cuckoo_q$]←*NewLevyFlight* ($Cuckoo_i$, $Cuckoo_j$, *CuckooSwarm*, *CuckooSize*) based on Algorithm 2
15: $Cuckoo_i$ ← $Cuckoo_p$
16: $Cuckoo_j$ ← $Cuckoo_q$
17: Fitness($Cuckoo_i$) ← Fitness($Cuckoo_p$)
18: Fitness($Cuckoo_j$) ← Fitness($Cuckoo_q$)
19: End-of-inferior-Changes
20: Repeat partitioning of the newly updated *CuckooSwarm* into two *inferior* and *superior* sets;
21:For random pairs of cuckoos ($Cuckoo_i$, $Cuckoo_j$) *superior* set Do
22:[$Cuckoo_p$, $Cuckoo_q$]←*NewLevyFlight* ($Cuckoo_i$, $Cuckoo_j$, *CuckooSwarm*, *CuckooSize*) based on Algorithm 2
23: Select two random cuckoos: $Cuckoo_{r1}$ and $Cuckoo_{r2}$ from *CuckooSwarm*
24: If Fitness($Cuckoo_p$) dominates Fitness($Cuckoo_{r1}$) Then
25: $Cuckoo_{r1}$ ← $Cuckoo_p$
26: Fitness($Cuckoo_{r1}$) ← Fitness($Cuckoo_p$)
27: End-if
24: If Fitness($Cuckoo_q$) dominates Fitness($Cuckoo_{r2}$) Then
25: $Cuckoo_{r2}$ ← $Cuckoo_q$
26: Fitness($Cuckoo_{r2}$) ← Fitness($Cuckoo_q$)
27: End-if
32: End-of-superior-Changes
33: End-For-Iteration
34: Sort *Cuckoos* in *CuckooSwarm* ; name the best *cover* be *BestCover*
35: *EngageTime* ← Set appropriate engagement time of *BestCover* based on the weakest sensor in terms of remaining battery.
36: *Lifetime* ← *Lifetime* + *EngageTime;*
37: *NPR* ← *NPR* +1; (* The new *cover* was constituted *)
38:*SchedulingSolution* ← *SchedulingSolution* ∪ { (*NPR*,*BestCover*, *EngageTime*) }
39:Update sensors' battery level which were engaged in current round;
40: End-Repeat-Until
41: **Return** *SchedulingSolution*, *Lifetime*, and *NPR*
42:**End-Algorithm 1**

Algorithm 2. *Shuffle-based-LevyFlight*

Input:
CuckooSwarm : Swarm of cuckoos;
SwarmSize : size of cuckoo swarm;
Cuckoo_i, Cuckoo_j : two consecutive cuckoos in *CuckooSwarm*;
n : number of targets;
m : number of sensors;
P_α : probability of *inferior* cuckoos in the cuckoo swarm ;
Output:
Cuckoo_p, Cuckoo_q : two new cuckoos;

1: Draw two random integers $R1$ and $R2$ whereas $1 \leq R1 < R2 \leq n$;
2: $\text{Cuckoo}_p \leftarrow$ Concatenate ($\text{Cuckoo}_i[1..R1]$, $\text{Cuckoo}_j[R1+1..R2]$, $\text{Cuckoo}_i[R2+1..n]$)
3: $\text{Cuckoo}_q \leftarrow$ Concatenate ($\text{Cuckoo}_j[1..R1]$, $\text{Cuckoo}_i[R1+1..R2]$, $\text{Cuckoo}_j[R2+1..n]$)
4: **Return** Cuckoo_p and Cuckoo_q ;
5: **End-Algorithm 2**

As Algorithm 2, Fig. 3, and Fig. 4 illustrate the parents chromosome is partitioned in three parts, the first newborn solution gets the first and third parts of its body from its daddy's parts while it heirs the second body part from mommy's second part. The same behavior is done for second newborn cuckoo. Fig. 3 and Fig. 4 shows how the Shuffle-based-LevyFlight procedure works.

Cuckoo_i

Target	t_1	t_2	t_3	t_4	t_5
Utilized Sensor	[1,1,2]	[1,1,2]	[3,2,1]	[4,1,2]	[5,1,2]

Cuckoo_j

Target	t_1	t_2	t_3	t_4	t_5
Utilized Sensor	[1,1,2]	[1,1,2]	[2,1,2]	[4,1,2]	[3,3,2]

Fig. 3. Two cuckoos as two candidate solutions

Cuckoo_p

Target	t_1	t_2	t_3	t_4	t_5
Utilized Sensor	[1,1,2]	[1,1,2]	[2,1,2]	[4,1,2]	[5,1,2]

Cuckoo_q

Target	t_1	t_2	t_3	t_4	t_5
Utilized Sensor	[1,1,2]	[1,1,2]	[3,2,1]	[4,1,2]	[3,3,2]

Fig. 4. Two newborn solutions

5. Performance Assessment

In this section, the performance of the proposed algorithm is assessed. To this end, some scenarios, environment description, and comparative algorithms are introduced.

A) Scenarios

Table 1 informs the considered scenarios. To compare the effectiveness of proposed algorithm, one heuristic and one meta-heuristic algorithm have been selected from literature. The former is heuristic maximum network length adjustable range (HMNLAR) [11] and genetic-based algorithm (GA) [17].

Table.1.
Scenarios Description

Scenario no.	***Number of Sensors***	***Number of Targets***
1	40	20
2	50	30
3	100	30
4	150	50

B) Simulation Environment

Table 2 shows the specification of observing environment in terms of observing area, sensor battery usage pattern, directions, etc.

Table.2.
Description of Observing Environment

Metric	***Value***
Monitoring district	1000×1000 (m^2)
Number of Targets	20~50
Number of Sensors	40~150
Max direction	1~3 (each 120^0)
Max Adjustable Range	1=(30 m), 2=(60m), 3=(90 m)
Battery Usage level	$\Delta^L = L$, L=1,2,3

C) Simulations and Data Analysis

All comparative algorithms are executed on the same datasets and running on the same platform to reach fair results. To this end, all of the literature with their proposed algorithm have been coded in MATLAB 2019a programming language. The used platform was a laptop with Intel® Core i5-3230M CPU@ 2.60GHz, 8 GB RAM memory specification and Windows 7 64-bit as an operating system. Except for *HMNLAR* that is a heuristic algorithm, others have been independently run 30 times. The minimum, maximum, and average of network lifetime are reported. To this end, Table 4 is dedicated which informs the proposed algorithm beat others in terms of lifetime expansion significantly. In addition, the relative percentage deviation (RPD) parameter is adopted from literature to show in how extent the proposed algorithm improves network lifetime in comparison with other comparative approaches [21]. Therefore, last columns' of Table 4 indicate the dominance amount of the proposed algorithm against other comparative algorithms. Since the heuristic algorithm works based on predetermined criteria, their performance does not change in different running; on the other hand, the meta-heuristic algorithms work probabilistically this is the reason why they have fluctuations. To this end, the minimum, maximum, an average values are monitored. One of the best features for deciding is to focus on average results. By a closer look at the average results, it can be concluded that the proposed algorithm has the best performance in terms of network lifetime expansion.

Table.3.
Performance assessment by comparison of different algorithms in terms of objective

Sen.	*Min-Max Lifespan in terms of unit of time (UT)*			*Average (UT)*			*RPD (UT)*	
	HMN LAR	*GA*	*DCSA*	*HMN LAR*	*GA*	*D-CSA*	*Vs HMN LAR*	*Vs GA*
1	1.55	1.65-1.95	2.10-2.35	1.55	1.77	2.22	43%	25%
2	2.43	2.47-2.63	2.50-2.68	2.43	2.53	2.58	6%	2%
3	3.05	2.95-3.15	3.14-3.60	3.05	3.09	3.40	11%	10%
4	3.55	3.57-3.85	3.95-4.47	3.55	3.72	4.23	19%	18%

Also, in terms of *RPD*, the proposed algorithm has the dominance mount of 43% and 25% in comparison with *MHNLAR* and GA-based algorithm in the first scenario. In the second scenario, the marginal dominance is about 6% and 2% respectively. In the third scenario, this dominance against other competitors is about 11% and 10%. Finally, in the larger scale input known as fourth scenario, the dominance of D-CSA is about 19% and 18% respectively in terms of network lifetime improvement. In this regard, Fig. 5 schematically depicts performance comparison of all comparative algorithms in a picture.

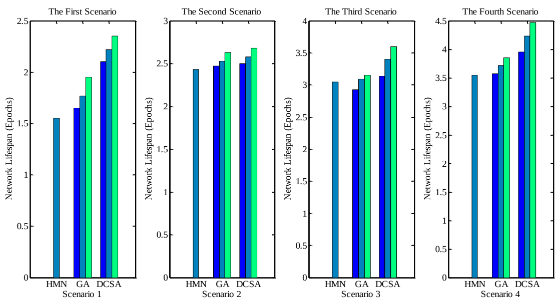

Fig. 5. The average lifetime comparison between comparative algorithms separated in each scenario

As Fig. 5 shows the proposed DCSA has the first place, after that the GA and HMNLAR are placed. Note that, the gained results of fourth scenario shows that proposed DCSA has high potential of scalability once the number of sensors and targets are significantly increased.

6. Conclusion and Future Work

This paper formulates the DSN lifetime expansion into a linear programming model (LPP) that is an NP-Hard problem. To figure out this combinatorial problem, a discrete version of CSA algorithm was extended. It takes the benefits that are designing new discrete *lévy flight* operator. This novel operator efficiently permutes discrete search space which leads to efficient solutions. The effectiveness of the proposed algorithm was verified against a famous heuristic and a successful meta-heuristic algorithm on the same datasets running in the same platform. Simulation results verified the effectiveness of proposed algorithm in solving the stated problem. For future work, we intend to extend an efficient hybrid meta-heuristic algorithm which makes balancing in local search and global search of discrete search space by incorporating machine learning techniques.

References

[1] Mortazavi, M.G., Hosseini Shirvani, M., Dana, A. et al. Sleep-wakeup scheduling algorithm for lifespan maximization of directional sensor networks: a discrete cuckoo search optimization algorithm. Complex Intell. Syst. 9, 6459–6491 (2023). https://doi.org/10.1007/s40747-023-01078-4

[2] G. Chen, Y. Xiong, J. She, M. Wu and K. Galkowski, "Optimization of the Directional Sensor Networks With Rotatable Sensors for Target-Barrier Coverage," in IEEE Sensors Journal, vol. 21, no. 6, pp. 8276-8288, 15 March15, 2021, doi: 10.1109/JSEN.2020.3045138.

[3] Ekhlas, V. R., Hosseini Shirvani, M., Dana, A. et al." Discrete grey wolf optimization algorithm for solving k-coverage problem in directional sensor networks with network lifetime maximization viewpoint," in Applied Soft Computing, vol. 146, no. 110609. https://doi.org/10.1016/j.asoc.2023.110609.

[4] H. Zannat, T. Akter, M. Tasnim, A. Rahman, "The coverage problem in visual sensor networks: A target oriented approach," journal of network and computer applications, vol. 75, pp. 1-15. 2016. http://dx.doi.org/10.1016/j.jnca.2016.08.015.

[5] J. Jia, C. Dong, Y. Hong, L. Guo, Y. Yu, "Maximizing full-view target coverage in camera sensor networks," Ad Hoc Networks, Vol. 94, no. 101973, pp. 1-10, 2019, https://doi.org/10.1016/j.adhoc.2019.101973.

[6] P. Singh and S. Mini, "A Heuristic to Deploy Directional Sensor Nodes in Wireless Sensor Networks," 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), 2016, pp. 11-15, doi: 10.1109/CICN.2016.11.

[7] W. M. Tan, S. A. Jarvis, "Heuristic solutions to the target identifiability problem in directional sensor networks," in journal of network and computer applications, vol. 65, pp. 84-102, 2016, https://doi.org/10.1016/j.jnca.2016.02.011.

[8] J. Xu, F. Zhong, Y. Wang, "Learning Multi-Agent Coordination for Enhancing Target Coverage in Directional Sensor Networks", 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

[9] S. Sharmin, F. N. Nur, M. Islam, M. A. Razzaque, M. M. Hassan and A. Alelaiwi, "Target Coverage-Aware Clustering in Directional Sensor Networks: A Distributed Approach," in IEEE Access, vol. 7, pp. 64005-64014, 2019, doi: 10.1109/ACCESS.2019.2916407.

[10] H. Ahmad N. Kohil, "Bi-directional sensor placement for K-coverage problem," in International Journal of Advanced Intelligence Paradigms, Vol. 23, no, 1-2, pp 186–202, 2022. https://doi.org/10.1504/ijaip.2022.125241.

[11] H. Mahamadi, S. Salleh, M.N. Razali, "Heuristic methods to maximize network lifetime in directional sensor networks with adjustable sensing ranges," in journal of network and computer applications, Vol. 46, pp. 26-35, 2014, https://doi.org/10.1016/j.jnca.2014.07.038.

[12] A. Al Zishan, I. Karim, S. Saha Shubha, A. Rahman, "Maximizing heterogeneous coverage in over and under provisioned visual sensor networks" journal of network and

computer applications, vol. 124, pp. 44-62. 2018, https://doi.org/10.1016/j.jnca.2018.09.009.

[13] D. Arivudainambi , R. Pavithra & P. Kalyani, “Cuckoo search algorithm for target coverage and sensor scheduling with adjustable sensing range in wireless sensor network,” Journal of Discrete Mathematical Sciences and Cryptography, vol. 24, no. 4, pp.975-996, 2021. DOI:10.1080/09720529.2020.1753301.

[14] M. Tanha, M.S. Hosseini Shirvani, A.M. Rahmani, “A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments,” Neural Comput Appl, vol. 33, pp.16951–16984. 2021. https:// doi. org/ 10. 1007/ s00521- 021- 06289-9.

[15] Abedpour, K., Hosseini Shirvani, M. & Abedpour, E. A genetic-based clustering algorithm for efficient resource allocating of IoT applications in layered fog heterogeneous platforms. Cluster Comput (2023). https://doi.org/10.1007/s10586-023-04005-x.

[16] A. Javadian Kootanaee, A. Poor Aghajan, S.M. Hosseini, Ahybrid model based on machine learning and genetic algorithm for detecting fraud in financial statements, J. Optim. Ind. Eng. 14 (2) (2021) 180–201, https:// doi. org/ 10. 22094/ joie. 2020. 18774 55. 1685.

[17] A. Alibeiki, H.Motameni, H. Mohamadi, “A new genetic-based approach for solving k-coverage problem in directional sensor networks,” in Journal of Parallel and Distributed Computing, vol. 154, pp. 16–26, 2021. https://doi.org/10.1016/j.jpdc.2021.03.006.

[18] M. Hosseini Shirvani, “An energy-efficient topology-aware virtual machine placement in Cloud Datacenters: A multi-objective discrete JAYA optimization,” in Sustainable Computing: Informatics and Systems. 2023. vol. 38, no. 100856 .https://doi.org/10.1016/j.suscom.2023.100856.

[19] Seyyedamin Seifhosseini , Mirsaeid Hosseini Shirvani , Yaser Ramzanpoor , Multi-objective Cost-aware Bag-of-Tasks Scheduling Optimization Model for IoT Applications Running on Heterogeneous Fog Environment, Computer Networks (2023), doi: https://doi.org/10.1016/j.comnet.2023.110161.

[20] X.-S. Yang; S. Deb (December 2009). Cuckoo search via lévy flights. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009). IEEE Publications. pp. 210–214. arXiv:1003.1594v1.

[21] M. Hosseini Shirvani, “A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems,” in Eng. Appl. Artif. Intell., vol. 90, no. December 2019, p. 103501, Apr. 2020, doi: 10.1016/j.engappai.2020.103501.