

RESEARCH

Open Access



Computational intelligence-based connectivity restoration in wireless sensor and actor networks

Solmaz Mohammadi and Gholamreza Farahani*

* Correspondence: farahani.gh@irost.org

Department of Electrical Engineering and Information Technology, Iranian Research Organization for Science and Technology (IROST), Tehran, Iran

Abstract

Network failure is categorized into the two types of software and hardware (physical layer) failure. This paper focuses on the physical layer failure in the wireless sensor and actor networks (WSANs). Actors play an important role in data processing, decision-making, and performing appropriate reactions. Single or multiple nodes failure of actors due to the explosion, energy depletion, or harsh environments, can cause multiple disjoint partitions. This paper has proposed a new computational intelligence-based connectivity restoration (CICR) method. It uses a combination of advanced computational intelligence methods to solve restoration problem. The proposed algorithm applies the novel enhanced Lagrangian relaxation with a novel metaheuristic sequential improved grey wolf optimizer (SIGWO) search space algorithm in simultaneous selection of k sponsor and p pathway nodes. The reactive proposed method aims to reduce the travel distance or moving cost and communication cost. As a result, the restored network has minimum of topology change and energy consumption. In terms of total traveled distance, CICR has 37.19%, 71.47%, and 44.71% improvement in the single-node failure averagely in comparison with HCR, HCARE, and CMH, respectively. Also, it has an average of 61.54%, 40.1%, and 57.76% improvement in comparison with DCR, PRACAR, and RTN in multiple partitions resulted from multiple nodes failure, respectively. The reliability of CICR method has improved averagely by 35.85%, 38.46%, 22.03% over HCR, CMH, and HCARE in single-node failure. In multiple nodes failure, reliability of CICR has averagely 61.54% and 20% over DCR and PRACAR, respectively.

Keywords: Wireless sensor and actor networks, Restoration, Travel distance, Distributed optimization process, Partitioning

1 Introduction

Wireless sensor and actor networks (WSANs) refer to a set of sensors and actors that communicated by wireless medium to perform sensing and acting [1]. In this network, sensors sense the humidity, temperature, chemical parameters, and other environmental parameters, then analyze and even calibrate the information. The actors such as robots make decisions, forward the messages to sink or sink to sensors, and then perform appropriate actions upon routing, failures, and communications between equipment. Applications of these networks are monitoring and collecting data from

the battlefield, environment, and human health body. The latest applications of WSNs are smart home, smart city, smart grids, industrial internet, smart farming, and cloud computing [2]. Stability of connectivity and coverage in these networks is crucial. Therefore, connectivity restoration approaches in wireless sensor networks (WSNs) and WSNs have attracted significant attention.

Due to energy depletion, hardware malfunction, failure of nodes in WSNs, or explosion in a harsh and hostile environment, one or multiple actors may fail and results partitioning the network. A network partitioning is the reduction of a graph to a smaller graphs by segmenting its nodes into the exclusive groups, that each group with the other group has no communication and data exchange. This failure can disconnect a part of nodes and make them unreachable from their adjacent nodes. Thus, restoring connectivity repairs the failures which can be due to hardwares such as sensors and actors malfunction. This paper ignores software failures or traffic crashes in routers.

After the wide explosion, several actors and nodes will fail at the same time. In connectivity restoration process, there should be a comprehensive algorithm that can repair two or more created partitions of single or multiple failures. In order to retain the connectivity under the failure of some nodes, the node redundancy method has been proposed by deploying more nodes than necessary [3]. But redundancy of nodes has overhead cost of hardware. In addition, it is not possible to replace the actor in the harsh or impassable places. Therefore, in the recent researches, one of the neighbors of actors with respect to distance, degree, or other parameters will replace with the failed actor. In these methods, the topology and routing of the network will change.

Connectivity restoration must be intelligent, distributed approach and automatic self-healing process. Performance of restoration algorithms depends on the energy efficiency, scalability, node density, network recovery time, or time complexity of algorithms for online applications and communicational complexity.

The less travel distance has a less energy consumed by the network. In harsh environments, the nodes are equipped with solar cells in addition to the battery [4]. Energy saving is very important. Therefore, energy consumption during restoration must be low. The minimum total energy consumed prolongs the lifetime of the network. Therefore, an energy-efficient approach to solve this issue is necessary. Optimal energy-efficient algorithm is an algorithm with minimum traveled distance by all nodes and minimum message overhead. Also, the fewer relocated nodes, cause the less topology change, and the sense of critical area information is more covered.

This paper proposes comprehensive, intelligent, distributed, hybrid, localized, and energy-efficient algorithm that can repair all kinds of failures. Unlike proactive mechanisms, reactive methods can restore any type of failures, whether single or multiple failures by evaluating the number and the location of the failed nodes. The goal of this method is to decrease the movement and communication overhead during the restoration process.

The rest of the paper has been organized as follows. Section 2 reviews related literature respect to the two categorizations including proactive and reactive methods. Section 3 formulates the problem of restoring and describes the network model. Section 4 introduces the proposed method in detail. Section 5 describes the theoretical analysis of the algorithm. The simulation setup and results are presented in section 6. Finally, section 7 expresses the conclusions and future work.

2 Related works

A node failure can create multiple partitions in a WSN. The presented strategies for connectivity restoration can be categorized into the proactive and reactive restoration [5]. In the proactive strategies, the problem of restoration can be solved by using redundant actors [6]. As well as the backup node for each actor is also chosen at the network setup time to control the network message overhead and replacement during the process [7]. In the other proactive methods, k -connected topologies will generate [8]. k -connected topologies are included k paths between two nodes. Proactive methods put a lot of resources cost overhead on the network, in contrast, they will not interrupt the network.

Many metaheuristic approaches have been proposed for solving the proactive relay placement problem to restore connectivity such as genetic [9], artificial bee colony [10], particle swarm optimization-based [11] algorithms. Uwitonze et al. [12] has proposed connectivity restoration in wireless sensor networks via space network coding (RPSNC) based on relay placement. This algorithm studies network coding in Euclidean space and places excess relay nodes at the network. Deployment of redundant relays and actors cannot be an optimal solution due to time complexity, and it is not possible to use in the combat areas and rough paths. Therefore, these processes replaced with relocation of existing nodes by self-restoration techniques with non-extensive damages. In reactive strategies, the process relocates neighbors. Reactive approach distributed actor recovery algorithm (DARA) [13] has been used for single-node failure with multiple partitions. DARA selects the proper node among the two-hop failed node's neighbors with considering degree and distance. Selected node moves to the location of the failed node. If the moved node causes the partitioning of the network; therefore, the algorithm will execute iteratively until the whole of the network is connected. Like as DARA algorithm, Younis et al. [14] proposed the recovery through inward motion (RIM) algorithm to minimize messaging overhead and reduce the distance traveled of neighbor nodes. RIM initiates a local recovery process by relocating all the single-hop neighbors of the failed node toward it, until it is within its transmission radius. Other disconnected nodes have cascaded inward motion to connect with the network. Unnecessary neighbor movements cause high energy consumption and increase message overhead. As well as, the cascaded inward motion will decrease the coverage of the network. To save energy by reducing relocation overhead, an energy-efficient method, intelligent on-demand connectivity restoration for wireless sensor networks (IDCRWSN) [15] has been presented which considers the energy level of the participant nodes in the process. The adjacent node will broadcast a recovery message and will increase its transmission radius instead of moving toward the failed node.

Hybrid connectivity restoration (HCR) combines proactive and reactive restoration mechanisms for single-node failure [16]. In the proactive process, an actor selects a backup node through its single-hop neighbors based on the minimum travel cost and distance. The reactive process includes cascaded nodes movement. In this method, the backup selection and restoration process start only for critical actors. The node whose failure breaks its directed adjacency connectivity is named a critical node. The HCR is designed for restoring two partitions and cannot recover connectivity of more disjoint network. Another weakness of this algorithm is that it checks the moving cost for backup assignment in each recovery step and it has no overall vision and causes of the

cascading relocations and higher moving cost, as well as, the cascaded motions will decrease the coverage of the network and not suitable for large-scale networks. For this reason, Zhang et al. has suggested a hybrid optimization scheme on connectivity restoration processes for large scale networks that both the backup selection and the relocation process are reactive. This method is a hybrid coverage-aware perception-based connectivity restoration (HCARE) algorithm [17]. The goal of HCARE method, which is a hybrid method, in addition to increasing coverage and network reliability, is to reduce the total traveled distance. In this method, the overlapping function will calculate. Then with respect to the overlapping function and degree of nodes, the node that leaves the least loophole in its location with the lowest degree, will be selected as the backup node. This selected node is one of the neighbors of the failed node and reduces the total traveled distance. The optimal location for the backup node is the place where it maintains the best network coverage with minimum overlapping coverage after network connectivity recovery. This method has fewer cascading motions and will be repeated until a non-critical node selects as the backup.

To maintain the robustness of the topology against failure and reducing the total traveled distance, a distributed k -connectivity maintenance and restoration algorithm for heterogeneous (CMH) WSNs has been presented [18]. After a node failure, a neighbor of the failed node sends a move message to actors with minimum moving cost to the location of the failed node. The objective of CMH is reducing of moving cost. In this method, failure occurs when k decreases. In single-node failure $k = 1$ can be defined for critical nodes. CMH uses the basic idea of distributed Dijkstra's algorithm [19] for finding the paths with minimum moving. After selection of an actor with minimum moving cost, the move message is forwarded by the other nodes until it reaches the selected actor which can increase the number of exchanged messages. Proactively selection of critical and backup nodes for multiple nodes failure has been carried out by Imran et al. [20]. They proposed a distributed connectivity restoration algorithm (DCR) for restoring multiple partitions of single and multiple nodes failure. DCR proactively detects critical actors and considers non-critical backups for them based on distance and node degree. If the backup is also a critical node, cascaded relocation of backups will be carried out. The authors also proposed an extended version of DCR named recovery algorithm for the failure of multiple nodes failure (RAM), for restore connectivity of two adjacent nodes failure. The goal of this method is reduction of the restoration time and overhead. Backup movement to the same position of the failed node is drawback of this scheme and grows the total moving cost.

Permanent relocation algorithm for centralized actor recovery (PRACAR) and self-route recovery algorithm (SRRA) also solves the single-node failure resulting multiple partitions restoration problem [21]. The PRACAR replaces the failed centralized actor by its neighbor. PRACAR selects reactively the best actor node for substitution in terms of distance from the failed node, lowest cluster sensors, and lowest degree of nodes that covers lowest area. Then, it applies SRRA that finds the optimal path for the remaining neighboring sensors of relocated actors. The moved nodes in this algorithm are low but the total travel cost is not minimum. Also, the messaging overhead is large.

A method with low data exchange overhead has been suggested for cluster based restoration process which clusters are unaware of their own size, surviving nodes, and

links as well as size and location of other survivor clusters while the nodes are equipped with GPS. This algorithm solves the problem of restoration of connectivity in mobile wireless sensor networks after multiple simultaneous nodes failure named round-table negotiation (RTN) approach [22]. In this method, each cluster sends a negotiator to participate in a round table negotiation and decision-making process. They exchange information and find paths for connecting clusters through known dead node locations and then select the nodes to be moved to those paths. Negotiators return to their clusters and restoration begins. The relocation of nodes in this method increases the energy consumption and the total traveled distance. Another method of collocated multiple nodes failure is the restoring connectivity in a resource constrained WSN (RCR) [23] algorithm that restores connectivity by relocating nodes to act as relays. If the existing nodes are insufficient, many of selected nodes will be used as mobile data collectors with optimal tours to decrease latency. The recovery goal of RCR is moving the fewest nodes and reducing the moving cost.

One of the researches in scattered failures is the distributed autonomous connectivity restoration method based on finite state machine (DCRMF) that employs a regional restoration [24]. Finite state machine identifies whether a node is a critical or non-critical. If a failed node is a critical node, at the same time, the nearest non-critical nodes will be relocated to replace the scattered failed nodes. The goal of this algorithm is decreasing the movement overhead.

In section 4, the problem will be formulated as constrained optimization problem. Related literature has solved this problem by metaheuristic approaches. Little information requirement about the problem, search capability in very large spaces of candidate solutions, reliable and robust performance, and easy implementation are the advantages of these algorithms [25]. Wang [26] applied an effective co-evolutionary particle swarm optimization (CPSO) for constrained engineering problems. PSO is used with two kinds of swarms for exploration and exploitation in spaces of solutions. The proposed CPSO is population based. Huang et al. [27] has proposed a differential evolution approach based on a co-evolution differential evolution mechanism (CDE) to solve the constrained problems. To handle the constraints, a special penalty function is introduced. Then, a co-evolution model is presented and differential evolution (DE) is used to perform evolutionary search in spaces of both solutions and penalty factors interactively. Brajevic [28] proposed a crossover-based artificial bee colony (ABC) algorithm. To enhance the exploitation of ABC, two modified ABC agents are used in employed and onlooker phases, and crossover operator is used in scout phase instead of random search. These methodologies do not find the optimal solution and will often find the near-optimal solutions for the constrained problem.

Compared to other population-based algorithms, grey wolf optimizer (GWO) has easy implementation and few parameters to set [29]. GWO outperforms in terms of the optimality of the results, efficiency, robustness, and convergence speed. It provides optimal results in most cases compared with other metaheuristic algorithms. As a result, to solve the connectivity restoration problem in this paper, a novel metaheuristic SIGWO is applied to find optimal solution in search spaces of candidate nodes in subsection 4.2.2. A summary of the classical GWO algorithm is given in subsection 2.1.

2.1 Classical GWO algorithm

The GWO algorithm is a new meta-heuristic approach inspired by the social hierarchy of the grey wolves. GWO imitates the behavior of the grey wolves in terms of the leadership hierarchy and hunting mechanism in nature. Grey wolves such as alpha (α), beta (β), delta (δ), and omega (ω) are assumed for presenting the leadership hierarchy [30]. Delta wolves must take orders from alpha and beta, but they dominate the omega. In GWO algorithm, search process can be performed by wolf sets as solutions that deployed in search area randomly [29]. The main steps of grey wolves hunting are including searching prey, tracking, encircling, and then attacking [31]. The mathematical model of GWO and novel method stated at subsection 4.2.2.

3 Network model and problem description

3.1 Network model

In this paper, a WSN composed of ordinary and stationary sensor nodes via movable actor nodes. Actors have more processing capabilities and larger transmission radius than sensors. Sensor nodes have limited energy and computational unit, that sense data from environments. Actors in a WSN are gateways to sink. Sensors send their data to actors through multi-hop paths and then actors forward the data to the sink. Sink nodes process or analyze the data and send processed data to the control center or transmit commands from the control center to other nodes with respect to the IDs of nodes. Sensors and actors are randomly deployed in the sensing area of the environments. Assumed network in this paper have mobile actors and low powered stationary sensors. Sensors periodically update their status like residual energy and their location information to actors and exchange this information with adjacent nodes. Some nodes and actors that are in the transmission radius of each other can communicate together.

Each actor node just keeps and updates a single-hop adjacency table including information such as neighbor's locations, node ID, and criticality. Communication between multi-hop adjacent nodes is performed by exchanging single-hop adjacency information.

In the distributed methods, computations and decision-making will carry out locally. One advantage of the distributed methods is that fewer messages are transmitted over the network. Distributed computations like distances are carried out by their own neighbor actors. Master actor (MA) proactively has been introduced for each segment at the network setup time for performing centralized main computations of the proposed algorithm and finding an optimal solution. MA has control over the information and positions of all the nodes in the partition.

To define the assumptions and conditions of the algorithm, the following definitions are described.

Definition 1 (multi-hop adjacency): A WSN will be represented as an undirected Graph, $G = (V, E)$, where V defines set of vertices, E defines a set of edges and $E = \{e_{ij} | (i,j) \in V\}$ defines bidirectional communication between nodes in the network. N_i^m is the multi-hop adjacency set of i , if and only if there is at least one path between i and j with multiple hop. j is the multi-hop adjacent of i .

Definition 2 (criticality): The critical node might be cut-vertex node or strategic node. A cut-vertex is a critical actor in the network topology and failure of it breaks direct connectivity and divides the network into the multiple segments [32]. In

addition, in this paper, another definition of the criticality of a node is defined. The node that is located in the region of great importance is named strategic node, such as flood-prone areas. This metric is determined by the control center. A node is a non-critical if and only if it is removed, all its neighbors have a connectivity path. The criticality of each node is determined through exchanging information of its single-hop adjacency table.

Other CICR algorithm assumptions are in the following:

- Sponsor and pathway nodes are selected from actor nodes.
- The transmission radius is R_c . To strengthen the connectivity between actors, both in the CICR and the baseline approaches, the distance between adjacent actors is less than or equal to R_c .
- Transmission radius of actors is the same. Actors are homogeneous. An actor can communicate and forward messages and commands of the arbitrary number of actors.
- Until the network is fully connected, actors are static and when failure occurred and the recovery process started, actors can move.
- There will be no new failures during the process. If the new failure occurs, it will be repaired after the end of the first process.
- Single-node failure can lead the network to the two or multiple partitions.
- Actors are divided into the critical and non-critical categories. If a non-critical actor failed, the cost of restoration is zero and process is not initiated.
- Nodes are equipped with GPS and geographical coordinates of all nodes are assumed to be known.

3.2 Problem description and actor failure problem

In a WSN, actors act as a gateway to base station. The number of actors is less than sensors. They can manage the network to be energy-efficient, since an actor calculates the optimal path between its connected sensors, and it can avoid the interference. Furthermore, it has a vital role in medium access control (MAC) layer of the network. The actor failure either disconnects its connectivity with its connected sensors and with other actors. Therefore, actor's failure is more important than the failure of the sensors. Due to the critical actor failure resulting multiple disjoint partitions, parts of the connected actor networks will lose their connectivity with the rest of the network. Therefore, the sensing performance of the network drops significantly.

In the steady state, actors exchange heartbeat messages to determine liveness of the links. With the heartbeat messages exchanging, each actor notifies its neighbor's failure locally and informs to MA or sink. Nodes select single-hop neighbor to send heartbeat message to monitor it. If the sender node does not receive the ACK (acknowledgement) of the message from its neighbor after predefined time, it declares neighbor node as a failed node and forwards this announcement to the MA to carry out CICR algorithm computations.

The ACK message contains the survival time for the ACK, that is, the time to live (TTL). The value of TTL minus 1 is when an ACK passes one checkpoint node. Checkpoint nodes are neighbors of a nodes in the heartbeat routing path from the neighbors

to the source node in the ACK-based scheme. As long as a checkpoint node receives a data packet, it will return an ACK to the upstream node. If the value of TTL in an ACK message is 0, then the ACK message is discarded. After the heartbeat message is forwarded by the node, it then waits for the arrival of the ACK packet. If the node has not yet received the expected number of ACK messages, it will send an alert message to the source node and then the source node declares the failed node and sends the alert message to the master actor for running CICR algorithm to restore connectivity. For example, when using $TTL = 2$, if some neighbor nodes do not receive 2 ACK packets, they will send an alert message to the source node to warn that there is a failure [33]. Nodes receive heartbeat messages based on the performance and duration of its sleep and active modes that are defined by the control center.

If an actor fails, there must be existed at least one path between all actors in the network. In this case restoring connectivity has not cost. Restoration initiates for critical actor.

The proposed method is explained with details in the next section.

4 Proposed method

The proposed method in this paper named CICR restores connectivity after partitioning of the network in a WSN. This method is the hybrid of proactive and reactive restoration process. Proactive part includes the MA assignment. The CICR uses a reactive algorithm to select the relocating nodes and move them to the designated locations which is described.

4.1 Reactive k sponsor and p pathway nodes selection

When single or multiple critical nodes failed, CICR relocates $k = p-1$ number of sponsor nodes to joint p pathway nodes to minimize the travel cost of the relocated sponsor nodes, where p is the number of partitions. A pathway node has been selected from each partition so that other nodes can communicate with each other through this node and the sponsor node. The sponsor node can move to establish a connection.

This method is a hybrid kind of advanced computational intelligence or soft computing. Soft computing, unlike traditional computing, includes approximate models and solves complicated real-life problems. In order to reduce the computational and time complexity, the enhanced Lagrangian relaxation method can be used and a metaheuristic search method or sequential improved grey wolf optimizer (SIGWO) finds more accurate solutions that are described in subsection 4.2.2.

Unlike the HCR and HCARE that are used for restoring the connectivity of two partitions resulted from single-node failure, DCR, PRACAR, and RTN can repair failure of multiple partitions in single or multiple adjacent node failure. In addition to restoring a single-node failure, CMH can also be extended to solve multiple nodes failure. After occurring multiple failures in CMH, the neighbors of the failed nodes start the restoration simultaneously. For multiple nodes failure, DCR can be run more times. CICR is applied in both single and multiple nodes failure. The CICR selects simultaneously sponsor and pathway nodes for connecting created partitions in order to minimize the total traveled distance of the moved sponsor nodes in the single operation. CICR detects sponsor nodes for connecting pathway nodes of partitions, even if pathway nodes

are in multi-hop adjacency. In other words, it selects a pathway node from each partition in order to minimize the total moving cost of moved sponsor nodes. Sponsor nodes are moved to establish a connection between pathway nodes.

According to Fig. 1, CICR determines pathway and sponsor nodes through the algorithm mentioned in subsection 4.2. Suppose two nearest nodes of two disjoint partition, as shown in Fig. 1a, are $A1$ and $A2$. One solution determines $A1$ and $A2$ as pathway nodes. As well as, $A3$ is selected as a sponsor node. This is one solution that is found by CICR among existing solutions as can be seen in Fig. 1b, the optimal position for sponsor node $A3$ is $A3'$. In the second solution as shown in Fig. 1c, $A1$ and $A3$ are determined as pathway nodes and $A2$ is selected as the sponsor node, which relocates to the optimal location $A2'$ for restoring. The dashed line in Fig. 1 b and c shows the total distance moved. The total distance moved by sponsor node in Fig. 1c is lesser than in Fig. 1b.

Generally, the CICR does not select the p nearest pathway nodes of the partitions. It selects p pathway and k sponsor nodes that decrease the total traveled distance of all sponsor nodes. CICR intelligently assumes the nodes $A1$ and $A3$ as the pathway nodes and $A2$ as the sponsor node. In dense and semi-dense WSANs, CICR selects only this combination for sponsor and pathway nodes. This combination is one of the reasons why this approach is better than HCR, CMH, and HCARE and other approaches like DCR, PRACAR, and RTN. These selected nodes generate a pathway for communicating partitions. The rest of the nodes communicate together through these nodes in the path with an optimal routing. The total moving cost is the distance moved by a single sponsor node. As a result, in multiple disjoint partitions, the distance moved by all sponsor nodes is the total moving distance or failure cost.

For two disjoint partitions, the proposed algorithm restores the connectivity through a single actor given the distance between them if it is not more than $2R_c$. Otherwise, it estimates the distance to restore connectivity with the lowest number of sponsor nodes and the least moving cost.

Theorem 1: *Best position for sponsor node must be on the intersection line of sponsor and farthest pathway node and R_c far away from farthest pathway or non-adjacent pathway node.*

Proof. In this algorithm, if one of the pathway nodes is non-critical, the sponsor node will definitely be in the adjacency of non-critical pathway node. The robust neighborhood condition for the two nodes is to be located at distance R_c from each other. Sponsor node goes to the best position and since it is in the transmission radius of the pathway node, the connection will be established. The best position of sponsor node to restore connectivity between two pathway nodes $A1$ and $A3$ in Fig. 2, must be in the $A2'$. $A1$ is the farthest pathway node.

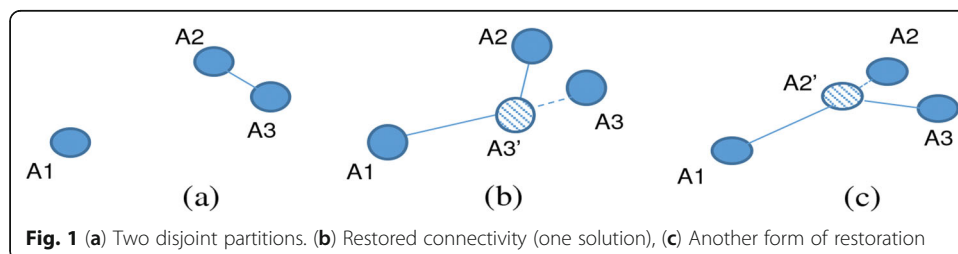


Fig. 1 (a) Two disjoint partitions. (b) Restored connectivity (one solution), (c) Another form of restoration

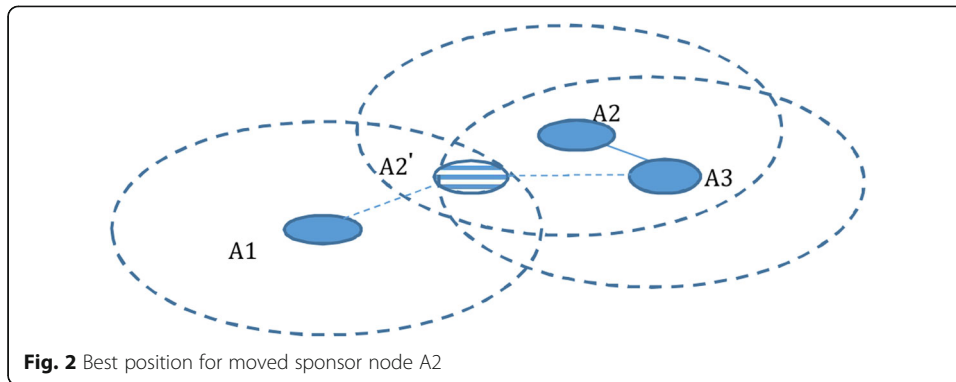


Fig. 2 Best position for moved sponsor node A2

The HCR selects properly the best position in some cases and in most cases, it is the intersection of communication circles of any two nodes. In CMH, DCR and PRACAR, the backup or selected node moves to the position of the failed node. In the best case, the total moving cost of all backups is the nearest adjacent distance. HCARE selects the location of moved backup nodes based on the overlap of the transmission radius of nodes in the failed area. Destination location can be exactly the intersection of two nodes and does not have more overlap with one of the nodes, which increases the total traveled distance. RTN does not accurately identify boundary nodes (described in the subsection 6.3.1) and increases the total traveled distance. Therefore, the CICR outperforms HCR, CMH, HCARE, DCR, PRACAR, and RTN in terms of total distance moved.

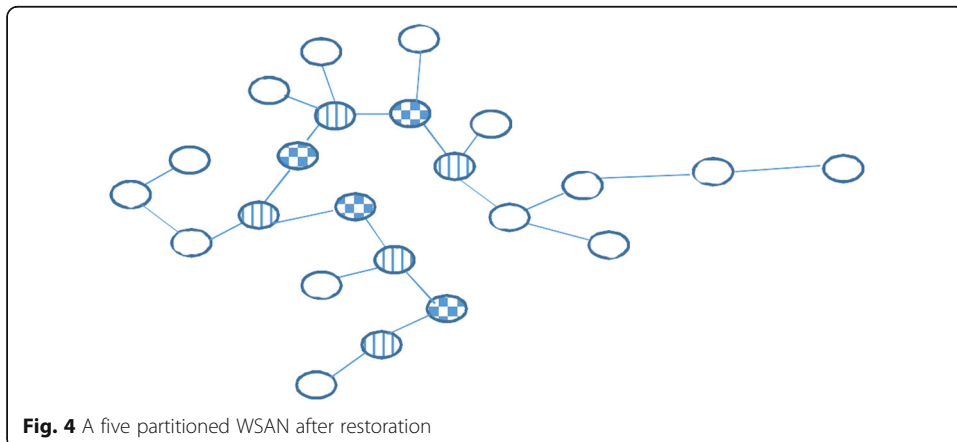
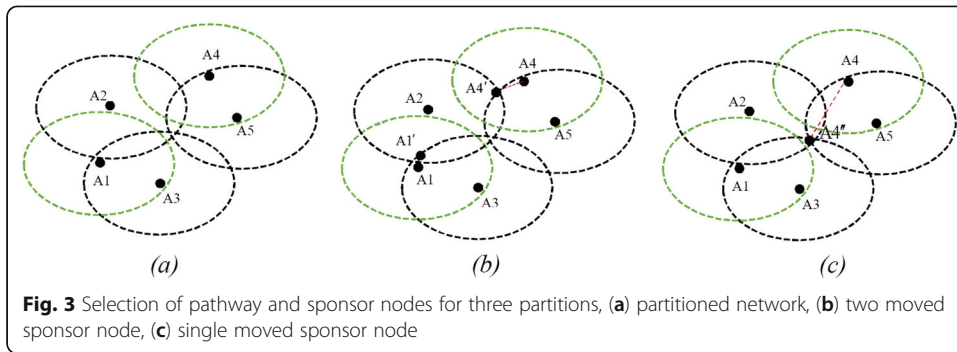
Theorem 2: *The minimum moving cost of CICR is when the sponsor nodes will be located between two partitions after moving unless the pathway nodes are not in the single or two-hop neighborhood of the failed node.*

Proof. In Fig. 3, the transmission radius of sponsor nodes is denoted by dashed line and three nodes A2, A3, and A5 are pathway nodes of three disjoint partitions. Figure 3b shows the solution that has the minimum total traveled distance and two sponsor nodes moved to restore connectivity ($d_{A1,A1'} + d_{A4,A4'}$). Therefore, it causes the low energy consumption and little topology change. The second solution in Fig. 3c is the case that a single sponsor node moves to the intersection of three transmission radii and restores the connection of three partitions (total traveled distance is $d_{A4,A4'}$). The total traveled distance in the latter case in Fig. 3c is more than the first case in Fig. 3b, i.e., $d_{A1,A1'} + d_{A4,A4'} < d_{A4,A4'}$. The proposed algorithm intelligently considers all of these solutions and ultimately selects the first solution.

If the distance between two nodes is more than $2R_c$, more than one sponsor node is essential to restore the connectivity between two pathway nodes. Figure 4 shows the five partitioned networks after restoration with CICR method. It has five pathway and four sponsor nodes.

Efficient area coverage fulfilled after restoration, because of the robust topology form in this algorithm.

CICR comprises of three nested algorithms 1 to 3. The algorithm 1 is the pseudo-code of the restoration process.



Algorithm 1: Restoration process of CICR algorithm

R_c : The transmission radius for each actor

1: initialize single-hop adjacency table for each actor and selection of MA for clusters or segments.

2: read ID of failed actor

3: neighboring nodes send alert message (containing announcement of failure and failed actor IDs) to the MA

4: **for** every failed_ID in a WSAN do

5: **if** Is_Critical (failed_ID) do

6: read the number of partitions, generating groups for each partition, pathway and sponsor node candidate sets: input for Algorithm2

7: **running of Algorithm2: Pathway and sponsor nodes selection**

8: obtain sponsor and pathway nodes ID from algorithm2 for coverage of these partitions

9: **else**

10: restoring_cost = 0

11: **end if**

12: **end for**

13: MA sends a message to notify sponsor nodes to move to the best position

14: **while** a sponsor node receives a movement message

15: sponsor nodes add best position into its adjacency table and pathway nodes to update its adjacency table

16: sponsor nodes go to best position (Best_Pos) according to the Theorem 1.

17: **end while**

18: **output: restored WSAN**

In dense and large WSANs as well as in large-scale damage, assignment of the k sponsor nodes to p pathway nodes, between single or multi-hop adjacent nodes of the failed nodes and minimizing the total cost is hard. To solve this problem, if the number

of candidate nodes for sponsor and pathway nodes are n , and finally $p + k$ of them are selected, the problem will be n^{p+k} -hardness problem. This problem is well-known as a NP-hard problem. Due to the NP-hard nature of this problem, exact methods involve intense computation. Therefore, usage of enhanced Lagrangian method and SIGWO mathematical relations for NP-hard problem solution has been explained in subsection 4.2.

4.2 Mathematical computations

4.2.1 Enhanced Lagrangian method

To reduce the time complexity of the problem, an enhanced Lagrangian relaxation and a novel metaheuristic optimal search algorithm has been proposed to find the exact and best solution for solving the actor failure recovery problem.

Let $I = \{1, \dots, m\}$ is the candidate set of pathway nodes containing single-hop and two-hop neighborhood of the failed node's adjacents. Assume vector $J = \{1, \dots, n\}$ is the candidate set for sponsor nodes. The nodes in I and J are denoted by i and j indices. (i, j) denotes the edges between vertices i and j , so that $i \in I, j \in J$. Since simultaneous pathway and sponsor node selections, the candidates set of pathway and sponsor nodes are the same.

The output pathway number of the algorithm will be equal to the number of partitions. k is the maximum number of sponsor nodes that can either interconnect the partitions or can have minimum total traveled distance as it is proved in Theorem 1. Selecting k sponsor and p pathway nodes with minimum cost and determining best locations for sponsor nodes are output of algorithm 2. The nodes in a partition are grouped into the same groups and after selecting the nodes, it checks whether they are in a same group, if they are, other nodes will be selected instead of them. These groups are denoted by $\{P_r | r = 1, \dots, p\}$. Where p is the number of created partitions. Assume the definitions of variables in Equations (1) and (2) as below.

$$a_{ij} = \begin{cases} 1 & \text{if sponsor } j \text{ is connected to pathway node } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$S_j = \begin{cases} 1 & \text{if a node } j \text{ is sponsor} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

a_{ij} is a decision variable that determines whether a sponsor node j is assigned to a pathway node i . d_{ij} is defined as the cost of selecting pathway node i to a sponsor node j , therefore, $d_{ij} > 0$ for all $i \in I$ and $j \in J$. If a sponsor node to be assigned to three pathway nodes, there exist three d_{ij} with three different i and single j . The sum of the three values of d_{ij} is the total cost for sponsor node j . The total cost of assignment k sponsor nodes for p pathway nodes is the sum of all these costs in one solution. The total cost of reestablishing of the failed network is as Equation (3).

$$R = \min \sum_{i=1}^m \sum_{j=1}^n d_{ij} a_{ij} \quad \forall i \in P_r \& i + l \notin P_r, r = 1, \dots, p, l = 1, \dots, m - i \tag{3}$$

where R is the minimum total cost of selecting p pathway and k sponsor nodes. P_r represents nodes belong to the same partition. This function R is called the objective function, subject to Equation (4).

$$\sum_{j=1}^n S_j = k \tag{4}$$

where k is the total number of required sponsor nodes. Considering the problem that each pathway node is connected to the two different sponsor nodes as shown in Fig. 4, the following relation (5) exists for the decision variable a_{ij} :

$$\sum_{j=1}^n a_{ij} = 2 \quad \forall i = 1, \dots, m \tag{5}$$

Another condition specifies if a decision variable a_{ij} is equal to 1, corresponding S_j is also 1 and j will be the sponsor for pathway node i . It is defined as Equation (6).

$$x_{ij} \leq S_j, \forall i = 1, \dots, m, j = 1, \dots, n \tag{6}$$

$D_{m \times n}$ is a matrix with minimum Euclidean distance elements d_{ij} between all candidates.

The equations (4, 5 and 6) are constraints on the variables that aim to find the connected i and j that will be achieved the minimum cost R . Enhanced Lagrangian has been used to solve it. Lagrangian relaxation gives a vector of Lagrangian multipliers. Generally, Lagrangian relaxation is the sum of the objective function and constraints and Lagrange multiplier λ [34]. Equations (1) and (2) transform to the Equations (7) and (8) to generate the Lagrangian relaxation.

$$a_{ij} \in \{0, 1\}, \forall i = 1, \dots, m, j = 1, \dots, n \tag{7}$$

$$S_j \in \{0, 1\}, \forall j = 1, \dots, n \tag{8}$$

The constraint in Equation (5), is relaxed with objective function. Then, a Lagrangian relaxation is obtained as Equation (9).

$$R = \max_{\lambda_i} \min_{a_{ij}, S_j} \sum_{i=1}^m \sum_{j=1}^n d_{ij} a_{ij} + \sum_{i=1}^m \lambda_i \left(2 - \sum_{j=1}^n a_{ij} \right) \tag{9}$$

Subject to Equations (4), (6), (7), (8).

where λ_i is a vector of Lagrangian multipliers. Equation (9) is equivalent to Equation (10).

$$R = \max_{\lambda_i} \min_{a_{ij}, S_j} \sum_{i=1}^m \sum_{j=1}^n (d_{ij} - \lambda_i) a_{ij} + 2 \sum_{i=1}^m \lambda_i \tag{10}$$

Subject to Equations (4), (6), (7), (8).

The relation (10) and constraint (6) are unified into the Equation (11).

$$a_{ij} = \begin{cases} S_j & \text{if } d_{ij} - \lambda_i \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The Lagrangian (10) is changed into the Equation (12).

$$\begin{aligned}
 R(\lambda) &= \min_{S_j} \sum_{j=1}^n \sum_{i=1}^m \phi_{ij} S_j + 2 \sum_{i=1}^m \lambda_i \\
 \text{Subject to } &\sum_{j=1}^n S_j = k \text{ and (8)}.
 \end{aligned}
 \tag{12}$$

where vector $\lambda_i = [\lambda_1, \dots, \lambda_m]$ and $\phi_{ij} = \min(0, d_{ij} - \lambda_i)$ with considering relation (11). S_j is the unknown variable that must be determined in such a way minimize $R(\lambda)$. The desired S_j are related to the k number of minimum $\sum_{i=1}^m \phi_{ij}$.

For solving the problem, first, the search space of Lagrangian multipliers has been determined. In the earliest approaches λ_i can be found by decomposition $R(\lambda)$ into the m maximization problems of equation (13) and is performed a random search for λ .

$$\max_{\lambda_i} R(\lambda_i) \cong \sum_{j=1}^n \min(0, d_{ij} - \lambda_i) + \lambda_i \quad \text{for } i = 1, \dots, m
 \tag{13}$$

In the proposed CICR, a Lagrangian relaxation (12) is used for finding the optimal solution of Equations (3) to (8). A SIGWO search has been carried out in the space of the Lagrangian multipliers λ to find the λ vector with respect to Equation (11) and the optimal solution set of Equation (4) is $\{j \in J \mid S_j = 1\}$.

CICR determines a lower and upper bound for Lagrangian multipliers with respect to (13). $R(\lambda_i)$ will be maximized if λ_i is in the range between $d_{i[1]}$ and $d_{i[2]}$, as Equation (14).

$$\lambda_i \in [d_{i[1]}, d_{i[2]}], i = 1, \dots, m
 \tag{14}$$

$d_{i[1]}, d_{i[2]}$ are the first and second smallest minimum element in the i th row of the cost matrix D respectively.

Theorem 3: $R(\lambda_i)$ will be maximized if $\lambda_i \in [d_{i[1]}, d_{i[2]}]$ for $i = 1, \dots, m$.

Proof. Because for $\lambda_i = d_{i[2]}$, $R(\lambda_i) = \sum_j \min(0, d_{ij} - d_{i[2]}) + d_{i[2]}$ where $d_{ij} \geq d_{i[2]}$, $\min(0, d_{ij} - d_{i[2]}) = 0$, therefore, $R(\lambda_i) = \min(0, d_{i[1]} - d_{i[2]}) + d_{i[2]} = d_{i[1]}$. It is the smallest in row i . Thus, $\lambda_i = d_{i[2]}$ maximizes $R(\lambda)$. Then, assume $\lambda_i = d_{i[1]}$, $R(\lambda_i) = \sum_j \min(0, d_{ij} - d_{i[1]}) + d_{i[1]}$. When $\min(0, d_{ij} - d_{i[1]}) = 0$ for all j , $R(\lambda_i) = d_{i[1]}$ which is optimal.

In this paper, corresponding pathway and sponsor nodes i, j , and Lagrangian multipliers λ_i are determined by solving the Lagrangian problem (12) due to the upper and lower bound for λ_i . Then, SIGWO is used for finding $\{j \in J \mid S_j = 1\}$.

Therefore, H is a $m \times 2$ matrix that has $d_{i[1]}, d_{i[2]}$ for each row $i = 1, \dots, m$, that is defined as Equation (15).

$$H = \begin{bmatrix} d_{1[1]} & d_{1[2]} \\ d_{2[1]} & d_{2[2]} \\ \vdots & \vdots \\ d_{m[1]} & d_{m[2]} \end{bmatrix}
 \tag{15}$$

The pseudo-code for k sponsor and p pathway nodes selection for p disjoint partitions is shown in algorithm 2.

Algorithm 2: pathway and sponsor node selection

inputs:

$k=p-1$, p number of partitions, \vec{I} and \vec{J} vectors

Generate $m \times n$ cost matrix D

output:

R , and assignments a_j and S_j for all i and j ,

Steps:

1: Generate H matrix according to the cost matrix

2: Generate initial $\vec{\lambda}$ matrix with random λ_{α} , λ_{β} , λ_{δ}

3: Running Algorithm 3: The SIGWO search method to find the optimal solution

4: Select best position (Best_Pos) for every sponsor

Return the node's ID of sponsor and pathway nodes which has the minimum traveled distance

A summary description of the GWO algorithm and finding an optimal solution with proposed SIGWO is introduced in subsection 4.2.2.

4.2.2 Finding optimal solution with SIGWO

The aim of using this algorithm is finding the fittest Lagrangian multipliers and assignments S_j for all i and j in problem of (12). Generally, the GWO has not been used to restore the connectivity of actors in the WSANs. In GWO algorithm, to mathematically simulate social and hunting behavior of them, the three best solutions are considered to be alpha (most optimal and best solution), beta (second best solution), and delta (third best solution) that have better knowledge about the potential location of the prey. The remainder of the solutions are supposed as Omega. The initial Lagrangian multiplier vector λ_i is a random population of the grey wolves. The optimal location of the prey is unknown. During iterations, alpha, beta, and delta wolves estimate the value of the prey. The objective function is Equation (12). Each solution calculates the value of the objective function and acquires vector $\{j \in J | S_j = 1\}$. If the final solution is true, then relation (4) will satisfy.

The mathematical encircling behavior of grey wolf due to [29] has been proposed in Equations (16, 17 and 18). The first three best solutions obtained are saved and the other search agents must update their positions according to the position of the best solutions as Equations (16, 17 and 18):

$$\vec{D}_{\alpha} = \left| \vec{C}_1 \cdot \vec{X}_{\alpha} - \vec{X} \right|, \vec{D}_{\beta} = \left| \vec{C}_2 \cdot \vec{X}_{\beta} - \vec{X} \right|, \vec{D}_{\delta} = \left| \vec{C}_3 \cdot \vec{X}_{\delta} - \vec{X} \right| \tag{16}$$

$$\vec{X}_1 = \vec{X}_{\alpha} - \vec{A}_1 \cdot (\vec{D}_{\alpha}), \vec{X}_2 = \vec{X}_{\beta} - \vec{A}_2 \cdot (\vec{D}_{\beta}), \vec{X}_3 = \vec{X}_{\delta} - \vec{A}_3 \cdot (\vec{D}_{\delta}) \tag{17}$$

$$\vec{X}(t+1) = (\vec{X}_1 + \vec{X}_2 + \vec{X}_3)/3 \tag{18}$$

where $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ are the position vector of the three best solutions. \vec{X} indicates the position vector of a grey wolf. t indicates the current iteration. For random initialization of them, values for $\vec{A}_1, \vec{A}_2, \vec{A}_3, \vec{C}_1, \vec{C}_2,$ and \vec{C}_3 are selected according to the Equations (19 and 20).

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{19}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{20}$$

where \vec{A}, \vec{C} are coefficient vectors, \vec{r}_1, \vec{r}_2 are random vectors in $[0, 1]$, and \vec{a} is linearly decreased from 2 to 0 in each iteration. In classical GWO, exploitation is ensured by parameter \vec{a} . Teng [35] has investigated the improvement of the exploration process in GWO by decreasing the value of \vec{a} according to a nonlinearly function as given in Equation (21).

$$\vec{a} = \vec{a}_{initial} \cdot (1 - \mu \cdot (iter/iter_{max})^2) \tag{21}$$

where $\mu \in (0, 3)$ is a nonlinear modulation index, $iter$ denotes current iteration and $iter_{max}$ is the maximum number of iterations. $\vec{a}_{initial}$ is the initial value of the control parameter \vec{a} . Based on results of [35] for $\mu \in (1, 1.5)$, to improve exploration process and robustness of searching, μ is set to 1.3, resulting in the number of iterations used for exploration is 70% and the number of iterations of exploitation is 30%.

In the iterative period, when $|\vec{A}| < 1$ is the encircling area will shrink, and the candidate solutions converge toward the value of the prey. In every iteration, the best three solutions will save and oblige the search agents to update their positions in the search space and converging to optimal locals of the solution space. This is the reason of fast convergence of GWO to find Lagrangian multipliers.

In SIGWO, for each row $i = 1, \dots, m$, there exists three best solutions for λ_i . The SIGWO runs m times to find the smallest value for Equation (12) and acquires all pathways and sponsor nodes. The three best solutions of SIGWO for λ vectors are in the following matrix (22).

$$\lambda = \begin{bmatrix} \lambda_{1\alpha} & \lambda_{1\beta} & \lambda_{1\delta} \\ \lambda_{2\alpha} & \lambda_{2\beta} & \lambda_{2\delta} \\ \vdots & \vdots & \vdots \\ \lambda_{i\alpha} & \lambda_{i\beta} & \lambda_{i\delta} \end{bmatrix} \tag{22}$$

The pseudo-code of the proposed SIGWO is shown in algorithm 3.

Algorithm 3: The SIGWO search method

- 1- initialize parameters a, A, C
 - 2- initialize the prey wolf population λ_i ($i=1, \dots, n$) due to the lower and upper bounds
 - 3- three random initializations of grey wolves (three candidate solutions)
 - 4- **while** ($t < \text{Max number of iterations}$) or $iter_{max}$
 - 5- **for** ($i=1:1:n$)
 - 6- form $(m \times 3)$ λ matrix
 - 7- Update $(\lambda_{\alpha}), (\lambda_{\beta}),$ and (λ_{δ}) (Equation (18))
 - 8- **end for**
 - 9- calculate the objective function of each grey wolf (search agents) $R(\lambda)$ according to Equation (12)
 - 10- update \vec{a} using (21), \vec{A} and \vec{C}
 - 11- $t = t + 1$
 - 12- **end while**
 - 13- **return best solution** $(\lambda_{\alpha}),$ sponsor nodes $\{j \in J \mid S_j = 1\}$ and corresponding pathway nodes
-

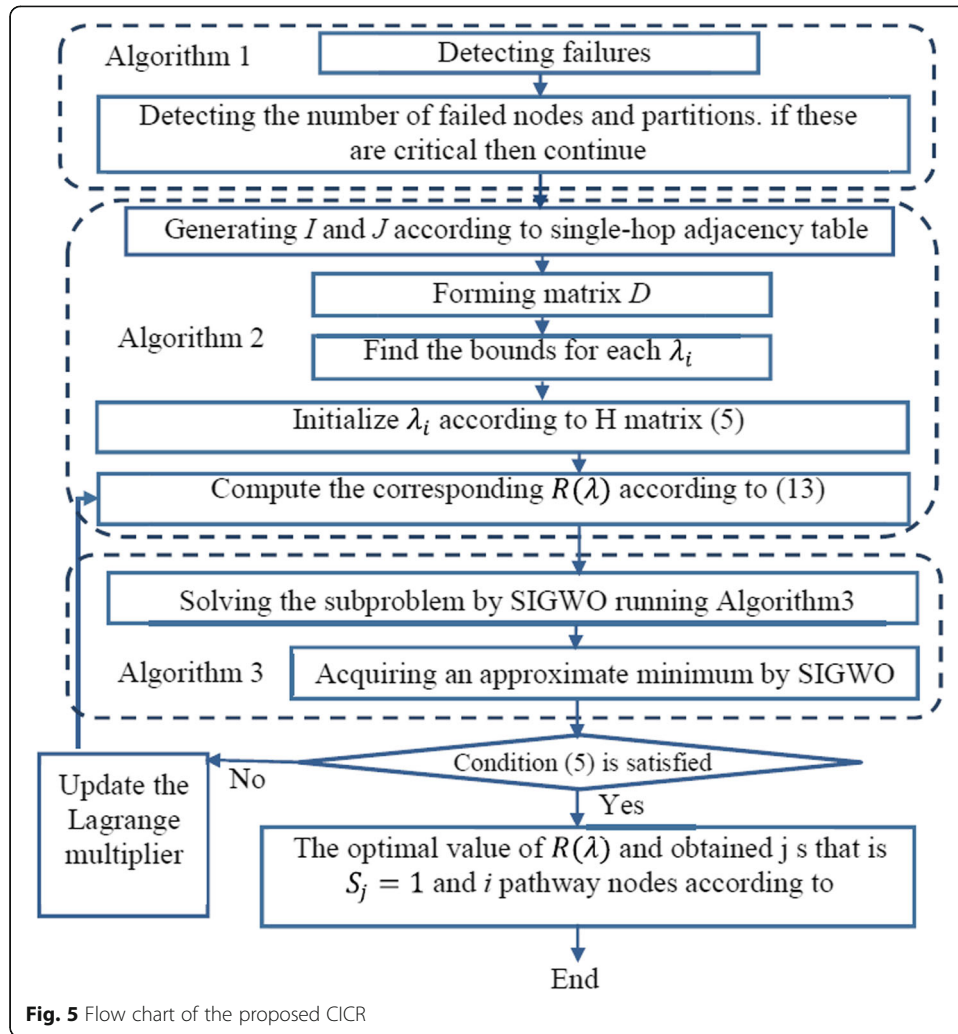
The overall hierarchical flowchart of the proposed CICR algorithm is presented in Fig. 5.

In the sparse network that has more critical nodes, the probability of choosing a critical node as a sponsor node will be high and it results from recursive process more often. In order to avoid an unfinished loop, each moved sponsor node can only relocate once in the case of cascading motion.

5 Analysis of CICR algorithm

This novel proposed method can improve the performance in terms of time and computational complexity specially in the multiple partitions or large network size. The recovery time of this algorithm is very low, in contrast, in HCR and DCR, assigning the backup and failure cost for all critical nodes in dense networks with more than 200 nodes, is time-consuming and practically impossible for very dense networks. The major disadvantage of Dijkstra’s algorithm in CMH is the fact that it does a blind search by consuming a lot of time waste of necessary resources. Dijkstra’s algorithm has a lot of memory wasting [36].

At continuation, the performance (subsection 5.1) and complexity (subsection 5.2) of the CICR will be analyzed.



5.1 Performance analysis

To analyze the performance of the CICR, the following theorems are introduced.

Theorem 4: *CICR can restore connectivity of single or multiple nodes failure in collocated or non-collocated created partitions in a process execution.*

Proof: CICR is a reactive process because selection of sponsor and pathway nodes for multiple failure needs to detect the number of partitions and make a decision. The centralized main computations are performed in the MA. MA estimates scale of failure and carries out the computations of any failure. Considering this issue that the pathway and sponsor nodes are selected all at once, calculations can be carried out with the parallel computing using FPGAs (field programmable gate arrays).

Theorem 5: *In the worst case, the maximum number of moved nodes in the CICR equals to $k = p - 1$, where p is the number of partitions.*

Proof: The number of partitions is equal to the number of pathway nodes. In the worst case, one sponsor node is required to bridge two pathway nodes that each one is representative for one partition. In this way, all failures and partition connections are restored. Theorems 1 and 2 can confirm this statement.

Theorem 6: *In the CICR, the maximum distance moved by one sponsor node, is equal to the transmission radius R_c .*

Proof: In this approach, a sponsor node is the neighbor of one of the pathway nodes. sponsor node moves to the best position that mentioned in Theorem 1. In the worst case, a sponsor node moves to the location of the failed node. Also, each sponsor node is relocated once, so the maximum relocation of sponsor is equal to R_c . Because the maximum distance between the neighboring nodes is R_c .

5.2 The computational and time complexity analysis of the CICR

In the CICR method, to reduce the time and computational complexity, enhanced Lagrangian method is used to speed up finding optimal solution using a convergence approach, also the SIGWO has been used instead of a stochastic search. Enhanced Lagrangian relaxation keeps the optimality as well as decreasing the computational complexity.

In addition, the search space of Lagrange multipliers is reduced with determining the lower and upper bounds of each λ_i . The time complexity of the best position calculation for sponsor nodes is constant. Given a $R(\lambda)$ in Equation (12), y_j is determined with definition of vector λ_i and a SIGWO search is performed to find Lagrangian multipliers. This sentence $2\sum_{i=1}^m \lambda_i$ in Equation (12) defines the overall time complexity of the CICR. So, the complexity of the problem is just equal to the time complexity of the SIGWO. λ matrix in Equation (22) has m number of rows, that m is the number of members of vector I . According to algorithm 3, the *for* loop runs m times in each iteration and the outer *while* loop runs $iter_{\max}$ times. The time complexity of the overall algorithm is $O(iter_{\max} \times m)$. $iter_{\max}$ for sparse networks which is very small.

6 Simulation results and discussion

The performance of CICR is compared to HCR, CMH, HCARE, DCR, PRACAR, and RTN methods. The performance metrics and setting of the simulation are presented in subsection 6.1. Analysis and obtained comparison results are explained in subsections 6.2 and 6.3. The reliability of CICR in comparison with HCR, DCR, and PRACAR is evaluated in subsection 6.4. Limitations of this study are mentioned in subsection 6.5.

6.1 Simulation setup and performance metrics

Two kinds of scenarios have been simulated using the MATLAB R2019b software. First, scenario is the case that single-node failure causes two partitions and the performance of CICR has been compared with HCR, CMH, and HCARE.

In the first scenario that is described in subsection 6.2, the number of deployed actors or nodes (DN) is varying from 20 to 140 with an increment step of 20 when $R_c = 80$ as assumed in [16]. Then, the transmission radius (R_c) of all nodes is the same. The value of R_c is chosen from 20 to 120 with changing 20 step each time when the number of nodes is fixed. All the sensor nodes are randomly deployed over a network size of $1000 \text{ m} \times 1000 \text{ m}$.

In the second evaluation that will describe in subsection 6.3, the performance of CICR is compared with the DCR, PRACAR, and RTN methods. In these topologies, more than two partitions, i.e., six partitions resulted from single or two adjacent failures

are evaluated. In addition to all the above conditions and random topologies, DN is varying from 25 to 200 with an increment step of 25 when R_c is fixed at 100 like as [20, 21]. Then, R_c is varying from 25 to 200 with an increment step of 25 when $DN = 100$. All the sensor nodes are randomly deployed over a network size $600\text{ m} \times 600\text{ m}$. When nodes fail, for each network, all methods include CICR, DCR, PRACAR, HCR, CMH, HCARE, and RTN are run and evaluation metrics are computed for these methods.

In order to obtain the correct average results, for the same DN and R_c , 15 random WSANs are created in both scenarios and all of the deployed critical nodes are randomly failed. Also, all results are the average over these random simulations.

In the following, four metrics that are used to evaluate the performance of CICR has been described.

- Number of relocated actors: This parameter represents the overall number of actors participating in the restoration process. According to this parameter, whatever the number of relocated actors is fewer, then the higher efficiency of the algorithm will be resulted.
- Total traveled distance: It represents the sum of the total distance moved by all the relocated nodes during the restoration process which evaluates the efficiency in terms of overhead and energy consumption. This parameter that is called moving cost must be minimized.
- The number of exchanged messages: It indicates the total number of exchanged messages during restoration and evaluates the energy consumption and overhead.
- Reliability: Reliability of a connectivity restoration method describes the proper operation of the restored area without a second failure and partitioning. As reliability increases, the network becomes more resistant to the second failure.

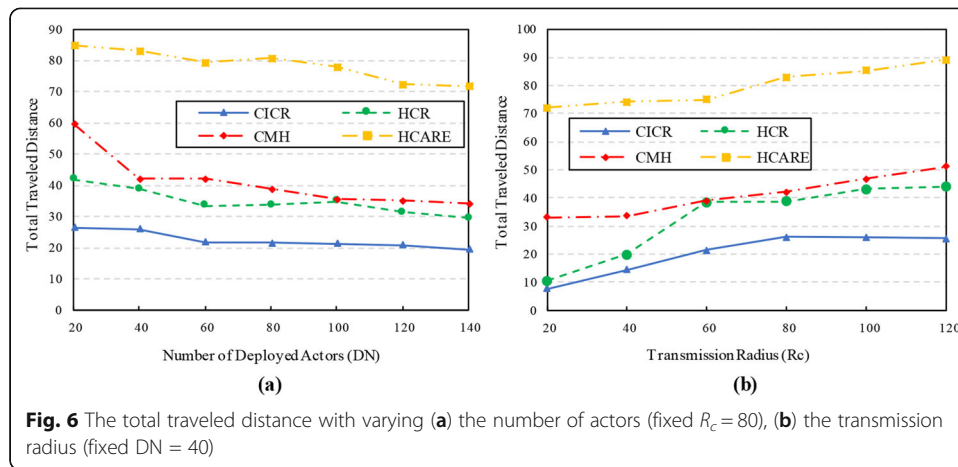
6.2 Single-node failure results and discussion

In this section, the CICR has been compared with HCR, CMH, and HCARE in terms of total traveled distance, number of relocated actors, and number of exchanged messages.

6.2.1 Total traveled distance

As shown in Fig. 6, CICR outperforms CMH, HCR, and HCARE in all values of DN and R_c . Figure 6a shows the change of varying DN on the total traveled distance with fixed $R_c = 80$. HCR has many cascaded movements which result in more travel cost than CICR. CICR generally converges the restoration process toward a higher density of region. The total traveled distance reduces with increasing DN and when $DN > 60$, remains almost constant and linear. In general, the total traveled distance decreases during increasing DN. Because for the same R_c , if DN grows, then the density gets higher. As a result, the area covered by the nodes increases. In this case, the CICR incorporates more nodes, and then selects pathway and sponsor nodes between them that minimizes the total traveled distance of all moved sponsor nodes.

The curve trend almost is the same and with increasing DN drops just a little. The linearity of CICR is more than other methods. In HCR, a sparse WSAN encounters large traveled distance compared to a dense WSAN, since deployment of nodes is more



linear, and this leads to the long inter-node distance and cascading movements. For non-critical actors, cost will be zero in both HCR and CICR. In the simulations of HCR in [16], both critical and non-critical actors have randomly failed, and values are calculated with having zeros. But in this paper, just failure of critical actors is considered. Therefore, the curve of HCR is nearer to CICR. According to Table 1, the CICR method has a 37.19% improvement over the HCR averagely.

In the simulations for CMH algorithm $k = 1$, and like the other algorithms, if a node is critical and loses all its connections, failure is declared. The purpose of this method, in addition to keeping the stable connections, is to reduce the moving cost. As shown in Fig. 6a, in lower DN values, the moving cost is higher because the number of critical nodes is higher and CMH in 1-connectivity mode does not select the backup nodes from the critical nodes. Because each critical node in this method must have at least one connection to another node. Therefore, the critical node does not move in this method. In this case, a non-critical node is selected from a distance that does not disconnect the connection of the critical nodes, which increases the total traveled distance relative to the CICR. As the number of nodes increases, the number of non-critical nodes decreases. Selected node moves along the shortest graph path toward the location of the failed node based on Dijkstra’s algorithm. This movement along the graph path, in contrast to the Euclidean path, increases the moving cost. As a result, one of

Table 1 The total traveled distance with varying the number of actors (fixed $R_c = 80$)

Total traveled distance							
Number of deployed actors	HCR (m)	CMH (m)	HCARE (m)	CICR (m)	Improvement over HCR (%)	Improvement over CMH (%)	Improvement over HCARE (%)
20	42.84	59.78	84.93	26.40	38.36%	55.84%	68.92%
40	39.762	42.12	83.1	26.048	34.49%	38.18%	68.66%
60	34.59	44.38	79.33	21.77	37.06%	50.95%	72.56%
80	34.646	38.76	80.8	21.504	37.93%	44.53%	73.39%
100	35.865	35.49	78	21.325	40.54%	39.93%	72.67%
120	32.392	35.03	72.4	20.954	35.31%	40.19%	71.06%
140	30.592	34.23	71.8	19.386	36.63%	43.38%	73.01%
Average	35.81	41.39	78.62	22.48	37.19%	44.71%	71.47%

the limitations of selecting a node in CMH is that it must have a path to the failed node. Refer to Table 1, CICR has 44.71% improvement over CMH.

By obtaining the overlapping function of the node transmission radius, HCARE examines the total traveled distance and proves that if the overlapping function in dense networks in the failed area is higher, then the total traveled distance will be shorter. Backup node from the neighboring of the failed node is selected in such a way that it has less overlap in its area and less degree. The selected backup node goes to a place where the overlapping function of the communication range of the neighboring nodes is minimized. This location can be exactly the intersection of two nodes and does not overlap with one of the nodes, which increases the cost of relocation. This method, like HCR, has cascading movements for critical backup nodes which increases the cost of moving. As shown in Table 1, the total traveled distance in the CICR method has 71.47% improvement over HCARE.

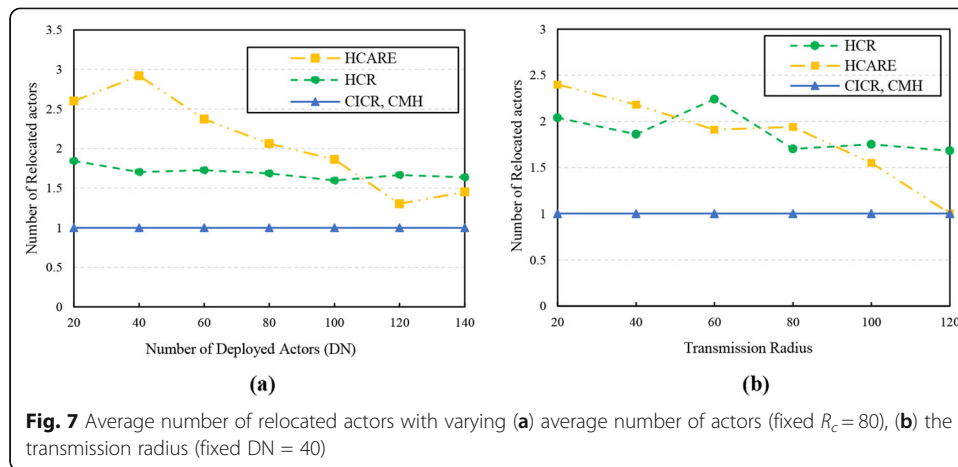
Figure 6b shows, CICR, HCR, CMH, and HCARE rise with the increasing the R_c . As can be seen, in CICR the total traveled distance increases with the transmission radius of $R_c < 80$. Because with the increasement of R_c , the inter-node distance will be long, therefore, the traveled distance grows until reach to 80 and coverage will grow in $R_c > 80$. Then for $R_c > 80$, the increasing R_c does not affect the distance traveled and almost will be stable.

In small $R_c < 60$, because of the short inter-node distance, HCR selects many nodes to move with minor movements. Therefore, the curve of HCR is nearer to CICR. In CMH, the total traveled distance in $R_c < 60$ increases significantly compared to the CICR method, because when the transmission radius is low, the number of critical nodes is higher. Therefore, the distance that a non-critical node needs to travel is longer. In $R_c > 60$, in CMH and HCARE as well as HCR, as the R_c increases, the inter-node distance will increase too, which the total traveled distance increases.

6.2.2 Number of relocated actors

According to Theorem 2, $k = p-1$ number of relocated actors can minimize the cost of moving nodes. In sparse networks, relocated actors during restoration is more than dense networks. Whatever the number of relocated actors is smaller, then the topology will change less, and the less critical area information will be lost during restoration.

Figure 7 shows the average number of relocated actors with increasing DN and R_c . The two curves of CICR demonstrate that the CICR moves only one sponsor node to restore two partitions, because the number of relocated actors depends on the number of generated partitions, while HCR moves more actors than the CICR. In the dense networks, in HCR, sometimes the number of moved nodes can reach to six nodes. Also, in the sparse networks, can reach to four nodes in one restoration process and has more moving overhead. In this case, since at higher density the number of neighboring nodes is higher, it is more likely to select backup which will break a new connectivity by HCR. These cascade motions increase the movement overhead of the network. But as shown in Fig. 7, HCR moves more nodes in low R_c and DN, because of the low distance between actors and low degree of nodes respectively. HCARE, like HCR, has cascading relocations. In a low-density network, there are more critical nodes and in HCARE, the coverage loophole must be repaired, therefore, the number of relocated actors



increases. Gradually, as the number of deployed actors increases, the coverage will increase, and the number of relocated actors decreases (as shown in Fig. 7a). In CMH, exactly one node is selected and goes to the location of the failed node. This node is a non-critical node and CMH has no cascading relocations.

In Fig. 7b, for low R_c , internode-distance is lower and decreases the number of relocated actors. R_c and DN has no effect on the number of nodes moved in the CICR.

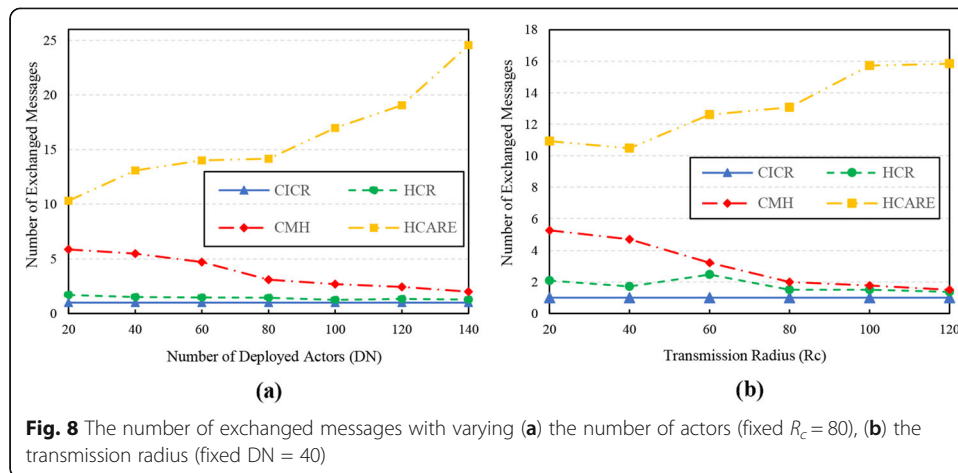
Figure 7 confirms that the CICR outperforms HCR, HCARE and is equal to CMH in terms of the number of relocated nodes. The plotted results are the average over 15 random simulations.

6.2.3 Number of exchanged messages

The total exchanged message or communication cost consumes resources, from the aspect of time, bandwidth, and hardware. Figure 8 shows the relationship of the total exchanged messages with DN and R_c in average. The CICR outperforms the HCR, CMH, and HCARE in terms of the exchanged messages with increasing both DN and R_c . As shown in Fig. 8, the total exchanged messages remain constant with CICR as DN increases. Message overhead in this case can divide into two phases. One step before restoration and another is during the restoration. Message overhead during the restoration is important and can effect on energy consumption and recovery time.

Proactively selection of MA reduces message overhead during the restoration. The MA sends the new coordinates of sponsor nodes to them. In two partitions, a single sponsor node connects two pathway nodes. Therefore, a message has been sent from MA to the sponsor node and reactive motion starts. In this case, the message overhead equals to one in all DN and R_c .

The exchanged message between neighbors at a new position of sponsor nodes has not been considered since it is the ordinary status update for single-hop adjacency table. Before the restoration process, the nodes maintain single-hop adjacency table. So, it also reduces the message overhead. The CICR has superiority than HCR on the exchanged message overhead. In the curve of Fig. 8a, R_c has been fixed at 80 m and the number of messages is shown with the increasing of DN. While this curve for CICR is constant, the HCR curve decreases almost with the growth of DNs. In HCR, the



number of messages in sparse WSANs is larger than dense WSANs. It is related to the high possibility of the cascading node motions.

Of course, the large number of moved nodes can exist in the dense networks. Multiple moved backup nodes grow the total messages overhead.

As shown in Fig. 8a, in CMH, when the non-critical nodes and paths to that nodes become larger in the dense network, the closest path to the failed node will be selected. The number of exchanged messages depends on the number of nodes along the selected graph path. As much as, the path is shorter, then the number of move messages is smaller. In this method, overhead messages are counted during restoration. Move message is forwarded by a neighbor of the failed node until it reaches an actor node that has the minimum moving cost. Each node that forwards the message to the next node, adds its location information to the message. Thus, after receiving a move message, the mobile node will have a complete path to the location of the failed node.

The number of exchanged messages in HCARE is $4L + 2$, where L is equal to the neighbors of the failed node. As the number of deployed actors increases, the degree and the neighbors of the failed node (L) increases. As a result, the number of messages increases with the number of deployed actors as shown by the HCARE method curve in Fig. 8a.

In the large R_c , the HCR and HCARE curves drop since the inter-node coverage is long and moved backups decreases as shown in Fig. 8b. In the case of CMH, large R_c results fewer and farthest non-critical nodes for movement which increases the path of the message to the destination node and increases the number of messages.

6.3 Multiple collocated node failures results and discussion

In the second scenario, 40% of the network is randomly selected and has been failed. Single or two adjacent nodes are assumed which can divide the network into the multiple disjoint partitions. In the simulations, the number of disjoint partitions is six. Proposed CICR is compared with DCR, PRACAR, and RTN in terms of the total traveled distance, the number of relocated actors, and the number of exchanged messages.

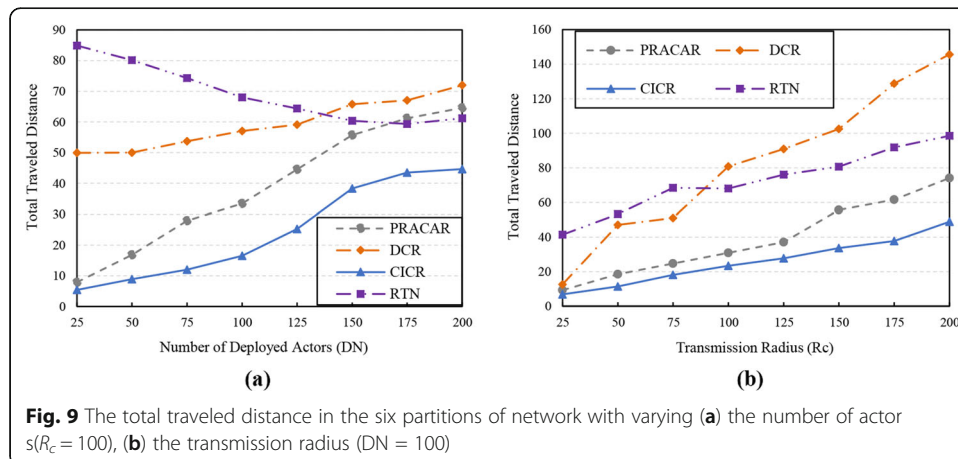
6.3.1 Total traveled distance

Figure 9 shows the total traveled distance of the moved nodes with changes in R_c and DN. As illustrated in Fig. 9, the CICR outperforms the DCR, PRACAR, and RTN. The PRACAR chooses a single non-critical actor for relocation even if it has a large total travel distance from the failed node. In most cases, there is no relocation of several nodes resulting in minimum moving cost. As a result, this method is not efficient in harsh environments, also it increases energy consumption. In DCR, the changing DN, has a little effect on the performance of the total traveled distance but as shown in Fig. 9a, the curve line will not be constant. According to Fig. 9a, CICR has a slow growth when $DN > 150$ and almost has a constant trend. Both DCR and PRACAR relocate the selected actor for relocation to the position of the failed actor.

In RTN approach, the negotiators of each cluster travels to the round table, then back to the cluster and estimates the final deployment destination of the relocated node. Negotiators move to the round table where the decision of reconnection paths and replacement nodes are made. In this method, the selected nodes do not move to the location of the failed nodes and are placed in the direct paths that exist between the partitions. These routes are selected by choice of path with random draw. One of the disadvantages of this method is that during selecting partition boundary nodes, the nodes that have lost one or more single-hop neighbors are identified as the boundary nodes. If the closest nodes are in the two partitions, they can be border nodes, but they are in the other side of the partition and cannot be recognized as the border nodes in RTN. Sometimes changing the connection of two partitions to a location that is not in the neighborhood of the failed node will significantly reduce the moving cost. For $DN > 150$, RTN has a more linear curve and even a slight increased trend because the distance between the partitions decreases due to the network density.

According to Theorem 1, all sponsor nodes in CICR relocate to closest positions such a way that results in the lowest total distance moved by mobile actors and lowest energy consumption.

The CICR curve trend is slow as shown in Fig. 9b and the DCR has the most total traveled distance with regards to the increasing R_c . In CICR, optimal sponsor nodes for joining pathway nodes rarely affect other nodes and rarely results in restoration again. Affected area by relocation of the sponsor nodes is low and this is another reason for



the minimal distance moved of the CICR. As shown in Fig. 9b, in RTN for higher R_c , as the transmission radius increases due to the long inter-node distance, the partitions are farther from each other and the number of nodes to create six partitions is greater. Therefore, the total traveled distance increases by the relocated nodes.

According to Fig. 9, the CICR presents the best performance. Based on the values of Table 2, the CICR method improved by 40.1%, 57.76%, and 61.54% compared to the PRACAR, RTN, and DCR, respectively.

6.3.2 Number of relocated actors

The number of relocated actors is evaluated on the issue of disjoint six partitions with growing of DN. Figure 10 shows the number of relocated actors with varying DN. It is true that the number of relocated nodes is less in DCR for $DN < 125$, but in $DN > 150$ the CICR outperforms the DCR, PRACAR, and RTN. The scalability of the CICR will be higher. The number of relocated actors is directly related to the number of partitions in the CICR. But in $DN > 150$, the density of the network will be higher and sometimes, selecting pathway and sponsor nodes from the multi-hop neighborhood of the failed node can reduce the number of relocated actors.

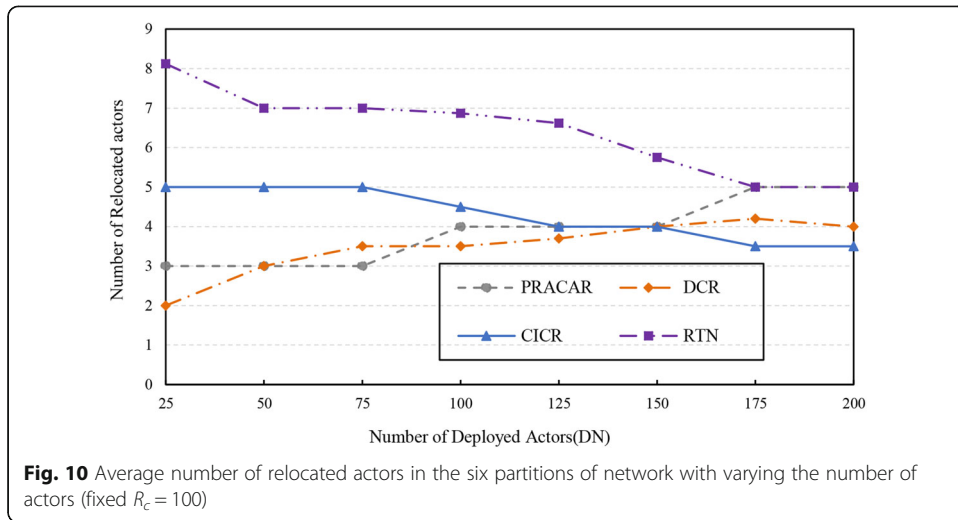
Since the long distance of these disjoint six partitions in dense network, the DCR and PRACAR which select the relocated nodes based on the neighboring distance, move more actors. The PRACAR and DCR relocate non-critical nodes and it is not suitable for sparse networks if a moved node is selected at a distance away. In RTN, at high densities ($DN > 175$), the minimum number of nodes for restoration is 1 between two partitions. The number of relocated actors is equal to the minimum value of 5. The number of partitions is equal to 6, and the relocated actors are located between two partitions. Then, for $DN > 175$, the distance between the partitions is longer, also the number of nodes needed for restoration is greater.

6.3.3 Number of exchanged messages

Figure 11 shows the impact of DN and R_c on the performance of the message overhead. As shown in Fig. 11 in CICR, the number of exchanged messages remains almost constant and even decreases, because the number of relocated actors is less than 5 in the

Table 2 The total traveled distance in six partitions of network with varying the number of actors ($R_c = 100$)

Total traveled distance							
Number of deployed actors	PRAC AR (m)	DCR (m)	RTN (m)	CICR (m)	Improvement over PRACAR (%)	Improvement over DCR (%)	Improvement over RTN (%)
25	7.87	49.93	84.9	5.38	31.64%	89.22%	93.66%
50	16.76	50.01	80.12	8.84	47.26%	82.32%	88.97%
75	27.87	53.72	74.33	12	56.94%	77.66%	83.86%
100	33.43	57.05	68	16.47	50.73%	71.13%	75.78%
125	44.54	59.12	64.35	25.24	43.33%	57.31%	60.78%
150	55.7	65.77	60.4	38.35	31.15%	41.69%	36.51%
175	61.21	67.02	59.36	43.52	28.9%	35.06%	26.68%
200	64.6	71.97	61.21	44.65	30.88%	37.96%	27.05%
Average	38.99	59.32	69.08	24.30	40.10%	61.54%	57.76%

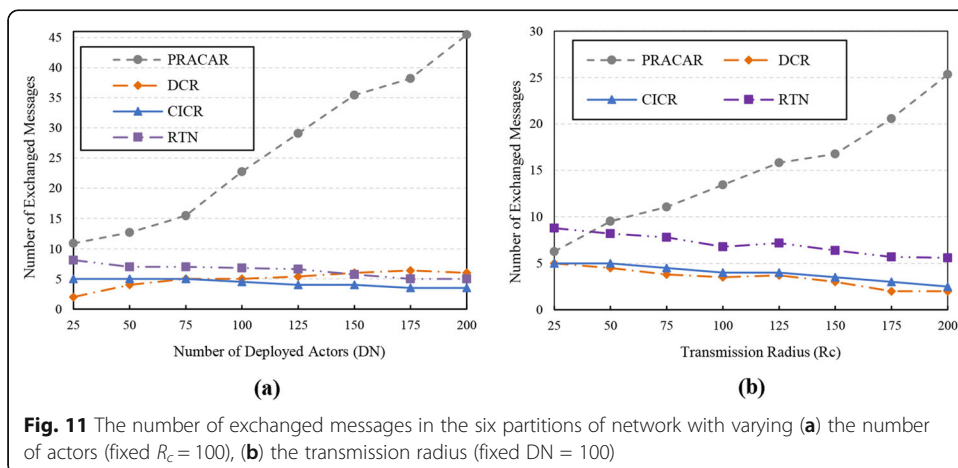


jointing of six partitions and the messaging in CICR is the same as relocated sponsor nodes.

MA sends messages containing the new location coordinates of the sponsor nodes. The DCR curve grows slowly too. But the slope of the PRACAR increases rapidly. It indicates PRACAR has not scalability and has high overhead. In this method, the relocated actor must send its ID and new location to all neighbors to know it. In the DCR, the first backup will not send any message, but if the cascading movements occurs and the number of moved backups be more than one during the restoration, the number of exchanged messages will be equal to Equation (23).

$$(\text{Moved backups} - 1) \times 2 \tag{23}$$

As the number of moved nodes increases, the message overhead also doubles. The number of exchanged messages in RTN is equal to the number of relocated backup nodes. In CICR the message overhead or communication cost is equal to the number of relocated actors.



6.4 Reliability of the proposed method

To evaluate the computational efficiency, the reliability of the proposed method has been compared with previous methods. Reliability determines the ability of a restoration algorithm to perform functions under-connected network and the ability to tolerate failure without partitioning for a specified period of time. All failed nodes that result in network partitioning in the region, also have an impact on the calculation of the network reliability. On the other hand, the network works well if all critical nodes properly work. Hence, a topology with n nodes in an area has reliability equal to the Equation (24) [37].

$$R(t) = 1 - \prod_{i=1}^n (1 - R_i(t)) \tag{24}$$

where $R_i(t)$ is the reliability of a node in an area of connectivity restoration at time t . n is the number of critical nodes for connecting the partitions and $R(t)$ is the reliability of restored connectivity of partitions. For example, the reliability of a WSA region with four required nodes to restore the partitions (each node has reliability set to 0.9) is 0.9999. It is assumed that all nodes are similar and have the same reliability.

HCR has cascaded motions, and in most cases, the number of nodes becomes the same as before. Therefore, reliability will not increase and is less than of CICR method as shown in Fig. 12. Like as HCR, CMH does not change the topology and the connection between nodes. Thus, the coverage does not increase. In HCARE method, since the backup node goes to a location that increases the coverage and connections of the nodes in the failed area, the reliability has been increased compared to the condition where the failure did not occur. Due to the total overlapping function calculated to select the backup node destination, the network becomes more resistant to the second failure. CICR also creates new connections in single-node failure and sponsor node covers two adjacent nodes as shown in Fig. 4. As shown in Fig. 12, the reliability of the region after restoration with HCR, CMH, HCARE, and CICR methods is plotted versus time. The CICR reliability has improved by 22%, 35%, and 38% in average compared to the HCARE, HCR, and CMH, respectively (refer to Table 3).

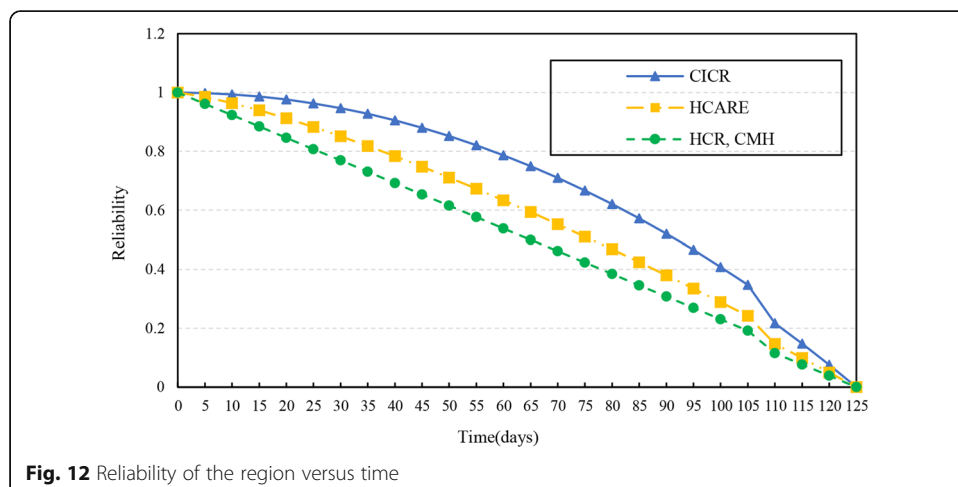


Fig. 12 Reliability of the region versus time

Table 3 Reliability of the region versus time

Time (days)	5	15	25	35	45	55	65	75	85	95	105	115	Average
HCR	0.95	0.88	0.8	0.73	0.65	0.57	0.49	0.41	0.33	0.26	0.19	0.11	0.53
CMH	0.95	0.88	0.8	0.73	0.65	0.57	0.49	0.41	0.33	0.26	0.19	0.07	0.52
HCARE	0.98	0.93	0.88	0.81	0.74	0.67	0.59	0.51	0.42	0.33	0.24	0.09	0.59
CICR	0.99	0.99	0.97	0.93	0.89	0.84	0.75	0.67	0.58	0.48	0.35	0.22	0.72
Improvement over HCR (%)													35.85%
Improvement over CMH (%)													38.46%
Improvement over HCARE (%)													22.03 %

Since same node goes exactly to the position of failed nodes in DCR and PRACAR, there are no changes in the number of nodes in the damaged location after the restoration and these methods do not change the reliability compared to pre-failure. But in CICR, at least $k = p-1$ actors are required to restore connectivity of partitions. As a result, the number of required nodes will double in comparison to the DCR and PRACAR methods. As a total result, CICR is more reliable than DCR and PRACAR methods, as shown in Fig. 13. CICR has fewer relocated nodes than RTN. The selected RTN paths are not optimal. As a result, the number of nodes between partitions and connections increases compared to CICR, and RTN reliability increases slightly compared to CICR. Also, in Fig. 13, four DCR, PRACAR, RTN, and CICR methods are evaluated and compared in terms of reliability versus time. The number of restored partitions has been 4.

The reliability of CICR has 61% and 20% improvement averagely over DCR and PRACAR, respectively, based on the values in Table 4. The reliability of RTN is about 3% higher than CICR reliability.

6.5 Limitations of study

One of the limitations of this study is that if there is an obstacle in the movement path of sponsor nodes, then the movement path into the new coordinate of sponsor nodes is not direct, and proposed equations will not satisfy. Unless, in the equations under

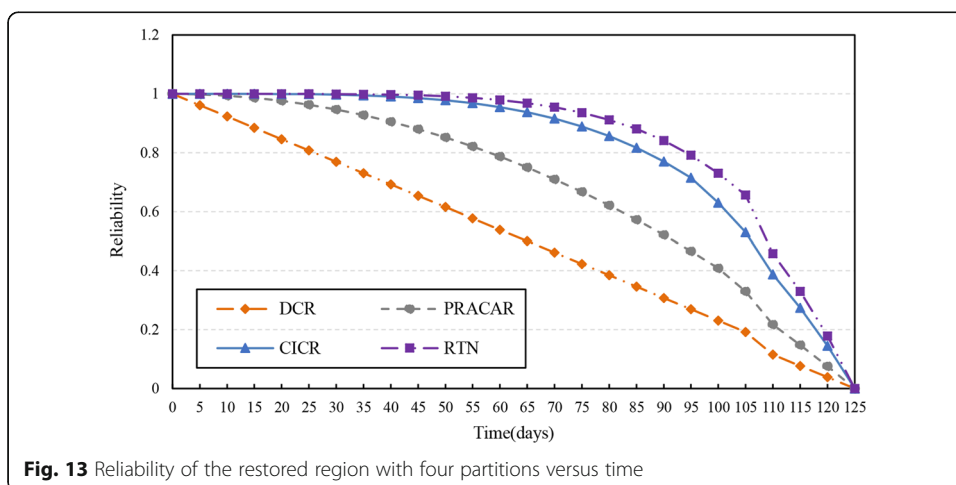


Fig. 13 Reliability of the restored region with four partitions versus time

In the future works, it is possible to research for connectivity restoration in a WSN with heterogeneous actors and various transmission radius. Also, future research could be obstacle aware algorithms.

Abbreviations

WSANs: Wireless sensor and actor networks; CICR: Computational intelligence based connectivity restoration; SIGWO: Sequential improved grey wolf optimizer; HCR: Hybrid connectivity restoration; HCARE: Hybrid coverage-aware perception-based connectivity restoration; CMH: k -connectivity maintenance and restoration algorithm for heterogeneous WSNs; PRACAR: Permanent relocation algorithm for centralized actor recovery; RTN: Round-table negotiation; WSNs: Wireless sensor networks; RPSNC: Connectivity restoration in wireless sensor networks via space network coding; DARA: Distributed actor recovery algorithm; RIM: Recovery through inward motion; IDCR WSN: Intelligent on-demand connectivity restoration for wireless sensor networks; DCR: Distributed connectivity restoration algorithm; RAM: Recovery algorithm for the failure of multiple nodes failure; SRRA: Self-route recovery algorithm; RCR: Restoring connectivity in a resource-constrained WSN; DCRMF: Distributed autonomous connectivity restoration method based on finite state machine; CPSO: Co-evolutionary particle swarm optimization; CDE: Co-evolution differential evolution; ABC: Artificial bee colony; GWO: Grey wolf optimizer; MA: Master actor; R_t : Transmission radius; MAC: Medium access control; ACK: Acknowledgement; FPGAs: Field programmable gate arrays; DN: Deployed nodes

Acknowledgements

Not applicable

Authors' contributions

SM designed and analyzed the proposed method. GF performed the interpretation of data and drafted the work. All authors read and approved the final manuscript.

Authors' information

Solmaz Mohammadi received his BSc degree in electrical engineering from Shariaty Technical College, Tehran, Iran, in 2012. She is currently an MSc student in the Electrical Engineering and Information Technology Department, Iranian Research Organization for Science and Technology (IROST), Iran. She is interested in IOT especially wireless sensor networks.

Gholamreza Farahani received his BSc degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1998, and MSc and PhD degrees in electrical engineering from Amirkabir University of Technology (Polytechnic), Tehran, Iran in 2000, and 2006, respectively. Currently, he is an associate professor with the Electrical Engineering and Information Technology Department, Iranian Research Organization for Science and Technology (IROST), Iran. One of his research interests is wireless sensors and data communication in the network.

Funding

Not applicable

Availability of data and materials

Not applicable

Competing interests

The authors declare that they have no competing interests.

Received: 12 January 2020 Accepted: 2 October 2020

Published online: 14 October 2020

References

1. I.F. Akyildiz, I.H. Kasimoglu, Wireless sensor and actor networks: research challenges. *J Ad Hoc Networks* **2**, 351–367 (2004)
2. O. Kaiwartya, A.H. Abdullah, Y. Cao, J. Lloret, S. Kumar, R.R. Hah, M. Prasad, S. Prakash, Virtualization in wireless sensor networks: fault tolerant embedding for internet of things. *IEEE Internet Things J* **5**(2), 571–580 (2018)
3. A. Cerpa, D. Estrin, ASCENT: Adaptive self-configuring sensor networks topologies. *IEEE Trans On Mobile Computing* **3**(3), 272–285 (2004). <https://doi.org/10.1109/INFCOM.2002.1019378>
4. H. Sharma, A. Haque, Z.A. Jaffery, Solar energy harvesting wireless sensor network nodes: a survey. *J Renewable and Sustainable Energy* **10**(2), 023704 (2018)
5. M. Younis, I.F. Senturk, K. Akkaya, S. Lee, F. Senel, Topology management techniques for tolerating node failures in wireless sensor networks: a survey. *J Comput Netw* **58**(254–283) (2014)
6. L. Sitanayah, K.N. Brown, C.J. Sreenan, A fault-tolerant relay placement algorithm for ensuring k vertex-disjoint shortest paths in wireless sensor networks. *Ad Hoc Netw* **23**(145–162) (2014)
7. G Wang, G Cao, T La Porta, W Zhang, Sensor relocation in mobile sensor networks. *Proc. 4th Ann. INFOCOM'05, Miami, FL, 2302–2312* (2005).
8. X Han, X Cao, EL Lloyd, , C-C Shen, Fault-tolerant relay node placement in heterogeneous wireless sensor networks. *IEEE Trans on Mobile Computing* **9**(5), 643–656 (2010).
9. SK Gupta, P Kuila, PK Jana, Genetic algorithm for k -connected relay node placement in wireless sensor networks. *Proc. Second International Conf. Computer and Communication Technologies, Advances in Intelligent Systems and Computing*. 379, 721–729 (2016).

10. A. Hashim, B.O. Ayinde, M.A. Abido, Optimal placement of relay nodes in wireless sensor network using artificial bees colony algorithm. *J Netw Comput Appl* **64**(3), 239–248 (2016)
11. D.R. Dandekar, P.R. Deshmukh, Relay node placement for multi-path connectivity in heterogeneous wireless sensor networks. *Procedia Technology* **4**, 732–736 (2012)
12. A. Uwitonze, J. Huang, Y. Ye, W. Cheng, Connectivity restoration in wireless sensor networks via space network coding. *Sensors* **17**(4), 1–21 (2017)
13. A. Abbasi, M. Younis, K. Akkaya, Movement-assisted connectivity restoration in wireless sensor and actor networks. *IEEE Trans Parallel and Distributed Systems* **20**(9), 1366–1379 (2009)
14. M. Younis, S. Lee, A. Abbasi, A localized algorithm for restoring internode connectivity in networks of moveable sensors. *IEEE Trans Computers* **59**(12), 1669–1682 (2010)
15. M. Khalid, A.K. Muhammad, U.H. Mahmood, M.S. Ansar, A. Shahzad, K.S. Muhammad, Intelligent on-demand connectivity restoration for wireless sensor networks. *Wirel Commun Mob Comput* **2018**(1–10) (2018)
16. K. Yan, G. Luo, L. Tian, Q. Jia, C. Peng, Hybrid connectivity restoration in wireless sensor and actor networks. *EURASIP J Wireless Com Network* **138**(1–16) (2017)
17. Y. Zhang, Z. Zhang, B. Zhang, A novel hybrid optimization scheme on connectivity restoration processes for large scale industrial wireless sensor and actuator networks. *Processes* **2019**, *7*, 939, 7(12), 939 (2019).
18. V.K. Akram, O. Dagdeviren and B. Tavli, Distributed k-connectivity restoration for fault tolerant wireless sensor and actuator networks: algorithm design and experimental evaluations, in *IEEE Transactions on Reliability*. 1–14 (2020).
19. P. A Humblet, An adaptive distributed Dijkstra shortest path algorithm. Tech. Rep. LIDS-P-1775, Lab. Inf. Decision Syst., Massachusetts Inst. Technol., Cambridge, MA, USA, 1988.
20. M. Imran, M. Younis, A.M. Said, H. Hasbullah, Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks. *J Netw Comput Appl* **35**(2), 844–856 (2012)
21. K. Mahmood, M. Hassan, M. Mahmood, Permanent relocation and self-route recovery in wireless sensor and actor networks. (IJACSA) International J. Advanced Computer Science and Applications **9**(3), 83–89 (2018)
22. S. Shrivastav, D. Ghose, Round-table negotiation for fast restoration of connectivity in partitioned wireless sensor networks, *Ad Hoc Networks* **77**. 11–27 (2018).
23. Y.K. Joshi, M. Younis, Restoring connectivity in a resource constrained WSN. *J Netw Comput Appl* **66**, 151–165 (2016)
24. Y. Zhang, J. Wang, G. Hao, An autonomous connectivity restoration algorithm based on finite state machine for wireless sensor-actor networks. *Sensors* **18**, 1 (2018)
25. H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl Soft Comput* **10**(2), 629–640 (2010)
26. Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problem. *Eng Appl Artif Intell* **20**(1), 89–99 (2007)
27. F.Z. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization. *Appl Math Comput* **186**(1), 340–356 (2007)
28. I. Brajevic, Crossover-based artificial bee colony algorithm for constrained optimization problems. *Neural Computing and Applications*. 26 (7) (2015).
29. S.A. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer. *Adv Eng Softw* **69**, 46–61 (2014)
30. M.C. Metz, J.A. Vucetich, D.W. Smith, D.R. Stahler, R.O. Peterson, Effect of sociality and season on gray wolf (*Canis lupus*) foraging behavior: implications for estimating summer kill rate. *PLoS One* **6**(3), 1–10 (2011)
31. K. Luo, Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey. *Applied Soft Computing Journal*. 77 (2019).
32. M. Jorgic, M. Hauspie, D. I. Stojmenovic, Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks. *Mediterranean Ad Hoc Networking Workshop, Turkey*.12. (2004).
33. A. Liu, M. Dong, K. Ota, J. Long, PHACK: an efficient scheme for selective forwarding attack detecting in WSNs. *Sensors* **15**, 30942–30963 (2018)
34. J.F. Bonnans, J. Ch Gilbert, C. Lemarechal, C.A. Sagastizabal, *Numerical optimization: theoretical and practical aspects* (Springer-Verlag, Berlin, 2006)
35. Zhi-jun Teng, Jin-ling Lv, Li-wen Guo, An improved hybrid grey wolf optimization algorithm. Springer-Verlag GmbH Germany, part of Springer Nature. 23(15), 6617–6631 (2019). <https://doi.org/10.1007/s00500-018-3310-y>.
36. S. Rajopadhye, M. Mills Strout, Languages and compilers for parallel computing, (24th International Workshop, LCPC 2011 Fort Collins, CO, USA, September 8–10, (2011).
37. M. Rausand, *System Reliability Theory*, 2nd edition. Wiley, (2003).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.