

A Dynamical Relay Node placement Solution for MANETs

Roberto Magán-Carrión^{a,*}, José Camacho^a, Pedro García-Teodoro^a,
Eduardo Feo Flushing^b, Gianni A. Di Caro^c

^a*Network Engineering & Security Group (NESG, <http://nesg.ugr.es>)
Dpt. of Signal Theory, Telematics and Communications - CITIC
University of Granada - 18071 Granada (Spain)*

^b*Istituto Dalle Molle di studi sull'intelligenza artificiale (IDSIA, <http://www.idsia.ch>)
Galleria 2, Via Cantonale 2c - CH-6928 Manno (Switzerland)*

^c*Carnegie Mellon University Qatar (<https://www.qatar.cmu.edu>)
CMU-Q Education City - PO 24866 Doha (Qatar)*

Abstract

Network deployment in wireless networks implies the distribution of the communication nodes to improve some key operational aspects, such as energy saving, coverage, connectivity, or simply reducing the network cost. Most node placement approaches are focused on static scenarios like WSNs, where the topology of the network does not vary over time. Nevertheless, there exist certain situations in which the network node locations can continuously change. In this case, the use of special nodes, so-called Relay Nodes (RNs), contributes to supporting, maintaining or recovering communication in the network. The present work introduces a multi-stage dynamical RN placement solution to lead the RNs to their time-varying optimized positions. The approach, named Dynamical Relay Node placement Solution (DRNS), is based on the use of Particle Swarm Optimization (PSO) algorithms and is inspired by Model Predictive Control (MPC) techniques following a bi-objective optimization procedure, where both network connectivity and throughput are jointly maximized. DRNS is validated in both simulated and real environments composed of mobile robotic nodes, the results showing its goodness

*Corresponding author. Tlf.: +34 958241717

Email addresses: rmagan@ugr.es (Roberto Magán-Carrión), josecamacho@ugr.es (José Camacho), pgteodor@ugr.es (Pedro García-Teodoro), eduardo@idsia.ch (Eduardo Feo Flushing), gdicaro@cmu.edu (Gianni A. Di Caro)

and operational suitability for real MANET environments.

Keywords: Relay node, Placement solution, Optimization process, PSO, MANET

1. Introduction

Node placement design strategies rely on a diversity of network performance goals: coverage, network connectivity and network longevity, among other objectives, and it will depend on the specific application/context. In particular, on the static or dynamic nature of the environment. In the former case, the initial location of the nodes remains constant over time, whereas in the latter case the locations will change [1]. An example of static network scenarios are Wireless Sensor Networks (WSNs) [2]. On the other hand, Mobile Ad hoc NETWORKS (MANETs) [3] constitute a typical example of dynamically changing scenarios.

MANETs are recommended, for example, when the fixed communication infrastructure is no longer available due to natural disasters to support communication among human rescue patrols. In such situations, maintaining communication is of utmost importance because connectivity degradation could cause human losses. For that, it is useful to consider additional nodes called Relay Nodes (RNs), whose main role is to forward the information among the remaining nodes. Consequently, the adequate use of RNs is particularly recommended to provide more resilient and survivable networks against disconnections regardless of the origin (movement of the nodes, malfunctions or malicious actions).

Optimal RN placement is a very challenging problem that has been proven to be NP-Hard [1]. To address such a complexity, most proposals in the specialized literature provide suboptimal solutions supported by heuristics, and almost all mainly focus on static environments [4]. Thus, most current works on RN placement are intended to find an optimal (minimum) number of RNs by considering specific performance goals [5, 6]. On the other hand, in dynamically changing scenarios, static solutions cannot be successfully applied because the location optimality at a certain time does not hold during network operation. Therefore, proposals addressing RN dynamic re-positioning according to the changes in the network topology are necessary [7, 8, 9, 10].

Although dynamic RN re-location solutions provide more adaptable systems against different circumstances, they involve additional challenges to be

addressed in comparison with their static counterparts. Basically, dynamical RN placement solutions should give response to the following two questions: (i) what are the RN optimized locations at a given time, and (ii) how should they move toward those locations. The former one implies considering extra factors such as the number of nodes, the area to be covered, the node movements, the nodes' coverage range, and the number of RNs considered. In regard to the second challenge, it is convenient to consider in some sense the network dynamism to infer future network node locations that lead the system to achieve better RN motion trajectories. Additionally, the RN nodes should be moved in a coordinated manner because the placement of one RN directly influences the position of the remaining RNs.

To address the above issues, we introduce here a Dynamical Relay Node placement Solution (DRNS) as the natural extension of the work in [4] from static network environments to dynamic ones (*e.g.*, MANETs). It is based on the work by Dengiz *et al.* in [9] (hereafter DKS, standing for the initials of the authors), which presents some severe drawbacks properly overcome by DRNS. As in [9], DRNS relies on optimization procedures inspired by Model Predictive Control (MPC) [11] methodologies together with Particle Swarm Optimization (PSO) [12] based algorithms to locate and control the RN movements. The so-called *evolutionary algorithms* are well suited for highly changing, complex and dynamic environments [13] where problem solutions must be obtained in a reasonable time. In particular, PSO has been selected due to its wide usage by the research community for RN placement problems [8, 9, 10, 14]. However, Genetic Algorithms (GAs) are a close variants of evolutionary solutions that could be suitable for this approach as well [15, 16].

DRNS is a multi-stage RN placement procedure that jointly optimizes network connectivity and throughput by simply considering the available number of RNs. The proposal follows a two-step iterative process to adapt to the topology changes over time. First, several reference points are optimally computed in a receding horizon considering the system performance goals. Then, the RNs are driven to the reference points in a controlled manner. The viability and efficacy of the system are assessed both under simulation and in a real robotic-based MANET scenario, which constitutes a significant contribution in comparison with similar proposals.

The remainder of the paper is organized as follows. Section 2 discusses the problem of RN placement and presents several principal works in the field. The general assumptions and the formulation of the RN placement

proposal are introduced in Section 3. Based on this, we describe in Section 4 the modular structure of our specific RN placement solution, DRNS. The simulation study scenario, the experimentation and the obtained performance results are discussed afterwards in Section 5. In Section 6, the solution is evaluated in a real MANET environment where the nodes are robots. Finally, Section 7 presents the main conclusions of the work and some further research directions.

2. Relay node placement: techniques, schemes and solutions

The RN placement problem has been recurrently addressed in the literature. However, it is still an engineering challenge. Among the vast literature on this topic, we can differentiate main groups or categories by their principal goal. Most solutions are aimed at providing k -connected networks, whereas other methods address different goals such as reducing the overall delay of communication or maximizing the network coverage. Additionally, multi-objective optimization RN placement solutions exist in the literature as well, where several performance objectives are jointly considered.

2.1. RN placement for k -connectivity

Several RN placement solutions have the same objective: to provide k -connected networks with the minimum number of RNs [5, 6, 17, 18, 19, 20]. A network is k -connected if there exists at least one path connecting every node after removing $k-1$ nodes in the network. To ensure fault tolerance, k should be greater than or equal to 2, whereas $k = 1$ satisfies network connectivity.

In [17], the authors propose several solutions for *single-* and *two-tiered* connected networks with the minimum number of deployed RNs. The same authors extend this work in [18] by designing a 2-connected approximation algorithm also for single- and two-tiered approaches. K -connected networks are studied in [19] for arbitrary k . The authors demonstrate that regardless of the selected k parameter, they can obtain an approximate solution in polynomial time. A similar approach is introduced in [5], which locates the minimum number of RNs for k -connectivity. The proposed heuristic outperforms [19] by applying a simple selection step to remove some RNs while the k -connectivity is preserved.

Like in [17], the objective in [20] is to provide fault tolerance in heterogeneous WSNs, where sensors have different transmission radii. They develop a $k \geq 1$ connected approximation solution considering the desired degree

of fault tolerance (full- or partial-fault resiliency) and the direction of the communication flows (one or two way). A connectivity and energy efficient approach for relay placement in WSNs is proposed in [6], where the relay nodes can only be placed at pre-specified candidate locations. To ensure network survivability, the goal is to form a 2-connected network.

Although further works have appeared in the field, they are similar to the previous ones. For example, the work in [21] considers two sets of stationary and mobile relay nodes to provide connectivity in WSNs. The relays in the first set are located to increase the connectivity among terminals. Instead, the relays in the second set are driven following an optimized route for reaching terminals not previously covered. Authors in [22] introduce an One-Step RN Placement (OSRP) algorithm for two-tiered constrained sensor networks. Ma *et al.* propose in [23] a Relay Location Selection Algorithm (RLSA) to build a connectivity based high-tier network. Likewise, authors in [24] develop a RN placement solution based on a realistic physical layer model and the calculation of Signal-to-Interference-plus-Noise-Ratio (SINR) to minimize the number of RNs that guarantee connectivity. A two-tier relay node placement solution is also proposed in [25]. In this case, a linear programming approach based on connectivity, coverage and network lifetime as performance objectives is developed for WSNs. A PSO-based optimization procedure is proposed in [26]. The system optimizes the location of a set of Unmanned Aerial Vehicles (UAVs) to improve the communications performance among ground nodes/vehicles in urban environments. The work concludes that maximizing the global message connectivity together with the minimization of the worst connectivity case lead the system to achieve the best performance.

Reactive approaches are also needed in dynamic environments, where the node movements give way to partitions in the network. Authors in [27] address the re-establishment of the network topology that is lost due to node movements. Similarly, in [28] military scenarios with connectivity failures are studied. In this case, the survivability of the network is ensured by locating Flying Aerial Platforms (FAPs) as relay nodes. To this end, deterministic annealing is used to find global solutions with the minimum number of FAPs. Alfaqdhly *et al.* investigate in [29] the optimal repositioning for recovering connectivity after one or multiple node failures. For that purpose, the problem is formulated as an integer linear programming problem to maximize coverage and minimize distances among nodes. The same authors introduce in [30] the Least Distance Movement Recovery (LDMR) algorithm, a

distributed approach that exploits non cut-vertex actors in the recovery process. In a similar line, [31] proposes the so-called Recovery through Inward Motion (RIM), a distributed algorithm to recover the network connectivity after a node failure by relocating the neighbors of the lost node. A WSN self-adaptation solution is proposed in [32], where the authors activate certain sensor nodes to restore the connectivity lost due to the *dumb* behavior of some of them. A sensor node becomes a dumb node when the environmental conditions makes it not able to transmit the measurements to the base station. This problem can be seen as a relay node restricted placement problem since the available candidate locations are predefined.

In [33], two distributed RN positioning approaches for restoring connectivity in partitioned WSNs are introduced. One approach is based on virtual magnetic forces, which gradually stretch the network by deploying additional relays until the pushing forces allow a stable state to be achieved. The other approach exploits game theory among the leaders of the partitions.

Lin *et al.* consider in [34] the problem of dynamic Router Node Placement (dynRNP) in Wireless Mesh Networks (WMNs), in which both mesh clients and mesh routers have mobility. Mesh clients can also switch their network access on or off at different times. The authors use a Bat-inspired Algorithm (BA) with a dynamic local search selection scheme. This simulates the echolocation of bats to find the optimal solution to maximize both network connectivity and client coverage.

Authors in [35] attempt to place the minimum number of relay sensors to maintain global connectivity, where the locations of the relay nodes are restricted to the minimum Steiner tree. Also a similar approach also appears in [36], where a Steiner tree-based heuristic is used to minimize the number of relays needed to restore the connectivity.

2.2. RN placement for different purposes

There exist a number of solutions whereby a variety of performance objectives are established. Authors in [37] design a multi-hop wireless mesh network with the minimum number of additional RNs. The placement of the relays is designed to ensure that the delay on the paths between the base station and the sensors meet a pre-specified delay bound. A branch-and-cut algorithm is defined to this end. For sensing coverage in WSNs, authors in [14] propose a PSO-based solution to minimize the existing coverage holes through the use of a fitness function based on the computation of Voronoi

regions. Authors in [38] propose to relieve bottlenecks in wireless ad hoc networks by deploying relay nodes. To that end, they firstly identify these key network locations through a novel network connectivity measurement based on the use of the so-called Cheeger constant. Secondly, they run an ILP algorithm to optimize the location of the relays such that the connections from both sides of the bottleneck are maximized. A modified PSO-based proposal is also used in [10] to address the sink placement problem in WSNs by minimizing the worst case delay path in the network. A minimax scheme is developed in [39] to optimize coverage ratio and uniformity.

On the other hand, authors in [7] propose an algorithm that emulates the attractive force (as in a stretched spring) and the repulsive force (such as the electrostatic force) in nature such that a robotic node simply follows the resultant virtual force to move to locations that maximize coverage and minimize moving distance. Wang *et al.* address in [40] the problem of node placement in wireless networks for ensuring complete coverage in a long belt scenario. They also propose an inter-node distance-based approach to minimize the number of nodes needed. The authors in [41] determine the number and deployment of heterogeneous devices such that the total network monetary cost is minimized while constraints regarding lifetime, coverage and connectivity are satisfied. The work has been further extended in [42] to address the problem of deploying a second-tier of RNs to balance the traffic using the fewest additional RNs.

Further works address the placement problem from other different perspectives. For example, in natural disasters scenarios where emergency rescue actions are needed, supporting live video transmissions with satisfactory Quality of Experience (QoE) is especially crucial for a precise guiding of the rescuers throughout the affected area. With this aim, the authors in [43] propose a relay node placement solution in Flying Ad hoc NETWORKs (FANETs). The system drives the UAVs considering firstly, the routing paths where they belong to, and secondly, their current location and the UAV mobility model. The solution precludes the negative effects of the UAV movements on the connectivity having a direct impact over the system performance. Similarly, the work in [44] proposes a feedback motion algorithm to guide and locate UAV devices between the coordination mission station and deployed vehicles in search and rescue maritime actions for communications support. The algorithm runs in the UAVs and gathers network environment information. This information is afterwards used to produce a QoS feedback for the coordination station to compute new UAV way points.

The authors in [45] consider burdens of all communication nodes. The burden is defined from the load of each node based on the number of edge disjoint paths. On the other hand, Misra *et al.* in [46] and Yang *et al.* in [47] try to minimize energy harvesting in WSN.

2.3. Multi-objective optimization for RN placement

Some RN location proposals are aimed at simultaneously optimizing more than one performance metric. For instance, coverage and connectivity are jointly addressed in [47]. Similarly, the work in [48] proposes a two-tier based RN placement solution for coverage and connectivity purposes in WSNs with communications delay restrictions. The authors develop an iterative Two-phase Set-Covering-based Algorithm (TSCA) where the RNs are deployed firstly, for sensor coverage and secondly, for network connectivity. A bi-objective (user coverage and network connectivity) genetic-based optimization algorithm is considered in [15] for node placement in wireless mesh networks. Another multi-objective optimization is carried out in reference [49], where a PSO algorithm is proposed to determine the best placement of nodes in industrial environments in terms of network reliability, load uniformity, total cost and convergence speed. On the other hand, [50] addresses RN location based on system performance and connectivity. For that, the authors maximize the Packet Delivery Ratio (PDR) by considering network aspects such as channel interference and congested areas.

An iterative convex optimization is proposed in [51] to maximize the throughput between two ground static nodes and an UAV acting as a relay node. To that end, they jointly optimize both the relay trajectory and the source/relay transmission power allocations that rely in a higher end-to-end system throughput. Aimed at improving 5G communication networks, the authors in [52] develop a new entropy-based approach for a cooperatively deployment of UAV vehicles. The solution offers a notable improvement in terms of throughput and communication delays while reduces the network deployment cost in comparison with fixed network architectures.

It is also remarkable the work of Ouchitachen *et al.* [16], where a genetic algorithm based on two criteria (cost and communication quality) are used to optimize the energy usage in mobile WSNs.

Maintaining or recovering the connectivity lost due to disconnections caused by node movements, failures or misbehavior may become a crucial task depending on the network application. In addition, preserving a level of Quality of Service (QoS) in terms of the available throughput is mandatory

to provide some network services. Based jointly on network throughput and connectivity, authors in [8] and subsequently in [9] make use of a PSO-based optimization procedure and MPC-inspired techniques for MANETs. In an attempt to address some deficiencies of this work, Magán-Carrión *et al.* provide in [4] a modular and scalable RN placement scheme to be applied in static environments. Taking this work as a base, we propose in the present work a further extension of the previous RN placement solution intended to be deployed in dynamic scenarios. Experimental results will support the feasibility of the placement solution for real MANET environments, which is not usually demonstrated in other current developments.

3. RN placement for connectivity and throughput optimization

Before describing our complete RN placement solution, we introduce here the mathematical formulation of the problem to solve. For this, let us first introduce some primary concepts and assumptions that shape our solution:

- *Two types of network nodes are considered: UNs and RNs.* Only user nodes and relay nodes are deployed in the network, both of them with mobility capabilities.
- *The optimization procedure is centralized.* This implies that there exists a node capable of, through adequate communication and processing means: (i) retrieving the necessary network information, (ii) running the optimization algorithm from this information, and (iii) sending control data (*i.e.*, the target locations) in accordance with the result of the optimization.
- *The positions of the UNs in the network are not controlled.* Only the positions of the RNs can be optimized. This is a reasonable assumption that complicates the problem. Moreover, this problem definition generalizes the simpler one in which the location of the UNs can also be optimized.
- *The number of RNs is limited.* This is a realistic assumption, since in most practical situations just a limited number of RNs is available.

Additional secondary assumptions to limit the scope of the present work are as follows:

- *The optimization is limited to a 2D space.*
- *The network is single tiered such that both UNs and RNs relay information from other UNs.*
- *The communication range for both UNs and RNs is the same: c units.*
- *Both types of nodes have a limited speed.*

In this context, a MANET consisting of several UNs and RNs can be specified as follows:

$$G = (N, E) \tag{1}$$

where N is the complete set of wireless nodes, *i.e.* user nodes and relay nodes ($N = U \cup R$), and E the wireless links or edges.

In a wireless network, the edges satisfy

$$E := \{\mathbf{e}_{ij} \mid \|\mathbf{e}_{ij}\| \leq c, \forall n_i, n_j \in N\} \tag{2}$$

with c being the communication range of a single link and n_i and n_j being the i th and j th network nodes, respectively.

In general terms, the optimal placement of the RNs can be represented by the following graph:

$$\mathbf{G}^* := \arg \max_{\mathbf{G}} \{f(G) \mid G = (N, E) \text{ and } U = U_0\} \tag{3}$$

where $f(G)$ is the function to be maximized and U_0 is the actual location of the UNs. In our solution, the function $f(G)$ represents a procedure wherein both connectivity and throughput will be directly or indirectly considered.

3.1. Optimized RN location and movement

The RNs have to be moved in accordance with the changes produced in the environment. To reduce response times, they may take advantage of a certain awareness of the future changes, leading to more efficient trajectories. Inspired by MPC techniques, the behavior of the MANET can be predicted within a receding horizon with H steps.

To this end, we propose a placement solution based on DKS (see the Appendix for a detailed explanation of the DKS solution). However, DKS does not address properly the two main aspects mentioned in Section 1: (*i*)

which is the best location of each available RN, and (ii) how it should move to that location. The principal reasons for that are [4]:

1. DKS considers three objective functions to drive optimization without a thorough study of their interactions. This provides suboptimal solutions and problems in the boundaries amongst these functions.
2. In DKS the network connectivity is addressed through the use of a discrete cost function $O_1(G)$ (see Eq. (A.1) in Appendix), which can divide the search space into flat regions without differential information to drive the optimization.
3. DKS moves the RNs to certain imaginary points, without taking into account whether such positions provides adequate coverage. Such points are the so-called *attraction points* (APs) and are roughly defined as the middle point between two network partitions.
4. DKS leads the actual movement of a single RN (the nearest one to a given AP), the rest of RNs remaining uncontrolled.

To solve the above problems, we propose a unique objective function dependent on the network status, *i.e.* if the network is totally connected or suffers from disconnections. For that, the network status (network connectivity) is computed without considering the RNs in G . Thus, we denote G^U as the new network graph only considering the UNs.

3.1.1. Optimization for connected networks

A connected network means that $O_1(G^U) = 1$. In such a case, the network connectivity is the maximum; however, the network throughput could still be maximized. We devise here a unique cost function expressed by

$$g(G') = \sum_{\forall u_i, u_j \in U: j > i} id(G'_i, i, j) \quad (4)$$

where G'_i is the spanning tree starting at the i th UN and minimizing the distance of the largest edge of each path in the network. The function $id(G'_i, i, j)$ is the inverse of the length of the longest edge in the path from node i to j in the G'_i network. $g(G')$ can be interpreted as an estimation of the overall network throughput, where the corresponding throughput between two adjacent nodes is inversely related to their distance. Additionally, $g(G')$ can be seen as a smoother and continuous version of $O_1(G)$, with $G' = (N, E')$ and $E' := \{e'_{ij} \mid \forall n_i, n_j \in N\}$.

To see more details about the $g(G')$ function, we recommend reading the previous work of the same authors [4], wherein an RN placement solution for static ad hoc environments is proposed.

3.1.2. Optimization for disconnected networks

Cases where $O_1(G^U) < 1$ correspond to disconnected networks. This situation is expected to be frequent in MANET environments due to the inherent nodes' mobility. Consequently, the system will attempt to move the RNs to those previously computed APs for recovering (or 'just' improving) the lost connectivity. We propose the function $p(R^c, A^*)$:

$$p(R^c, A^*) = \sum_{i=1}^{R^c} \sum_{j=1}^{A^*} d(r_i^c, a_j^*) - \sum_{i,j \in R^c: j>i} d(r_i^c, r_j^c) \quad (5)$$

where R^c is the set of candidate RN locations under evaluation, r_i^c is the i th RN candidate location, a_j^* is the j th optimized AP location, and $d(.,.)$ stands for the Euclidean distance.

The function $p(R^c, A^*)$ attempts to move the RNs to the APs while ensuring that the former ones remain separated. A more detailed justification of $p(R^c, A^*)$ can be found in [4].

3.1.3. Complete view

For the sake of clarity, the proposed complete optimization placement procedure can be summarized as follows:

$$R^* := \begin{cases} \arg \max_{R^c} \{g(G'(U^{(t+H)} \cup R^c, E))\}, & \text{if } O_1(G^U) = 1 \\ \arg \min_{R^c} \{p(R^c, A^{*(t+H)})\}, & \text{if } O_1(G^U) < 1 \end{cases} \quad (6)$$

where R^* is the resulting set of optimized RN locations depending on the network status.

4. Dynamical Relay Node placement Solution (DRNS)

According to the previous problem formulation, we devise a novel RN placement solution for dynamic scenarios. It is named DRNS, from 'Dynamical RN placement Solution'.

The functional and modular architecture of DRNS is shown in Figure 1(a). Leftmost, the *DKS motion prediction* infers the UN positions in a

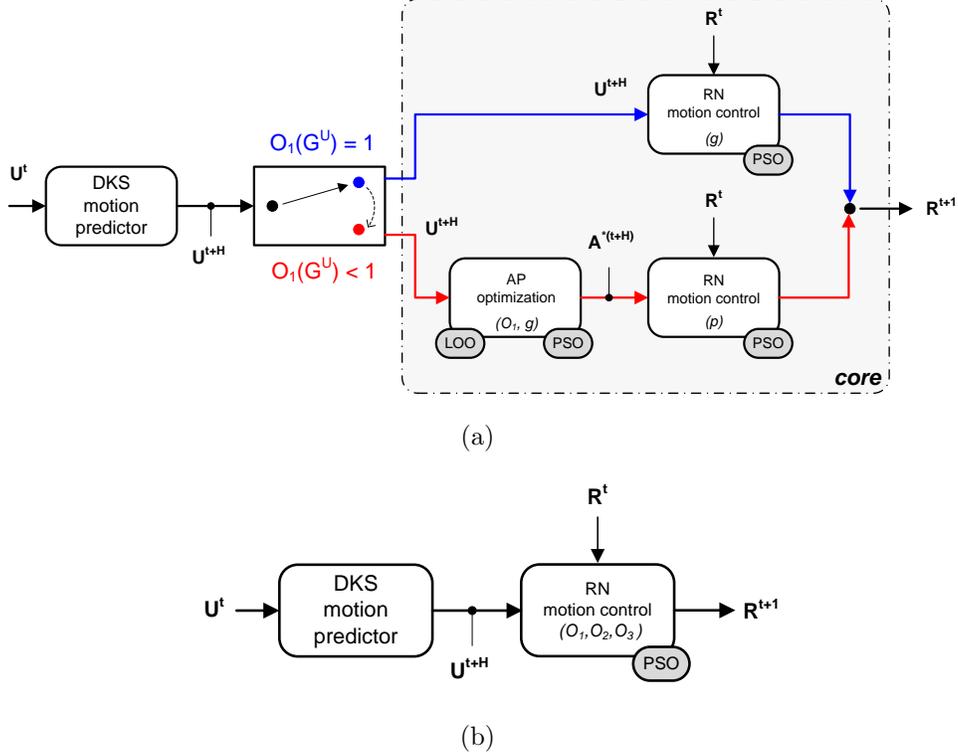


Figure 1: Functional blocks of DRNS (a) and DKS (b) proposals.

receding horizon $(t + H)$. Such a module is inherited from the DKS solution (see Figure 1(b)). The UN location prediction $U^{(t+H)}$ is then used to select the corresponding optimization method depending on the function $O_1(G^U)$ (see Eq. (6)). Only for the case of disconnected networks, we first need to compute an optimized set of APs ($A^{*(t+H)}$) through the execution of the *AP optimization* module. The *RN motion control* module comes into play by running a PSO algorithm considering the corresponding unique cost function in accordance with the network status. On the one hand, when the network is disconnected (*i.e.*, $O_1(G^U) < 1$) such a module drives the RNs toward the previous computed set of APs to recover the connectivity lost by minimizing the $p(R^c, A^*)$ function. On the other hand, when the network is totally connected (*i.e.*, $O_1(G^U) = 1$), the RN motion control module moves the RNs to maximize the network throughput through the function $g(G')$. In Figure 1(b), we can see that the DKS proposal uses a unique optimization procedure wherein three different cost functions are involved. As discussed in Section

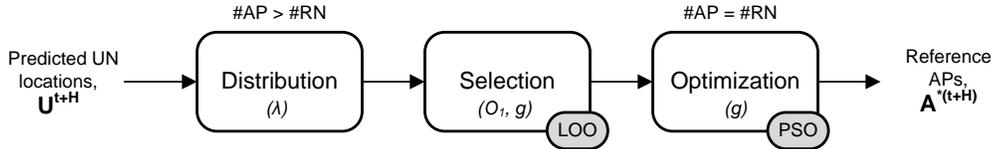


Figure 2: Stages of the AP optimization module.

3, this process causes one of the main drawbacks of DKS.

4.1. AP optimization module

The fundamentals and details of this module can be seen in the previous work [4] of the same authors. In that work, we prove the feasibility of such a solution as a complete RN placement scheme for static multi-hop wireless networks. DRNS inherits [4] and extends it by adding several modules. All together optimize and control the locations and movements of the RNs in dynamic environments. For the sake of conveying a better understanding, we briefly describe here the main parts and functionality of this module.

As previously mentioned, the DKS solution does not locate the APs properly. The present module solves this limitation by first considering the available number of RNs and then optimizing the AP positions in accordance with the system performance goals. Following the previous premises, the module is in turn divided into three associated modules, each devised to perform a specific task to generate efficient AP locations. The associated modules and their relationships are shown in Figure 2. First, the APs are placed equally spaced. They are distributed along the edges connecting partitions in the network according to a distance-based spanning tree. The number of APs is customized by a user-defined parameter $\lambda \in (0, 1]$. Then, a Leave-One-Out (LOO)-based procedure is executed to generate an AP *selection* from the previous set. The number of APs selected corresponds with the number of available RNs. Finally, from the previous candidate solution, a PSO *optimization* procedure re-locates the selected APs such that their new positions improve the system performance in terms of connectivity and throughput.

Figure 3 shows an example of the results of each module from the optimization process. A remarkable fact is the relocation of the APs from their positions obtained in the selection stage (Figure 3(c)) to those reached in the optimization stage (Figure 3(d)): the PSO extends the search space, leading the system to find a better solution not contemplated in the previous stage.

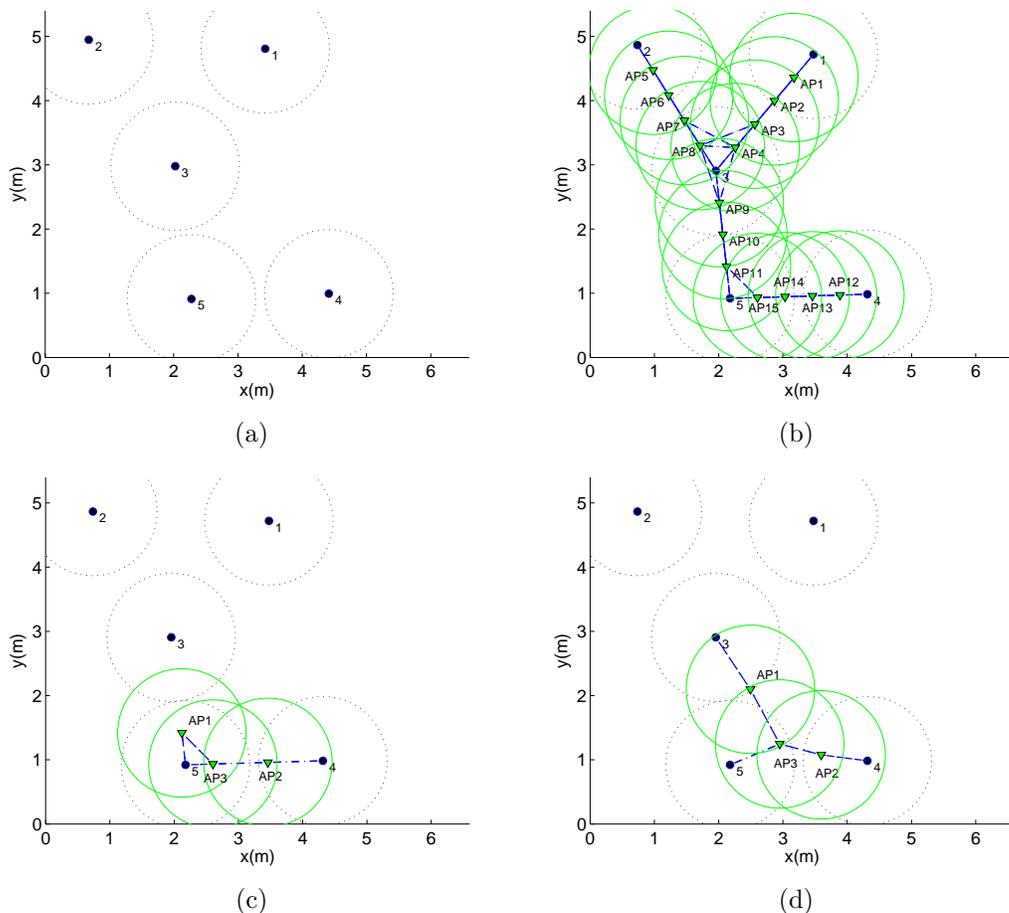


Figure 3: Locations of the APs after each stage considering 5 UNs (filled circles) and 3 available APs (filled inverted triangles): (a) initial UN distribution, (b) initial candidate AP locations along the spanning tree minimizing the inter-partitions distance, (c) AP selection, and (d) AP optimal locations after the PSO-based optimization.

4.2. RN motion control module

Regardless of the case, either moving the RNs to the previously computed APs for disconnected networks or toward the locations that maximize the network throughput, such movements have to be controlled. To this end, a motion control module inspired by the MPC methodology is implemented [11]. This module is again based on the use of the PSO algorithm, as detailed in Figure 4. It is worth mentioning that this PSO and its homologous in the AP optimization have different purposes. Therefore, PSO meta-parameters

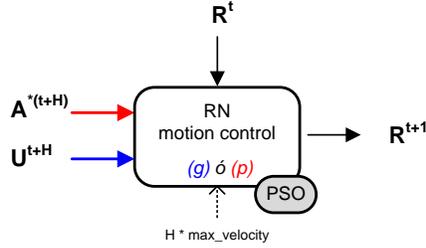


Figure 4: Detailed structure of the motion controller module.

need to be chosen separately. The AP optimization PSO is configured to be more exploratory than the PSO included in the motion control in order to enlarge the search space. These different behaviors are obtained by tuning parameters such as the *cognitive* and *social* PSO coefficients [12] and the *inertia* coefficient [53]. In particular, we have set higher values for the latter coefficient in the AP optimization block than in the RN positioning block because, in the first case, a larger search space is needed to explore new solutions that were not considered during the AP selection stage. Similarly, the velocity increments for the particles are also larger in the AP optimization stage.

Depending on the current network status (see Section 3), this module drives the RNs according to a specific objective function. For that, the optimization procedure uses the function $p(R^c, A^*)$ in the presence of disconnections and the function $g(G')$ for totally connected networks. The solutions obtained are constrained to $H \cdot \text{max_velocity}$, where max_velocity denotes the maximum RN velocity. This way, the procedure can obtain the optimal RN positions up to H time steps ahead from the initial RN locations, R^{t+H} . Following the MPC methodology, although the optimization is solved H steps ahead, only one step is implemented, and the iteration is repeated.

Although the valid RN locations are obtained time step by time step (*i.e.*, each time the complete PSO algorithm finishes, $R^t \rightarrow R^{t+1}$), working with the network evolution prediction H time steps ahead makes the system able to adapt the location solution to the continuous changes in the network topology more efficiently than simply considering the current network node locations. We will see in Section 5, through adequate experimentation, how the system reaches a stable state around a performance value from a certain simulation time step to the end.

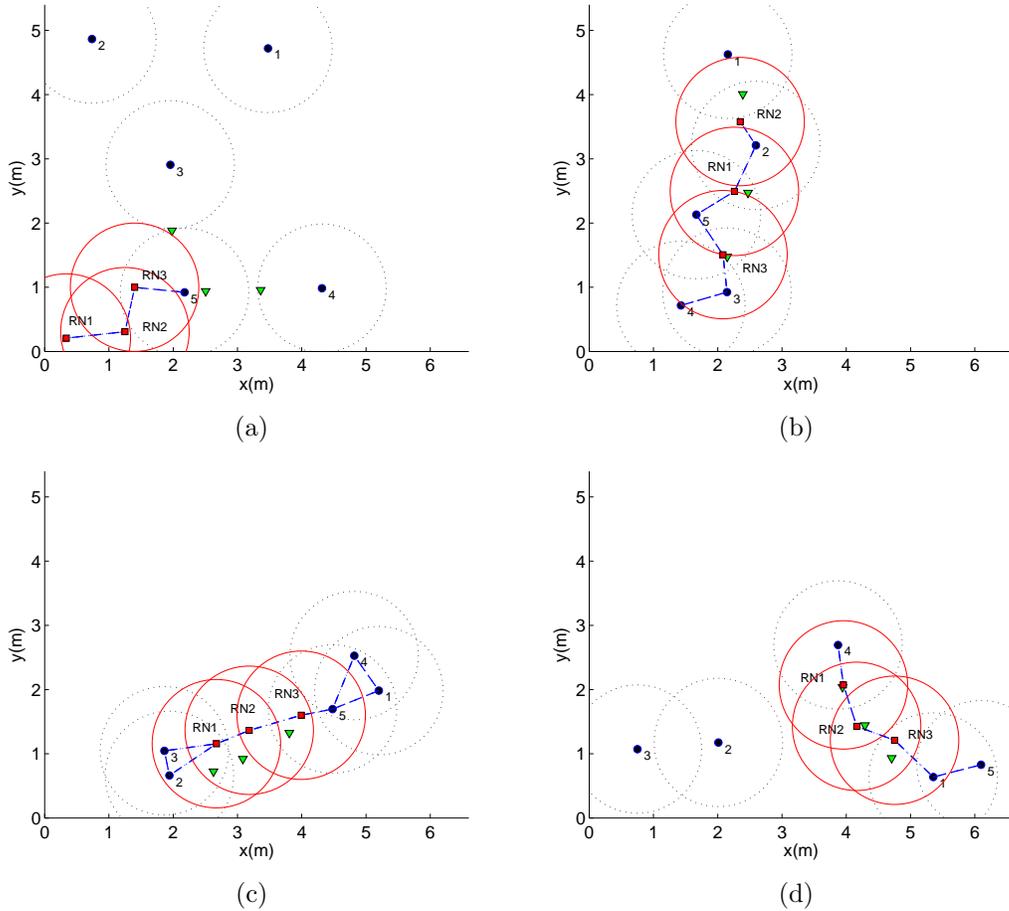


Figure 5: Illustration of the RN (red filled squares) movement over time when following the APs (green inverted filled triangles). Subfigures (a) and (d) show the initial and final positions of the UNs and RNs, respectively, and subfigures (b) and (c) depict intermediate locations.

Figure 5 shows a graphical example of how the RNs (red filled squares) move following the reference point APs (green inverted filled triangles), thus increasing the network connectivity. Subfigures 5(a) and 5(d) show the initial and final positions of the UNs and RNs, respectively, and subfigures 5(b) and 5(c) depict intermediate locations. The last three subfigures show how the system is able to adapt to the network topology changes in those situations such that the network is dispersed (subfigures 5(b) and 5(d)) or when the UNs are close to each other (subfigure 5(c)).

5. Evaluation and simulation results

To validate the efficiency of the DRNS proposal, we conduct several experiments for both simulated and real environments. In the current section, we evaluate the DRNS performance via simulation. In Section 6, we execute similar experiments in a real MANET where the nodes are robots. As we will see, such an experimentation corroborates the feasibility of the practical application of DRNS.

5.1. Simulation scenario

The chosen experimental simulation environment is based on Matlab, as is the case for the DKS solution [9]. Low-level network details are avoided because the system performance will be measured through the connectivity and throughput metrics. In order to directly compare DKS and DRNS, network connectivity is measured through Eq. (A.1) for $O_1(G)$, while throughput is measured using

$$th(G') = \sum_{\forall u_i, u_j \in U: j > i} ie(G'_i, i, j) \quad (7)$$

where G'_i corresponds to the minimum spanning tree starting at the i th UN. The function $ie(G'_i, i, j)$ measures the available throughput from nodes i to j in G'_i .

An important aspect to be considered when designing experiments is how to simulate the UN movements throughout a predefined area. To simulate a realistic MANET mobility scenario, three main aspects have to be addressed: the coverage radii, the UN and RN velocities, and the UN mobility pattern. Recall that the UN movements are not controlled.

The coverage radii are set to $1m$ to ensure network disconnections typically fall in a predefined area of $6.6m \times 5.4m$. Both UN and RN nodes are considered to have the same coverage radii, though the solution can be easily generalized considering different ones. This way, although it is common to have RNs with higher capabilities than UNs, like large coverage range, we think that the proposal should demonstrate good performance in more demanding situations. In particular, RNs with a smaller coverage range complicates the optimization problem, since a higher number of RNs is needed to provide the same communication capabilities.

The RNs velocity needs to be carefully selected to cover most realistic scenarios. The RN velocity should be equal to or higher than the UN velocity;

otherwise, the system will not be able to adapt to the changes in the network. This is a reasonable assumption in the real applications that we can think of, and imposes a constraint on the nature of the RNs that may be used in a given situation. An especially challenging situation is when the velocity of the RNs is equal to that of the UNs. We will check that situation by setting the velocity of all the nodes to $0.1m/ts$, where ts is the elapsed time in a *simulation time step*. On the other hand, a more adequate choice for the RNs would lead to a situation whereby the velocity of the RNs is above that of the UNs. To check that situation, we set the velocity of the RNs to $0.15m/ts$ and that of the UNs to $0.1m/ts$.

The third relevant aspect is the UN mobility pattern. Random Way Point (RWP) [54] and Reference Point Group Mobility (RPGM) [55] movement patterns have been chosen here to test the system behavior when dispersed and dense networks are used, respectively.

5.2. System performance and results discussion

Considering the aforementioned MANET scenario, Figures 6 and 7 show the cumulated mean achieved by DRNS and DKS (leftmost and rightmost subfigures, respectively) in terms of connectivity and throughput (subfigures at the top and bottom, respectively). The previous results are obtained by considering 25 repetitions using RWP mobility patterns to set the UN movements around the predefined area during the simulation. The network is composed of 3 UNs, with the number of RNs varying from 0 to 3. In addition, H and λ parameters have to be adequately set up. The first one might be chosen such that the prediction error remains low. In a similar way, the second one must ensure a sufficient number of APs for the initial distribution. In particular, we empirically conclude that $\lambda = 0.5$ ensures a good trade-off between the complexity introduced by the algorithm and the performance results obtained (see [4] for a better explanation of the impact of λ on the initial number of candidate APs and on the execution time) while higher or lower H values than 4 reduces the system performance. Additionally, we consider two different values for the RN velocity to observe the effects on the system performance. The chosen values are $0.1m/ts$ (see Figure 6) and $0.15m/ts$ (see Figure 7).

As observed, in all cases, DRNS outperforms DKS in both connectivity and throughput for 2 RNs or more, whereas both of them obtain similar performance values for 1 RN. This result demonstrates the limitation of DKS in utilizing all available RNs. On the contrary, DRNS always utilizes all

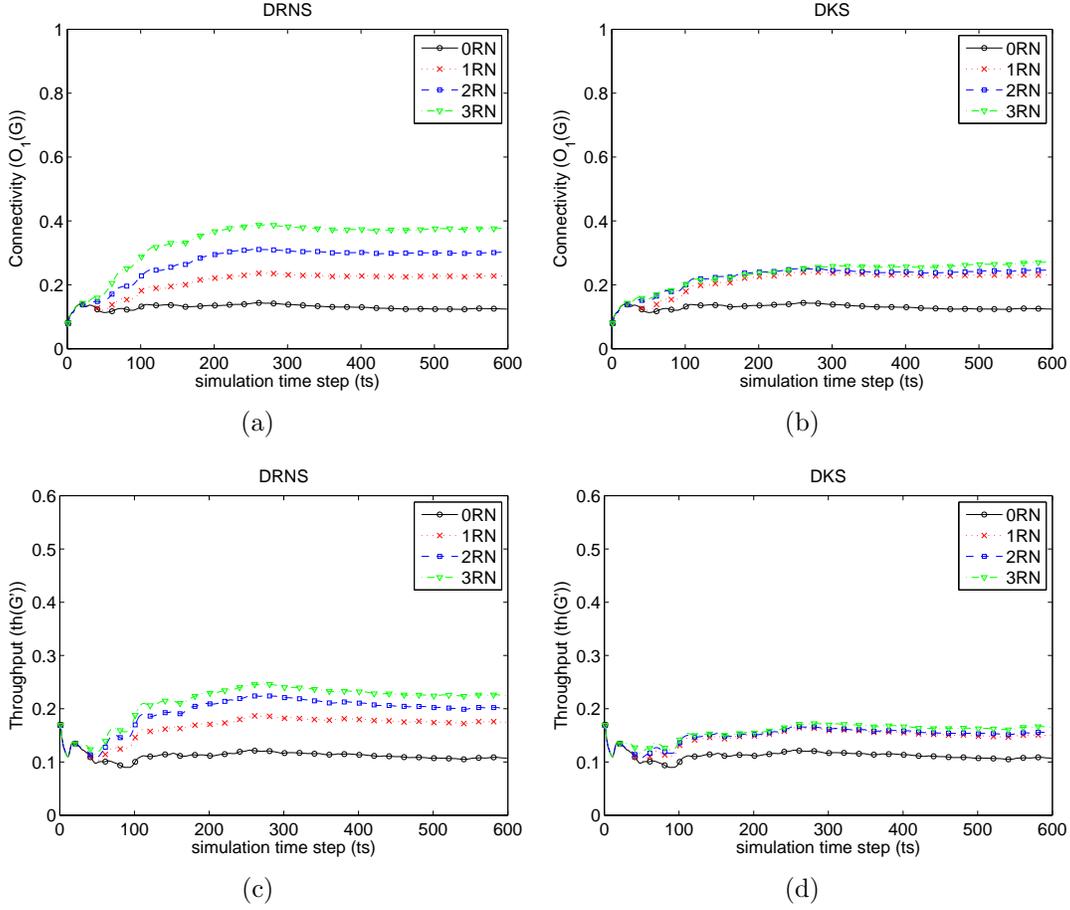


Figure 6: Evolution of the cumulative mean for the network connectivity and throughput by considering 3 UNs and with the number of RNs varying from 0 to 3, where the RWP mobility pattern is selected for the UN movements. The velocity for the RNs and UNs is fixed as $0.1m/ts$. Subfigures (a) and (b) show the connectivity results obtained by DRNS and DKS, respectively. The throughput values are depicted in subfigures (c) and (d) for the same approaches.

RNs. Both approaches achieve lower performance when RNs and UNs have the same velocity than when the velocity of the former is augmented. This is motivated by the fact that the system is not quick enough to follow the changes in the environment. Although this is an expected result, it should be remarked that the degradation suffered by DKS is higher. Therefore, DKS

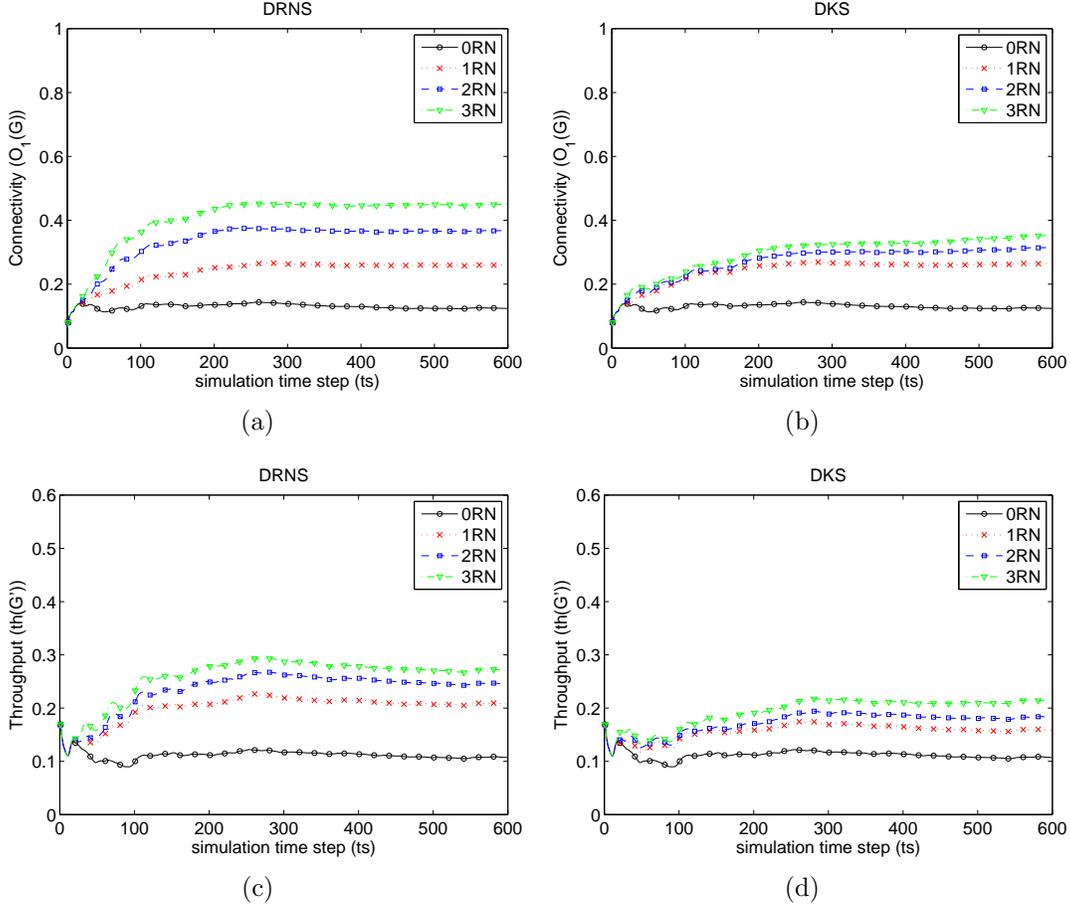


Figure 7: System performance when the RN velocity is increased in comparison with Figure 6 ($0.15m/ts$ instead of $0.1m/ts$). Then, 3 UNs are considered, and the number of RNs varies from 0 to 3. In addition, an RWP mobility pattern is used here for the UNs. Subfigures (a) and (b) show the connectivity results obtained when the DRNS and DKS approaches are selected, respectively. The throughput values are depicted in subfigures (c) and (d) for the same approaches.

leads the system to a similar performance regardless of the number of RNs used.

An additional experiment is shown in Figure 8, where we compare steady-state results for DRNS with a baseline solution that moves the RNs randomly throughout the area (RAND hereafter). With the aim of assessing the system for different scenarios RWP (Figures 8(a) and 8(b)) and RPGM (Figures

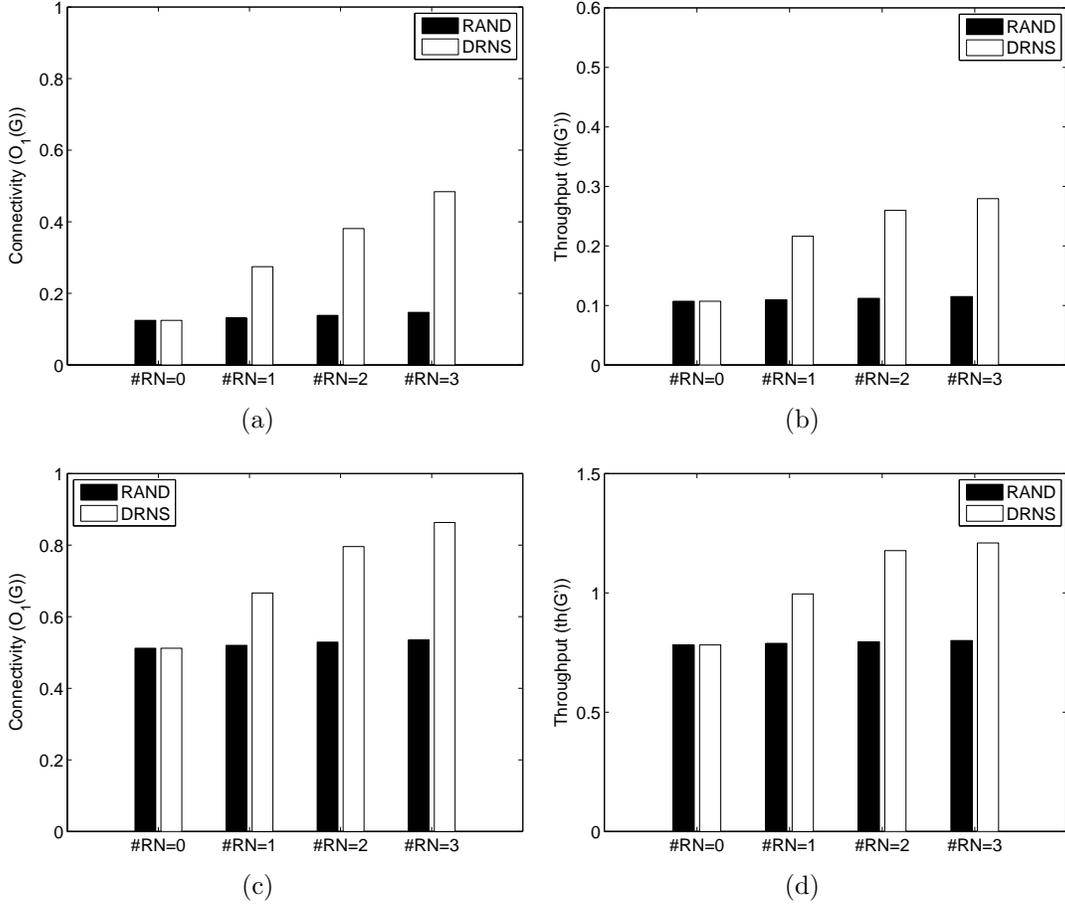


Figure 8: Performance of DRNS and RAND solutions in terms of connectivity and throughput for two different node movements patterns: RWP, (a) and (b); and RPGM, (c) and (d). In both cases the number of UNs is set to 3.

8(c) and 8(d)) node movements patterns are used for DRNS. In both cases DRNS outperforms the RAND solution where only a slight performance increment with the number of RNs is observed. This is mainly motivated by the increment in the number of total network nodes. Instead, DRNS drives the RNs efficiently and improves the network performance with increasing number of RNs.

Finally, aimed at evaluating the system behavior with a higher number of nodes, we consider up to 50 UNs and the number of RNs varying from 0 to 6. This way, Figures 9 and 10 show the results obtained for both network

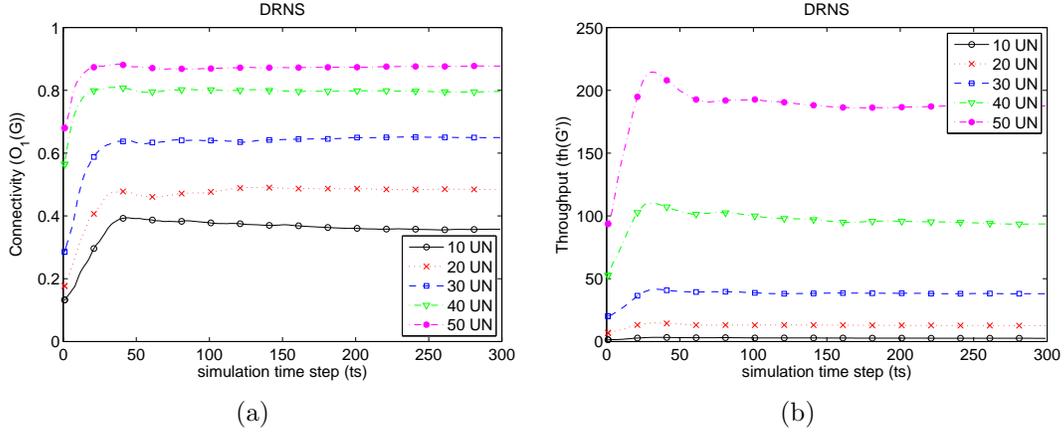


Figure 9: Connectivity and throughput behavior for 10 to 50 UNs and 3 RNs. The RWP mobility pattern is used for the UNs movement.

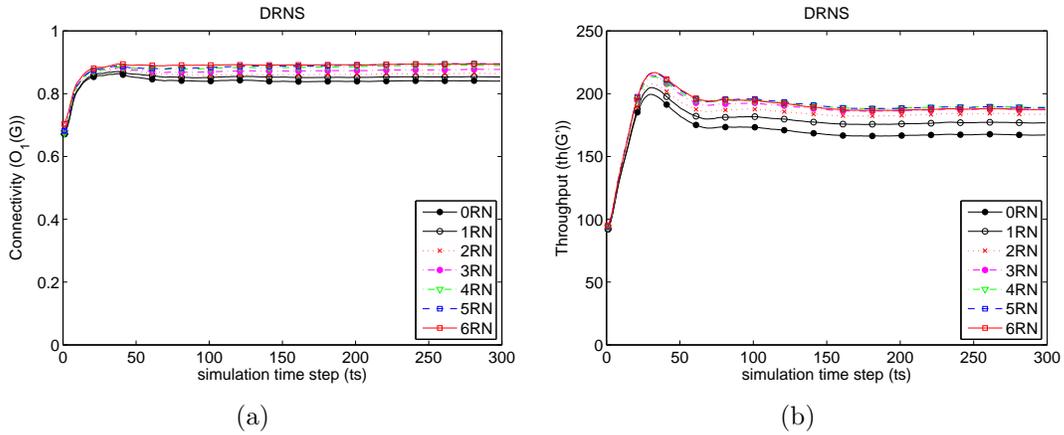


Figure 10: Connectivity and throughput behavior for 0 to 6 RNs and 50 UNs. The RWP mobility pattern is used for the UNs movement.

connectivity and throughput where the UNs follow a RWP mobility pattern. The first experiment (Figure 9), considers the number of UNs from 10 to 50 while the number of RNs remains fixed to 3. In this case, DRNS performs as expected: the system performance increases with increasing number of UNs. Another relevant observation is that DRNS is able to quickly reach a steady state as the number of UNs is increased. This is due to the lower number of partitions which makes to connect them easier. In the second experiment (Figure 10), the number of RNs varies from 0 to 6 while the number of UN

	$\mathcal{O}(\cdot)$
DKS	$\mathcal{O}((m \cdot k + n^2) \cdot p \cdot i)$
AP optimization	$\mathcal{O}((n^4 + m) \cdot p \cdot i)$
RN motion control	$\mathcal{O}(m^2 \cdot p \cdot i)$

Table 1: Computational complexity of each RN placement solution studied. (The expressions are obtained for the worst case, that is, when the network is totally disconnected)

remains fixed to 50. Similarly to the previous experiment, the performance increases with the number of RNs. However, adding extra RNs to the system does not necessarily leads to a notable performance improvements since the network is connected most of the time.

5.3. Execution time and computational complexity

Table 1 shows the computational complexity results in Big \mathcal{O} notation of the most relevant modules of DRNS: the AP optimization and the RN motion control modules. According to these results, one of the most influential parameters on the execution time is the number of UNs considered (*i.e.*, parameter n in the table). For this reason, we measure the execution time (elapsed time) of each module when the number of UNs is incremented. Figure 11 depicts the execution time evolution for each module, the whole DRNS system and for the DKS solution. Clearly, the DRNS execution time is conditioned by the AP optimization module. Such a behavior comes from the evaluation of the cost function $g(G')$ in the optimization block of this module (see reference [4] for a more detailed explanation).

In comparison with DKS, DRNS results in an increased computational load with increasing number of UNs due to the execution of the AP optimization module. Concerning DKS, its computational complexity is $\mathcal{O}((m \cdot k + n^2) \cdot p \cdot i)$, where $k = \frac{n \cdot (n-1)}{2}$ is the total number of APs computed when the network is totally disconnected. Remember that DKS locates one AP just in the middle point between two different partitions. If the number of UNs is incremented, the execution time is expected to be higher according to the term n^2 , which is obtained from the evaluation of the network connectivity ($O_1(G)$). Instead, as depicted in Figure 11(b), from a determined number of UNs, this time remains almost constant. Because the area does not change over time, a higher number of UNs leads to the network having fewer partitions. In the figure, we can see a similar behavior for the RN motion control module.

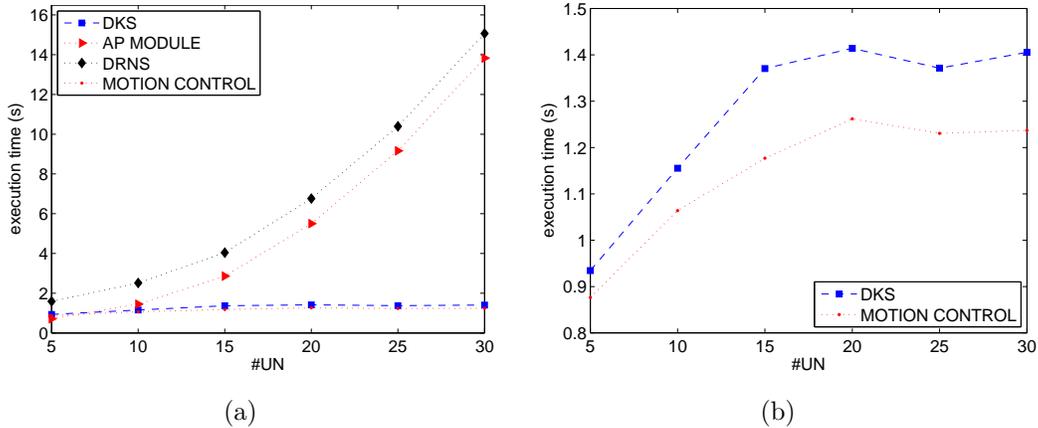


Figure 11: Execution time evolution with the number of UNs for every RN placement solution. Subfigure (a) shows the execution time evolution for both solutions: DRNS and DKS. In addition, the figure includes such an evolution obtained from the AP optimization module. Subfigure (b) shows a detailed comparison between DKS and the RN motion control module.

In summary, the elapsed execution time of DRNS is mainly influenced by the AP optimization module, as clearly observed in Figure 11. Additionally, the RN motion control module increases the computation time.

6. Real MANET scenario: IDSIA Swarm Robotics Laboratory

Most RN location approaches are validated through simulation environments and tools to reduce deployment effort, time and costs in comparison with real environments. Nevertheless, not all inherent aspects of real environments can be accurately simulated. Although simulation results can determine the efficacy and efficiency of a proposed solution at a first instance, its deployment in real scenarios can lead to the system obtaining very different results.

To compare the results in simulation and in real environments, we test the DRNS solution in a MANET scenario where the nodes are robots.

6.1. Environment description

Many aspects such as coverage radii, node movements, node control and communications, and physical aspects like node energy consumptions and obstacle avoidance, need to be considered before deploying the system in a real environment, especially when considering indoor MANET scenarios.

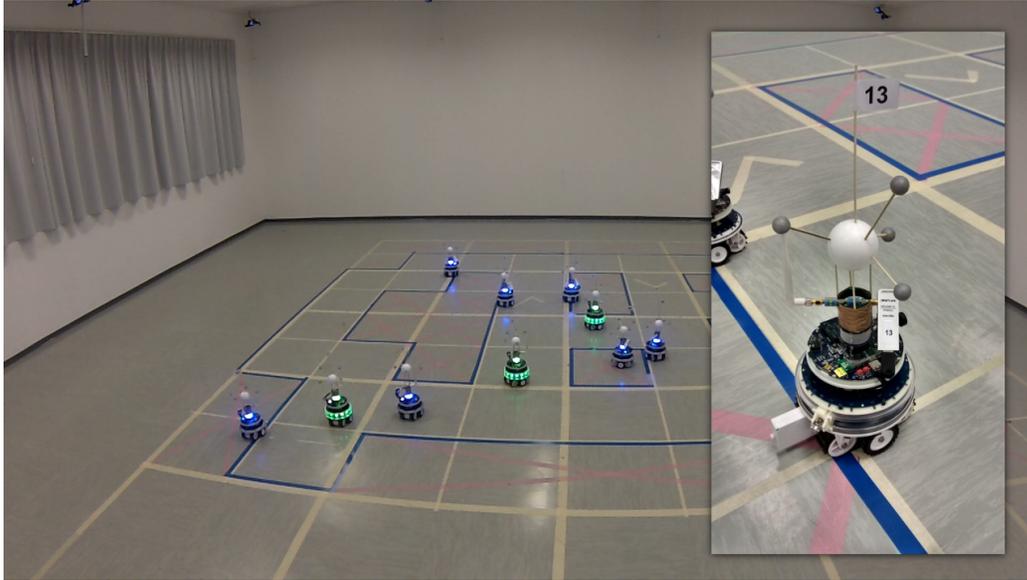


Figure 12: Snapshot of the real robot environment used in the experimentation. We can see 7 UNs (blue light) and 3 RNs (green light) whose locations are optimized.

In this work, we use a robotic scenario to evaluate our RN placement proposal so that the obtained results can be compared with those from the simulation and finally used to evidence the validity of the solution. Such a robotic scenario has been deployed at Institute Dalle Molle Institute for Artificial Intelligence (IDSIA) [56] and corresponds to an indoor laboratory: the *IDSIA Swarm Robotics Laboratory* [57]. This laboratory has an available area of $6.6m \times 5.4m$, which maintains a safety zone from the surrounding walls. Figure 12 shows a picture of the environment. In the same figure, on the right, a more detailed image of the robots is also shown (these robots are colloquially called *foot-bots* [58]).

A number of concerns arise in this real framework. The first concern is that of moving robots throughout the selected area. This movement implies considering the robots' dimensions, their current location and orientation, the location of the remaining robots, their velocities, and how to prevent obstacles (other robots as well as walls). Such issues have been addressed by integrating and deploying the navigation and collision avoidance algorithm in [59]. The main parameters to be customized in the algorithm are the node velocity and the surrounding safety collision zone. The first parameter was

selected from some experiments in the ARGoS simulator [60]: $0.1m/s$ for the UNs and $0.2m/s$ for the RNs. Because every robot has a circular shape with a diameter of $14cm$, we decided that a robot cannot be closer than $24cm$ to any other robot, that is, a safety collision zone of $5cm$ per robot is set. Moreover, RWP and RPGM are also used as mobility patterns for moving robots, as in simulation.

Another principal issue in real scenarios refers to the knowledge of each robot's location within the area. Such information is used, on the one hand, by DRNS to coordinately drive the RNs to the best locations according to the network performance goals and, on the other hand, by the algorithm in charge of the robot navigation and collision avoidance. To this end, a tracking system is deployed in the scenario. The tracking system is composed of a number of cameras homogeneously distributed, together with an array of infrared sensors. Figure 13 shows the complete system and modules involved in the deployment of the real robotic scenario to test the DRNS solution.

Each robot needs to know its current position in real time. For this, a *tracker interface* module (a procedure running inside a centralized computer) obtains the location information (*i.e.*, the (X,Y) coordinates) from the tracking system. Then, this module sends this information to each robot in a regular manner.

One more relevant issue is the temporal evolution of the overall system. In simulation, the DRNS engine is run every time step (ts). However, there is no equivalence between time steps in simulation and the time elapsed in real scenarios: the network does not change in simulation in the duration ts , whereas the nodes are moving continuously in the real environment. Thus, DRNS needs to be adaptable depending on the network dynamism.

Additionally, all the modules and components of the real deployment need to communicate with each other. For this, we choose LCM as the communication system. LCM stands for Lightweight Communications and Marshalling and refers to a publisher/subscriber message-passing technology [61]. LCM represents a reliable, scalable, low-latency and simple communications *middleware*. In LCM, two channels are defined to send messages: TRACK and TARGET. The first channel is used by the tracker interface module to publish the messages containing the location of the robots. This way, the robots are subscribed to this channel to determine their own location and the locations of the other robots. The TARGET channel communicates the new target-point-related messages for the UN and RN robots. The *DRNS engine* and *motion manager* modules use that channel to publish the target points

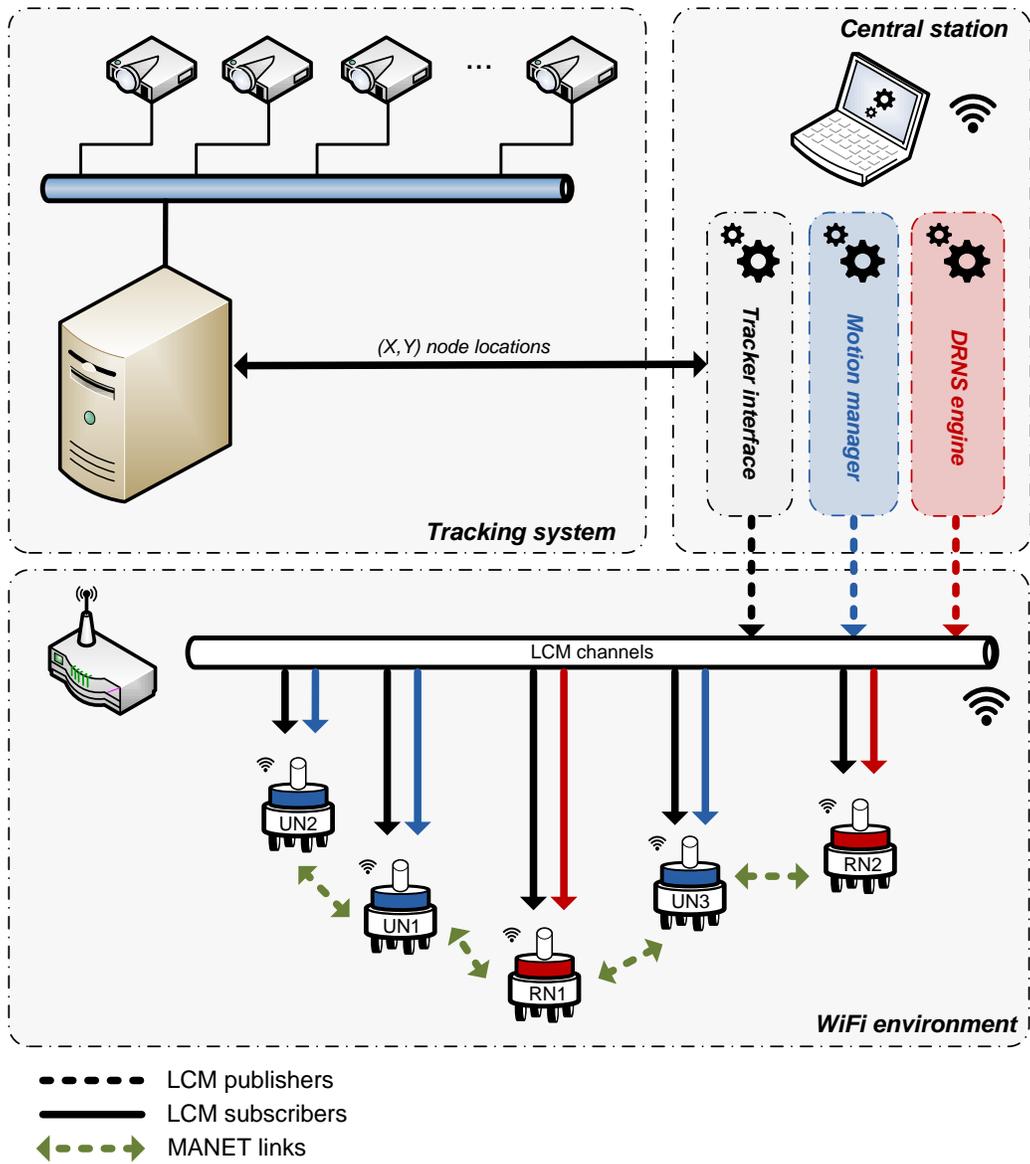


Figure 13: Overall functional architecture of the real scenario.

related to the RNs and UNs, respectively. These messages are then gathered by the corresponding robot acting as a subscriber.

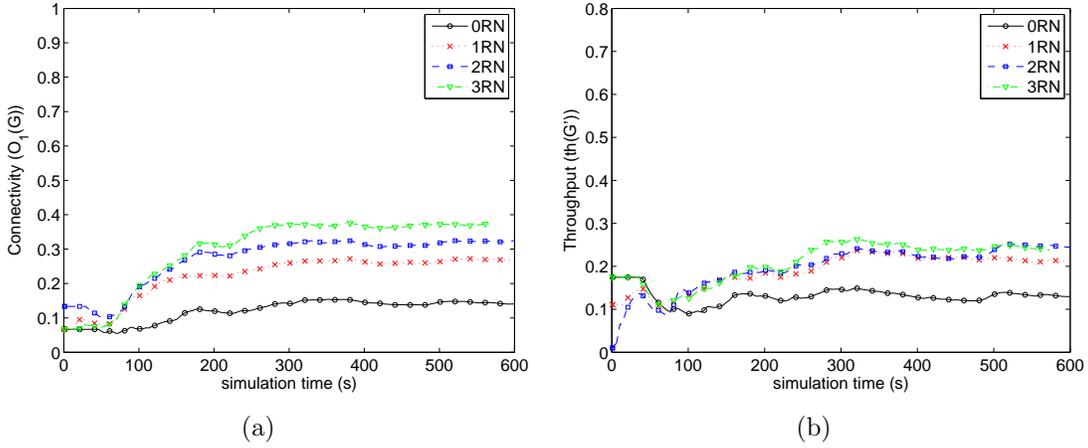


Figure 14: DRNS results obtained for the real environment. The number of RNs varies from 0 to 3, while the number of UNs is set to 3. The cumulative mean evolution of connectivity and throughput as a function of the number of RNs is depicted.

6.2. System performance and discussion of results

To evaluate the DRNS system performance in the real environment, we have replicated some of the experiments performed in simulation for comparison purposes. The main difference between both experiments is that we have considered 5 repetitions instead of the 25 used in simulation because running a real experiment has a high execution cost. On the other hand, the λ value is the same than in simulation, while the velocity of the RNs is set to $0.2m/s$, double than that of the UNs. As observed, we have slightly incremented the velocity of the RNs to compensate for the effects of acceleration and direction changes, which are not considered in simulation (see Section 5).

Figure 14 shows the connectivity and throughput evolution provided by the DRNS solution. We can observe a similar behavior than that in simulation: the system performance increases with the number of RNs. However, two main differences may be noticed in regard to the simulation results in Figure 6:

1. The system provides a slower adaptation to changes in the network topology.
2. The general performance values are lower than those obtained for simulation.

Clearly, the real scenario includes some unexpected elements not considered in the simulation case. These differences are mainly motivated by the fact that robots have to avoid obstacles such as other nodes and walls. Another aspect that affects the results obtained is the velocity of the robots. In simulation, this parameter is roughly approximated by fixed displacements in each time interval, whereas in a real environment, robots need to turn and accelerate while changing their direction of movement. In summary, the inherent characteristics of real scenarios result in a lower performance compared to simulation. Such restrictions are especially relevant in dense scenarios. This behavior is depicted in Figure 15, which show the system performance obtained by the DRNS and RAND solutions for the RWP (Figures 15(a) and 15(b)) and RPGM (Figures 15(c) and 15(d)) mobility patterns. In particular, in RPGM scenarios, the robots will more likely intercept other robots. This causes the improvement achieved with 3 RNs to be almost equal to that obtained with 2 RNs. A larger effect is obtained with a higher number of nodes.

In addition to the particular effects associated with physical environments, we must remark the validity of the DRNS approach. As a proof of not only the performance improvement achieved but also its actual usability, a video with some of the real experiment being conducted can be accessed at [62] as supplementary material.

7. Conclusions and future work

The present work addresses the RN placement problem in MANETs, where continuous topology changes occur due to the inherent mobility of the nodes. From that, a novel dynamical placement solution, named DRNS, is proposed. DRNS relies on a bi-objective optimization procedure that combines connectivity and throughput. The placement problem can be seen as two separated sub-problems: (a) where the RNs should be moved to, and (b) how they should be moved. To address both issues, we logically split the problem into several modules, which, on the one hand, optimize the target positions of the relays and, on the other hand, move the relays to the previous locations in a controlled and optimized manner. Such a modular division adds flexibility and scalability to the placement solution.

Despite the experimental benefits of DRNS, it is demonstrated to increase the computation time. The number of UNs (n) seems a relevant parameter because the higher the number of UNs, the higher the execution time. This

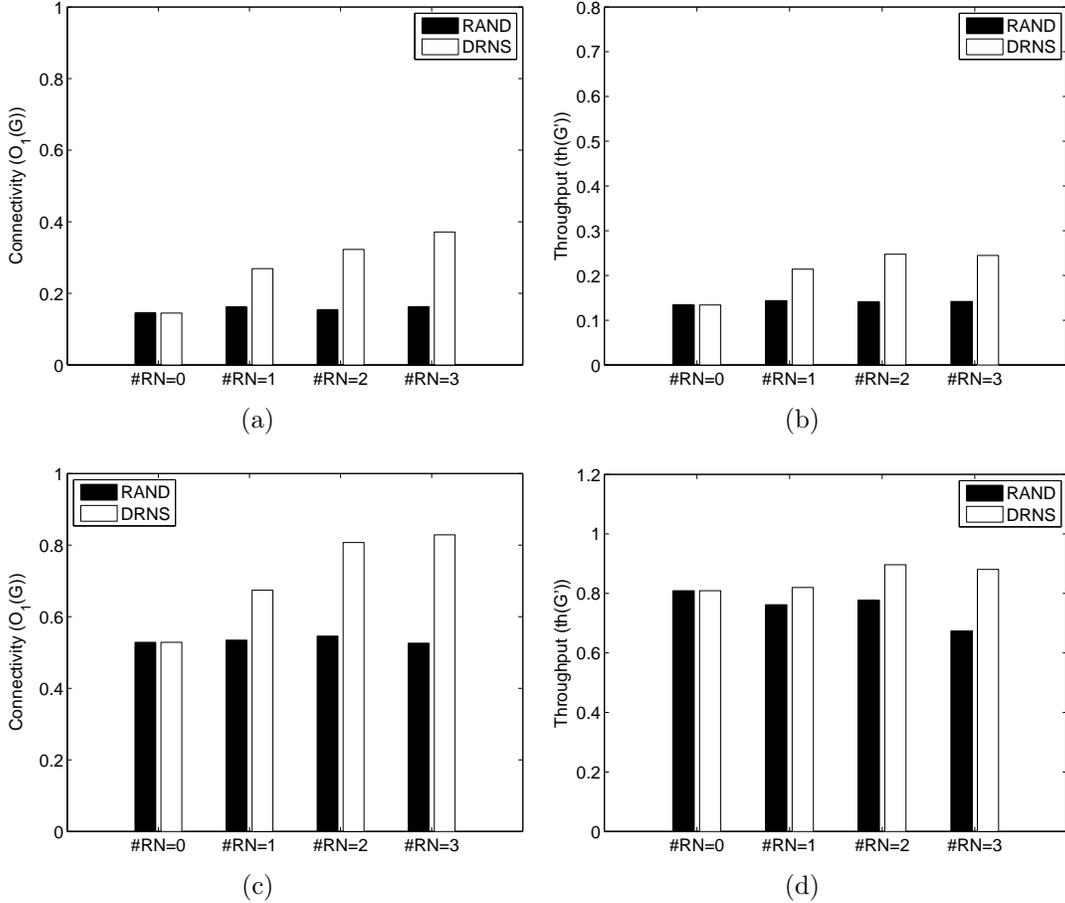


Figure 15: Performance of DRNS and RAND solutions in terms of connectivity and throughput for two different node movements patterns in the real robotic scenario: RWP, (a) and (b); and RPGM, (c) and (d). In both cases the number of UNs is set to 3.

fact can be a hindrance depending on the final application of the solution. For instance, DRNS is a valid scheme for restoring the lost connectivity in emergency rescue or natural disaster situations where the number of network nodes is not high.

As future work to further improve the proposal, we could develop a distributed version of the placement scheme to add versatility, self-managing capacities and greater resiliency against faults. Moreover, the inclusion in the optimization function of new real network metrics, such as PDR, and avoiding crowded node situations in the network where signal interference

and collisions are expected could prevent potential counter-productive RN placement solutions.

Acknowledgments

This work has been partially supported by Spanish MICINN (Ministerio de Ciencia e Innovación) through project TEC2011-22579, by Spanish MINECO (Ministerio de Economía y Competitividad) through project TIN2014-60346-R, and FEDER funds and the FPU P6A grants program of the University of Granada.

Appendix. DKS placement solution

DKS [9] assumes that, in a wireless environment, *e.g.* MANETs, any two nodes are accessible or connected (that is, there is a link between them) if the Euclidean distance between them is less or equal to c , where c is the coverage range of a radio node.

Based on this, DKS attempts to maximize two performance objectives: (i) the overall network connectivity, and (ii) the throughput. This optimization process is achieved through the use of a PSO algorithm [12] and several cost functions. The PSO algorithm obtains two particular entries: (i) the future motion predictions of UNs for a receding horizon ($t + H$), and (ii) the best solution obtained in the previous time step (t). PSO provides the placement for the RNs in ($t + 1$) by generating a set of potential solutions, referred to as *particles*.

Three objective functions are jointly used within the optimization procedure. Through the first one, named $O_1(G)$, the global network connectivity is evaluated. $O_1(G)$ is defined as follows:

$$O_1(G) = \frac{2 \times \sum_{i,j \in U: j > i} z_{ij}}{|U| \times (|U| - 1)} \quad (\text{A.1})$$

where U corresponds to the set of UNs in G , $|U|$ is the number of UNs (cardinality of the set U), and $z_{ij} = 1$ when the i th and j th nodes are connected (directly or indirectly through a *multi-hop* path); $z_{ij} = 0$, otherwise. The value of $O_1(G)$ depends on the number of interconnected pairs of nodes, its extreme values being 0 and 1 for completely disconnected and connected networks, respectively.

A second function, $O_2(G)$, computes the minimum network throughput. $O_2(G)$ is only evaluated when several candidate solutions in the optimization procedure (several particles under evaluation in the PSO) represent completely connected networks (*i.e.*, $O_1(G) = 1$ for each particle). $O_2(G)$ is described as follows:

$$O_2(G) = \min_{i,j \in U: j > i} \{T(G, i, j) : T(G, i, j) > 0\} \quad (\text{A.2})$$

where $T(G, i, j)$ corresponds to the maximum *throughput* between the i th and j th nodes in the network G . DKS approximates the throughput of a determined link (w_{ij}) between two adjacent nodes i and j as a function of their distance (d_{ij}). Note that $O_2(G)$ becomes very time consuming with the increase of the number of network nodes. DKS addresses this issue through a Minimum Spanning Tree (MST)-based clustering procedure. Following this procedure, the minimum throughput is obtained as the minimum-capacity cut-set among the clusters. However, in low density networks, this clustering procedure is equivalent to obtaining the throughput of the weakest link in the MST, which is much less time consuming than computing the clustering. For this reason, we skip the clustering computation in the results of this paper.

MANETs suffer from disconnections that lead to the network becoming partitioned. For this reason, to address a situation wherein the network is disconnected (*i.e.*, $O_1(G) < 1$), an alternative function $O_3(G)$ is considered in the PSO instead of $O_2(G)$. $O_3(G)$ measures the distance from each RN to the imaginary middle points, or *attraction points* (APs), among network partitions so that distances are minimized by locating the RNs as close as possible to the APs. In other words, by minimizing $O_3(G)$, the RNs are moved or attracted to the APs. $O_3(G)$ is defined as follows:

$$O_3(G) = \min_{i \in R, j \in A} \left\{ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right\} \quad (\text{A.3})$$

where i is the i th RN located at the coordinates (x_i, y_i) , j is the j th AP located at the coordinates (x_j, y_j) , and R and A represent the RN and AP sets, respectively.

In summary, the PSO algorithm of DKS follows a selective procedure to obtain the best location for each RN from a set of possible solutions or particles. First, the solution with the highest $O_1(G)$ value ($\max \{O_1(G)\}$) is chosen as the best solution. If there are several solutions corresponding to completely connected situations (alternative particles with $O_1(G) = 1$), the

one with the highest $O_2(G)$ value ($\max \{O_2(G)\}$) is selected. In the case of disconnected networks ($O_1(G) < 1$) with equal values for $O_1(G)$, the solution with the lowest $O_3(G)$ value ($\min \{O_3(G)\}$) is chosen instead.

References

- [1] M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks: A survey, *Ad Hoc Networks* 6 (4) (2008) 621–655. doi:10.1016/j.adhoc.2007.05.003.
- [2] Z.-E.-H. Kamal, M. A. Salahuddin, Introduction to Wireless Sensor Networks, in: D. Benhaddou, A. Al-Fuqaha (Eds.), *Wireless Sensor and Mobile Ad-Hoc Networks*, Springer New York, 2015, pp. 3–32. doi:10.1007/978-1-4939-2468-4_1.
- [3] M. Ayyash, Y. Alsbou, M. Anan, Introduction to Mobile Ad-Hoc and Vehicular Networks, in: D. Benhaddou, A. Al-Fuqaha (Eds.), *Wireless Sensor and Mobile Ad-Hoc Networks*, Springer New York, 2015, pp. 33–46. doi:10.1109/SURV.2009.090106.
- [4] R. Magán-Carrión, R. A. Rodríguez-Gómez, J. Camacho, P. García-Teodoro, Optimal relay placement in multi-hop wireless networks, *Ad Hoc Networks* 46 (2016) 23–36. doi:10.1016/j.adhoc.2016.03.007.
- [5] A. Kashyap, S. Khuller, M. Shayman, Relay placement for fault tolerance in wireless networks in higher dimensions, *Computational Geometry* 44 (4) (2011) 206–215. doi:10.1016/j.comgeo.2010.11.002.
- [6] D. Yang, S. Misra, X. Fang, G. Xue, J. Zhang, Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations, *IEEE Transactions on Mobile Computing* 11 (8) (2012) 1399–1411. doi:10.1109/TMC.2011.126.
- [7] H. Liu, X. Chu, Y. W. Leung, R. Du, Simple movement control algorithm for bi-connectivity in robotic sensor networks, *IEEE Journal on Selected Areas in Communications* 28 (7) (2010) 994–1005. doi:10.1109/JSAC.2010.100904.
- [8] A. Konak, O. Dengiz, A. E. Smith, Improving Network Connectivity in Ad Hoc Networks Using Particle Swarm Optimization and Agents, in:

Wireless Network Design, International Series in Operations Research & Management Science, Springer, New York, NY, 2011, pp. 247–267. doi:10.1007/978-1-4419-6111-2_11.

- [9] O. Dengiz, A. Konak, A. E. Smith, Connectivity management in mobile ad hoc networks using particle swarm optimization, *Ad Hoc Networks* 9 (7) (2011) 1312–1326. doi:10.1016/j.adhoc.2011.01.010.
- [10] H. Safa, W. El-Hajj, H. Zoubian, Particle swarm optimization based approach to solve the multiple sink placement problem in wsns, in: *Proceedings of the IEEE International Conference on Communications (ICC) - Wireless Network Symposium*, Ottawa, 2012, pp. 5445–5450.
- [11] E. F. Camacho, C. Bordons, *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing, 2nd Edition, Series in Statistics, Springer London, 2007.
- [12] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: *Proceedings of the the 6th. International Symposium on Micro Machine and Human Science*, Nagoya, 1995, pp. 39–43.
- [13] Y. Jin, J. Branke, Evolutionary optimization in uncertain environments—a survey, *IEEE Transactions on Evolutionary Computation* 9 (3) (2005) 303–317. doi:10.1109/TEVC.2005.846356.
- [14] N. Aziz, A. W. Mohemmed, M. Y. Alias, A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram, in: *Proceedings of the International Conference on Networking, Sensing and Control (ICNSC)*, Okayama, 2009, pp. 602–607.
- [15] A. Barolli, F. Xhafa, M. Takizawa, Optimization problems and resolution methods for node placement in wireless mesh networks, in: *Proceedings of the 14th. International Conference on Network-Based Information Systems (NBiS)*, Tirana, 2011, pp. 126–134.
- [16] H. Outchitachen, A. Hair, N. Idrissi, Joint mission and communication aware node placement problem in mission-specific mobile sensor networks, in: *5th Workshop on Codes, Cryptography and Communication Systems (WCCCS)*, 2014, pp. 1–4.

- [17] W. Zhang, G. Xue, S. Misra, Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms, in: Proceedings of the 26th. IEEE International Conference on Computer Communications (INFOCOM), Anchorage, 2007, pp. 1649–1657.
- [18] S. Misra, S. D. Hong, G. Xue, J. Tang, Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements, in: Proceedings of the 26th. IEEE International Conference on Computer Communications (INFOCOM), 2008, pp. 879–887.
- [19] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, D. Rus, Deploying sensor networks with guaranteed fault tolerance, *IEEE/ACM Transactions on Networking* 18 (1) (2010) 216–228. doi:10.1109/TNET.2009.2024941.
- [20] X. Han, X. Cao, E. L. Lloyd, C. S. Shen, Fault-tolerant relay node placement in heterogeneous wireless sensor networks, *IEEE Transactions on Mobile Computing* 9 (5) (2010) 643–656. doi:10.1109/TMC.2009.161.
- [21] A. Abbas, M. Younis, Establishing connectivity among disjoint terminals using a mix of stationary and mobile relays, *Computer Communications* 36 (13) (2013) 1411–1421. doi:10.1016/j.comcom.2013.06.004.
- [22] A. Chelli, M. Baga, D. Djenouri, I. Balasingham, T. Taleb, One-step approach for two-tiered constrained relay node placement in wireless sensor networks, *IEEE Wireless Communications Letters* 5 (4) (2016) 448–451.
- [23] C. Ma, W. Liang, M. Zheng, H. Sharif, A connectivity-aware approximation algorithm for relay node placement in wireless sensor networks, *IEEE Sensors Journal* 16 (2) (2016) 515–528.
- [24] M. Baga, A. Chelli, D. Djenouri, T. Taleb, I. Balasingham, K. Kansanen, Optimal placement of relay nodes over limited positions in wireless sensor networks, *IEEE Transactions on Wireless Communications* (2017) 1–13.
- [25] A. Bari, A. Jaekel, J. Jiang, Y. Xu, Design of fault tolerant wireless sensor networks satisfying survivability and lifetime requirements, *Computer Communications* 35 (3) (2012) 320–333. doi:10.1016/j.comcom.2011.10.006.

- [26] P. Ladosz, H. Oh, W. H. Chen, Optimal positioning of communication relay unmanned aerial vehicles in urban environments, in: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), 2016, pp. 1140–1147. doi:10.1109/ICUAS.2016.7502562.
- [27] A. Kashyap, M. Shayman, Relay placement and movement control for realization of fault-tolerant and hoc networks, in: 41st. Annual Conference on Information Sciences and Systems, Baltimore, 2007, pp. 783–788.
- [28] S. Perumal, J. Baras, C. Graff, D. Yee, Aerial platform placement algorithms to satisfy connectivity, capacity and survivability constraints in wireless ad-hoc networks, in: Military Communications Conference (MILCOM), San Diego, 2008, pp. 1–7.
- [29] A. Alfaqdhly, U. Baroudi, M. Younis, Optimal node repositioning for tolerating node failure in wireless sensor actor network, in: Proceedings of the 25th. Biennial Symposium on Communications, Kingston, 2010, pp. 67–71.
- [30] A. Alfaqdhly, U. Baroudi, M. Younis, Least distance movement recovery approach for large scale wireless sensor and actor networks, in: 7th. International Conference on Wireless Communications and Mobile Computing Conference (IWCMC), Istanbul, 2011, pp. 2058–2063.
- [31] M. Younis, S. Lee, A. Abbasi, A localized algorithm for restoring internode connectivity in networks of moveable sensors, *IEEE Transactions on Computers* 59 (12) (2010) 1669–1682. doi:10.1109/TC.2010.174.
- [32] A. Roy, S. Misra, P. Kar, A. Mondal, Topology Control for Self-Adaptation in Wireless Sensor Networks with Temporary Connection Impairment, *ACM Trans. Auton. Adapt. Syst.* 11 (4) (2017) 21:1–21:34. doi:10.1145/2979680.
- [33] I. F. Senturk, K. Akkaya, S. Yilmaz, Relay placement for restoring connectivity in partitioned wireless sensor networks under limited information, *Ad Hoc Networks* 13 (2014) 487–503. doi:10.1016/j.adhoc.2013.09.005.
- [34] C. Lin, Y. Li, D. Deng, A bat-inspired algorithm for router node placement with weighted clients in wireless mesh networks, in: 9th. Interna-

- tional Conference on Communications and Networking in China, Maoming, 2014, pp. 139–143.
- [35] X. Cheng, D. Z. Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks, *Wireless Networks* 14 (3) (2008) 347–355. doi:10.1007/s11276-006-0724-8.
 - [36] F. Senel, M. Younis, Relay node placement in structurally damaged wireless sensor networks via triangular steiner tree approximation, *Computer Communications* 34 (16) (2011) 1932–1941. doi:10.1016/j.comcom.2011.05.010.
 - [37] A. Nigam, Y. K. Agarwal, Optimal relay node placement in delay constrained wireless sensor network design, *European Journal of Operational Research* 233 (1) (2014) 220–233. doi:10.1016/j.ejor.2013.08.031.
 - [38] M. X. Cheng, Y. Ling, B. M. Sadler, Network connectivity assessment and improvement through relay node deployment, *Theoretical Computer Science* 660 (2017) 86–101. doi:10.1016/j.tcs.2016.11.029.
 - [39] H. Z. Abidin, N. Din, Provisioning wsn coverage via minimax based sensor node placement scheme, in: *Proceedings of the International Conference on Wireless Communications and Applications*, Kuala Lumpur, 2012, pp. 1–5.
 - [40] B. Wang, H. Xu, W. Liu, H. Liang, A novel node placement for long belt coverage in wireless networks, *IEEE Transactions on Computers* 62 (12) (2013) 2341–2353. doi:10.1109/TC.2012.145.
 - [41] K. Xu, Q. Wang, H. Hassanein, G. Takahara, Optimal wireless sensor networks (wsns) deployment: Minimum cost with lifetime constraint, in: *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2005, pp. 454–461.
 - [42] Q. Wang, G. Takahara, H. Hassanein, K. Xu, On relay node placement and locally optimal traffic allocation in heterogeneous wireless sensor networks, in: *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, Sidney, 2005, pp. 657–664.

- [43] D. Rosário, J. A. Filho, D. Rosário, A. Santosy, M. Gerla, A relay placement mechanism based on UAV mobility for satisfactory video transmissions, 2017, pp. 1–8. doi:10.1109/MedHocNet.2017.8001638.
- [44] J. Braga, A. Alessandretti, A. P. Aguiar, J. Sousa, A feedback motion strategy applied to a UAV to work as an autonomous relay node for maritime operations, in: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 625–632. doi:10.1109/ICUAS.2017.7991434.
- [45] G. Ishigaki, N. Shinomiya, Controller placement algorithm to alleviate burdens on communication nodes, in: International Conference on Computing, Networking and Communications (ICNC), Workshop on CNC, 2016, pp. 3549–3553.
- [46] S. Misra, N. Majd, H. Huang, Approximation algorithms for constrained relay node placement in energy harvesting wireless sensor networks, *IEEE Transactions on Computers* 63 (12) (2017) 2933–2947.
- [47] C. Yang, K. Chin, On nodes placement in energy harvesting wireless sensor networks for coverage and connectivity, *IEEE Transactions on Industrial Informatics* 13 (1) (2017) 27–36.
- [48] C. Ma, W. Liang, M. Zheng, Delay constrained relay node placement in two-tiered wireless sensor networks: A set-covering-based algorithm, *Journal of Network and Computer Applications* 93 (2017) 76–90. doi:10.1016/j.jnca.2017.05.004.
- [49] L. Wang, X. Fu, J. Fang, H. Wang, M. Fei, Optimal node placement in industrial wireless sensor networks using adaptive mutation probability binary particle swarm optimization algorithm, in: Proceedings of the 7th. International Conference on Natural Computation (ICNC), Shanghai, 2011, pp. 2199–2203.
- [50] E. F. Flushing, G. A. D. Caro, A flow-based optimization model for throughput-oriented relay node placement in wireless sensor networks, in: Proceedings of the 28th. Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 2013, pp. 632–639.

- [51] Y. Zeng, R. Zhang, T. J. Lim, Throughput Maximization for UAV-Enabled Mobile Relaying Systems, *IEEE Transactions on Communications* 64 (12) (2016) 4983–4996. doi:10.1109/TCOMM.2016.2611512.
- [52] V. Sharma, K. Srinivasan, H.-C. Chao, K.-L. Hua, W.-H. Cheng, Intelligent deployment of UAVs in 5g heterogeneous communication environment for improved coverage, *Journal of Network and Computer Applications* 85 (2017) 94–105. doi:10.1016/j.jnca.2016.12.012.
- [53] Y. Shi, R. C. Eberhart, Parameter selection in particle swarm optimization, in: *Proceedings of the 7th. International Conference on Evolutionary Programming VII*, San Diego, California, USA, 1998, pp. 591–600.
- [54] D. B. Johnson, D. A. Maltz, Dynamic source routing in ad hoc wireless networks, in: *Mobile Computing*, The Kluwer International Series in Engineering and Computer Science., Springer US, 1996, pp. 153–181. doi:10.1007/978-0-585-29603-6_5.
- [55] X. Hong, M. Gerla, G. Pei, C. C. Chiang, A group mobility model for ad hoc wireless networks, in: *Proceedings of the 2nd. ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 1999, pp. 53–60.
- [56] IDSIA, Institute dalle molle for artificial intelligence, [Online; Accessed 20-April-2017] <http://www.idsia.ch>.
- [57] IDSIA, IDSIA Swarm Robotics Lab, [Online; Accessed 20-April-2017] <http://robotics.idsia.ch/>.
- [58] EPFL (École Polytechnique Fédérale de Lausanne), Marxbot project, [Online; Accessed 20-April-2017] <http://robots.epfl.ch/marxbot.html>.
- [59] J. Guzzi, A. Giusti, L. Gambardella, G. Theraulaz, G. Di Caro, Human-friendly robot navigation in dynamic environments, in: *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, Karlsruhe, 2013, pp. 423–430.
- [60] C. Pinciroli, V. Trianni, R. O’Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. D. Caro, F. Ducatelle, M. Birattari, L. M.

Gambardella, M. Dorigo, ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems, *Swarm Intelligence* 6 (4) (2012) 271–295. doi:10.1007/s11721-012-0072-5.

- [61] A. S. Huang, E. Olson, D. C. Moore, Lcm: Lightweight communications and marshalling, in: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, 2010, pp. 4057–4062.
- [62] R. Magán-Carrión, J. Camacho, P. García-Teodoro, E. F. Flushing, G. A. D. Caro, DRNS: Dynamical Relay Node Placement Solution, in: *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, Springer, Cham, 2016, pp. 273–276, [Demo online; Accessed 20-April-2017] http://youtu.be/mW1Q_MUFYs4.