ORIGINAL ARTICLE

Open Access

Router and gateway node placement in wireless mesh networks for emergency rescue scenarios

Mariusz Wzorek^{1*} , Cyrille Berger¹ and Patrick Doherty^{1,2}

Abstract

The focus of this paper is on base functionalities required for UAV-based rapid deployment of an ad hoc communication infrastructure in the initial phases of rescue operations. The main idea is to use heterogeneous teams of UAVs to deploy communication kits that include routers, and are used in the generation of ad hoc Wireless Mesh Networks (WMN). Several fundamental problems are considered and algorithms are proposed to solve these problems. The Router Node Placement problem (RNP) and a generalization of it that takes into account additional constraints arising in actual field usage is considered first. The RNP problem tries to determine how to optimally place routers in a WMN. A new algorithm, the RRT-WMN algorithm, is proposed to solve this problem. It is based in part on a novel use of the Rapidly Exploring Random Trees (RRT) algorithm used in motion planning. A comparative empirical evaluation between the RRT-WMN algorithm and existing techniques such as the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and Particle Swarm Optimization (PSO), shows that the RRT-WMN algorithm has far better performance both in amount of time taken and regional coverage as the generalized RNP problem scales to realistic scenarios. The Gateway Node Placement Problem (GNP) tries to determine how to locate a minimal number of gateway nodes in a WMN backbone network while satisfying a number of Quality of Service (QoS) constraints. Two alternatives are proposed for solving the combined RNP-GNP problem. The first approach combines the RRT-WMN algorithm with a preexisting graph clustering algorithm. The second approach, WMNbyAreaDecomposition, proposes a novel divide-and-conquer algorithm that recursively partitions a target deployment area into a set of disjoint regions, thus creating a number of simpler RNP problems that are then solved concurrently. Both algorithms are evaluated on real-world GIS models of different size and complexity. WMNbyAreaDecomposition is shown to outperform existing algorithms using 73% to 92% fewer router nodes while at the same time satisfying all QoS requirements.

Keywords: Robotics, UAV deployed ad hoc networks, Wireless mesh networks, Router node placement, Gateway node placement, Emergency rescue

1 Introduction

The importance of effective communication and efficient data/knowledge transfer is essential for the coordination of life-saving activities in regions affected by natural or man-made disasters. The organizations and groups involved range from disaster relief, governmental

and non-governmental organizations at the macro level, to actual teams of emergency rescue responders on the ground at the micro-level. Emergency rescue teams have also recently been supported by heterogeneous robotic assistants. As can be observed, there has been an increasing number of natural disasters that include wildfires, hurricanes, earthquakes, and floods that require a state-of-the-art emergency response to minimize loss of life and property damage.

*Correspondence: mariusz.wzorek@liu.se

¹Department of Computer and Information Science, Linköping University, Linköping, Sweden

Full list of author information is available at the end of the article

In a seminal article, Denning [1], pointed out the importance of establishing *Hastily Formed Networks* (HFNs) in the broader sense as “the ability to form multi-organizational networks rapidly” and as being crucial to disaster relief. Here, Denning’s focus was on effective human communication and the use of autonomous systems in emergency rescue operations was not yet prevalent. Additionally, the term *conversation space* was introduced for the medium in which such communication takes place.

The original concept was very much focused on setting up the physical layer for communication in order to support the conversation space. Proposed components of the conversational space were the physical systems, the players involved, and the interaction practices. The latter include situation awareness, acquiring and sharing information, planning, making decisions, coordination, and command and control required by the players and teams involved. All these practices are essential to modern-day emergency support missions.

In a report describing the international rescue efforts after the 2010 earthquake in Haiti [2], the authors emphasized that within one hour of the earthquake all telecommunication systems were totally inoperative and mobile phone networks did not fully recover until after the formal end of the rescue operation. Initially, satellite phones were used, but the satellite communication became saturated and no phone calls could be made or received. “Reliable communication with the teams in the field was unpredictable during the entire operation and this caused difficulties in planning. In addition, and more importantly, it involved a security risk since many teams could not notify the Operations Cell (OPC) of their whereabouts or of possible security threats.”

The work presented in this paper is part of a larger infrastructural framework being developed with the

goal of leveraging the use of heterogeneous teams of Unmanned Aerial Vehicles (UAVs) in providing services and supplies to rescuers on the ground in collaboration with other human resources in disaster relief scenarios [3]. In many disaster scenarios such as earthquakes, floods or hurricanes, existing communication infrastructures are knocked out either partially or wholly. Therefore, one of the first tasks at hand is to rapidly restore or set up ad hoc communication networks in affected areas. In particular, this is prioritized in those areas where rescuers have to rapidly enter in order to acquire first-response situation awareness and convey this information to operations centers for additional decision support and mission planning.

The focus of this paper is on the base functionalities required for UAV-based rapid deployment of an ad hoc communication infrastructure in the initial phases of rescue operations. Figure 1 presents an example of such a mission scenario where a fleet of UAVs are tasked to deliver a set of communication nodes which would serve as the physical communication infrastructure for a specific region. The idea is that teams of UAVs can be used to dynamically deliver router nodes via air delivery that are then used in an ad hoc network. A proof of concept design of a UAV deliverable communication node prototype has been developed for this purpose [4]. It also includes efficient deployment technology based on autonomous release mechanisms and airdrop by parachute.

Assuming access to this technology, one important research question that is targeted in this paper, is in determining the optimal or semi-optimal placement of communication nodes in a physical region that can form a viable ad hoc wireless mesh network. Wireless Mesh Networks (WMNs) are an emerging technology that is well suited for applications where ad hoc network infrastructure needs to be deployed rapidly. WMNs offer advantages

Fig. 1 Example mission scenario for deployment of an ad hoc communication network in search and rescue environment

over many other technologies, such as satellite links, by being cost-efficient, robust, and easy to expand or repair on demand.

In short, the proposed hybrid WMN consists of a set of router nodes (RNs) organized in a mesh topology that make up the backbone of a network. The connection between the RNs is realized using radio waves; thus no cable connections are required. The RNs considered use low-power COTS router devices following the IEEE 802.11a/n standard [4]). A WMN client connected to the network can communicate with other clients of the network since RNs serve as access points and relay traffic.

Generally, WMNs consisting only of RNs do not scale well and offer limited throughput as the number of nodes in the network increases, both in the case of single-channel networks [5–7] as well as in the multi-channel case [8]. There are primarily two reasons for this: traffic increases with the total number of nodes in the network, and nodes need to relay more traffic when paths in the network become longer, further contributing to the total traffic increase. In order to alleviate the scaling problem of large WMNs consisting of hundreds or thousands of nodes covering a large region, some of the RNs can be resourced to act as gateways to other networks.

Typically, the gateway nodes (GNs) considered in common deployment scenarios [9, 10], provide Internet access through a wired connection mixing wireless multi-hop mesh networks with traditional wired LAN networks. This increases the capacity of the network, where most often traffic goes from RNs to gateways with gateways acting as sinks in the network. That is how the total capacity of the network is increased.

In the applications targeted in this paper, the use of wired LAN infrastructure by gateway nodes is not guaranteed as it may be physically impossible to deploy such infrastructure in the short period of time required during a rescue operation. Instead, inspired by previous work [11, 12], a hybrid WMN topology is proposed that utilizes gateways capable of using long-range wireless radio links for GN-GN communications. These gateway nodes can increase the total network capacity by acting as bridges between nodes that are far apart in multi-hop mesh networks reducing the amount of traffic that would otherwise be required to be relayed by a number of RNs. Additionally, they can provide a means for client connectivity to the main Command and Control Centers (C2).

Using a hybrid network design that relies heavily on wireless radio technologies may potentially suffer from performance problems caused by radio wave interference. One assumption made in the proposed hybrid WMN infrastructure is that gateways used in the targeted applications utilize an alternative wireless technology working with different frequencies than the basic RNs, so that their operation does not interfere with RN-RN links. Example

technologies that are currently available and could be used for that purpose include communication nodes based on the IEEE 802.16 WiMAX standard or mobile LTE cellular stations. Due to their nature, such as longer range, capacity, and power requirements, gateway nodes are more expensive than RNs, and their number should be minimized. Additionally, the GNs may require manual installation and deployment as opposed to their smaller, battery-powered counterparts, RNs, that can be autonomously delivered and set up by using UAVs.

Designing efficient and robust WMN network infrastructure is a challenging task with many aspects to consider. Included among them are two fundamental problems: 1) calculating the RN placement (RNP) to maximize overall network coverage while maintaining its connectivity; 2) and, finding a minimal number of gateway node placements (GNP) to ensure a sufficient level of Quality of Service (QoS) based on criteria such as maximum communication delay, relay load for each RN and capacity limits of gateway nodes.

In general, these problems have been shown to be in the class of NP-hard problems [13, 14], which rules out efficient polynomial-time algorithms that can provide optimal solutions. This being the case, the vast majority of approaches to these problems are heuristic-based. Due to the complexity of these problems individually, they are most often considered separately, with the exception of work in [10, 15] which uses an approach based on Integer Linear Programming, but requires extensive computational time and where node placement is limited to a set of predefined locations.

In the work presented in this paper, the initial focus is on solving an extended version of the RNP problem to create a fully connected WMN consisting of RNs (i.e. the WMN backbone). The hybrid multi-hop WMN infrastructure considered is depicted in Fig. 2. The RNP presented is an extended version of a common model considered previously in the literature [16, 17], but extended to guarantee line-of-sight constraints between RNs in order to increase the robustness of communication links in a real-world deployment.

A second focus is then placed on solving a combined RNP-GNP problem by leveraging an efficient heuristic algorithm proposed for the RNP problem. In the example network presented in Fig. 2, the RNP problem consists of calculating the optimal spatial locations for 21 RNs ($r_1, r_2, \dots, r_{20}$) restricted by a number of constraints. The RNP-GNP problem, on the other hand, considers finding a minimum number of GNs and their locations within an existing WMN backbone topology to guarantee certain QoS properties that directly influence the performance of the network. In the case considered in Fig. 2, three locations for GNs were found (at r_3 , r_{11} and r_{14}) creating three distinct sub-networks (clusters).

Fig. 2 Overview of the proposed hybrid Wireless Mesh Network topology

Typically, formal models proposed in the literature [13, 17] targeting the design of WMNs consisting solely of RNs, use the two network performance measures: *network connectivity* and *client coverage*. Network connectivity measures the degree of connectivity between router nodes and client coverage measures the number of client nodes connected to the WMN. Finding router node placement (RNP) in a specific geographical area for a WMN that optimizes both performance measures has been the focus of several recent research publications [16, 18, 19], although not in the context of UAV deployment.

When solving the RNP problem from the practical application perspective of deploying WMNs in emergency rescue scenarios, a special set of assumptions, different from those found in more theoretical research [16, 17] is required. In fact, this set of assumptions was derived through actual experimentation with UAV-deployed WMNs.

One strong assumption used in previous work [13, 17] is that user/client positions are known beforehand. Although in some environments typical client positions could be assumed or estimated (e.g. populated urban areas) due to urban structure, this cannot be assumed in emergency rescue scenarios where the environments are often unstructured, the network clients (rescuers or victims) are highly mobile, and where responders are usually required to perform an exhaustive search of the disaster area in question. This leads to an alternative network coverage performance measure that is expressed as the total area covered by the WMN.

Another assumption used in previous work [16, 17] relates to *network connectivity*, where obstacles in the environment are not dealt with at all. The signal strength of a mesh router depends not only on the distance to another router, but also on any physical structures, such

as obstacles, present in the space between them. Consequently, to fully utilize the range of a mesh router, obstacles in the environment have to be taken into account when calculating the RNP.

For this reason, two additional constraints are proposed and used when calculating a solution for an RNP problem: 1) enforcing line-of-sight requirement between two connected mesh routers, and 2) excluding obstacles from possible node placements. The latter constraint relates to the example mission scenario depicted in Fig. 1, where one wants to avoid delivery of mesh router nodes on collapsed building structures or densely vegetated areas with trees in order to increase the robustness of the network connections.

Given these additional constraints, the RNP problem becomes fundamentally more difficult, but its solution is radically more applicable in real-life scenarios. One major part of the solution to this problem is to develop an algorithm for RNP that is incremental in nature and appeals to intuitions derived from the motion planning domain where the optimization problem is combined with the use of the Rapidly Exploring Random Trees (RRT) [20] algorithm used in motion planning. This algorithm is generic in nature in that it can be viewed as searching through a search space, RNP graphs in this case, both incrementally and with a random exploration of that search space.

The most common formulation of the GNP problem used in previous work [14, 21, 22], assumes that the router node placement is given and the goal is to find a minimum number of gateways and their placement so that network QoS constraints are guaranteed. This is done by dividing the WMN backbone network, represented as a graph, into a set of disjoint clusters (sub-networks) covering the entire network. A single gateway, called a clusterhead, is then assigned to each cluster that will serve all RNs

assigned to that particular cluster. To maximize network performance, the location of clusterheads is chosen while considering a set of QoS constraints. Most often, three QoS constraints are used: 1) maximum communication delay (R_{QoS}); 2) maximum relay load for each RN (L_{QoS}); 3) gateway throughput (S_{QoS}). Assuming each router generates an equal amount of one unit of traffic and using a graph representation of the given WMN backbone network topology, the first constraint that represents the delay can be formulated as the maximum number of hops (R_{QoS}) information has to travel in between a router node and the gateway to which the RN is assigned to, as each re-transmission adds to the total delay. Thus, the R_{QoS} is an upper bound on the cluster radius. In the example presented in Fig. 2, three clusters are created with r_3 , r_{11} and r_{14} nodes chosen as clusterheads and the R_{QoS} being equal to 2.

Similarly, the second constraint (L_{QoS}) is placed to guarantee a throughput within each cluster as information sent to the particular gateway may need to be re-transmitted by intermediate RNs. Thus, the L_{QoS} is an upper bound on the number of router nodes that rely on re-transmission in

shortest paths that lead to a gateway from all RNs. In the example network depicted in Fig. 2, $L_{QoS} = 1$ in *Clusters 1-2*, and $L_{QoS} = 2$ in the case of *Cluster 3*, since the data from two RNs (i.e. r_{17} and r_{21}) has to be re-transmitted by node r_{20} .

Finally, the third constraint (S_{QoS}), considers the bandwidth capacity of gateway nodes since the total traffic within each cluster has to be potentially served by a gateway. The S_{QoS} is defined as an upper bound on the cluster size. In the example presented in Fig. 2, $S_{QoS} = 7$ in *Cluster 1*, $S_{QoS} = 6$ in *Cluster 2* and $S_{QoS} = 8$ in *Cluster 3*.

In this paper, two alternative algorithms are proposed for solving the combined RNP-GNP problem. These alternatives are depicted in Fig. 3. The first approach, *Sequential RNP-GNP*, uses a combination of the RRT-WMN algorithm and a popular and well-established graph clustering technique called Weighted Recursive Dominating Set (WeightedRecursiveDS) [22]. The RRT-WMN is used to efficiently solve an RNP problem for the entire target region of deployment. The resulting WMN backbone network topology is then used as an input

to the WeightedRecursiveDS graph clustering algorithm to solve the specified GNP problem. The RRT-WMN algorithm ensures that the network topology graph in the entire environment is fully connected and reaches the required network coverage, minimizing the number of router nodes used. The graph clustering technique selects a minimum number of gateways and their placement required in the network to satisfy the given QoS constraints.

The second approach, *WMNbyAreaDecomposition*, is a novel algorithm that uses a divide-and-conquer strategy to decompose the RNP-GNP problem into a set of single RNP problems solved concurrently. This is done by using a recursive algorithm that partitions the initial target deployment area into a set of equally sized disjoint sub-regions where smaller RNP problems can be solved efficiently with the RRT-WMN. A single gateway is assigned in each sub-region. The algorithm continues the decomposition with an increasing number of sub-regions until the QoS constraints are satisfied, thus finding the minimum number of gateways.

1.1 Contributions

This paper makes the following contributions:

- It provides a new definition of the RNP problem taking into account both technical constraints associated with WMNs and structural constraints commonly associated with areas affected by disasters.
- It provides an extended version of the RRT (Rapidly Exploring Random Trees) algorithm [20], called RRT-WMN, first reported in [23] for solving the extended RNP problem. The RRT-WMN algorithm uses the RRT algorithm as a basis for solving this problem. The RRT algorithm has been primarily used for motion planning applications. Unlike the standard version of the RRT, the RRT-WMN searches for the best possible graph configuration using a heuristic that maximizes optimization objectives.
- It provides a detailed experimental evaluation for solving the RNP problem, where the proposed algorithm is compared to two popular optimization techniques (CMA-ES [24] and PSO [25]), with the latter being used in previous work [13, 16, 19] for the RNP problem. The result of the experimentation shows that the new algorithm has far better performance in both time and coverage as the RNP problem scales to realistic scenarios. Two types of evaluation domains are used. The first type is a randomly generated set of environments with increasing complexity. The second type is a set of real environments based on data acquired from a GIS system for three cities in Sweden with increasing size and density.
- Two algorithms are proposed for solving the combined RNP-GNP problem. The first one is a simple sequential algorithm that leverages the efficiency of the RRT-WMN to find router node positions, which are then used as an input to a heuristic graph clustering algorithm. The second approach is a new algorithm that interleaves solving of the RNP with the GNP to find a minimal number of GNs by dividing the environment into a set of equally sized sub-regions until the given QoS constraints are satisfied.
- The two alternative algorithms are evaluated in the aforementioned actual domains based on real GIS data for three cities in Sweden. The results show that the second algorithm using the idea of area decomposition, outperforms a simple sequential algorithm with a solid margin of 73.7% to 92.5% less gateways on average required for the same QoS constraints.

1.2 Paper structure

Section 1 (this section) provides a thorough introduction to the problem area and a peek at some of the solutions that will be considered. Section 2 provides a detailed examination of related work relevant to the problems considered in the introduction. Section 3 considers the Router Node Placement problem by defining a generalization of the standard definition, new algorithms to solve the problem efficiently, and a comparative analysis of the proposed algorithms with several that exist in the literature. Section 4 considers the Gateway Node Placement problem and the combined problem of dealing with both Router and Gateway Node Placement simultaneously. Section 5 briefly describes development of UAVs customized to physically deliver routers and gateways based on guidance from the algorithms proposed here. Section 6 concludes the paper with a summary of work considered in the paper.

2 Related work

There are a wide range of research topics and approaches related to the deployment, maintenance and use of WMNs that have been reported in the literature in the past decades. Some of this work includes development of communication protocols with a focus on issues related to routing [26–29] and channel assignment [30, 31]; radio techniques and technologies including research in smart antennas, multiple-input and multiple-output (MIMO) [32, 33]; issues related to network security [34–36]; and last but not least, the fundamental problems related to network topology and infrastructure [13, 17, 22, 37]. The latter is the main focus of the work presented in this paper.

Problems dealing with the RNP and GNP are both NP-hard. Consequently, the majority of approaches presented in the literature target use of heuristic algorithms that can

solve problem instances with realistically large numbers of nodes in the resulting networks and that can also be generated in a reasonable time. In the following subsections, relevant work related to the topics in this paper is presented and comparatively summarized. The related work is presented in three parts. First, existing approaches that deal with the RNP and GNP problems separately are considered. This is then followed by a discussion about the limited number of proposals whose aim is to solve the combined RNP-GNP problem.

2.1 Router node placement

During the past decades, the problem of calculating router node positions in WMNs has been a major focus of many studies. As mentioned, since the RNP is an NP-hard problem [13] and generally modeled as a continuous or discrete multi-objective optimization problem, the main body of work is related to finding efficient heuristic or meta-heuristic methods for use in practical applications.

In [38], the authors propose a local search algorithm for solving RNP in networks with a two-tier architecture [39]. A four-step deployment scheme is proposed that includes division of the environment into grid cells, choosing convenient locations for potential RN placements, measurement of signal strength to estimate client node distributions, and finally calculating optimal placements of RNs using a local search method that optimizes the connectivity and coverage metrics. Both the connectivity and coverage metrics are represented as probabilities and require actual physical measurements of the signal strength to be calculated. In the application scenarios considered in this paper, such measurements are not feasible due to time constraints. Most importantly, due to the high mobility of rescue teams, the measured probability distributions would not give accurate representations. Consequently, in the approach pursued in this paper, as a metric of coverage, physical area coverage is combined with the calculation of positions of RNs that guarantee line-of-sight between RNs so that radio signal utilization is maximized.

Several meta-heuristic evolutionary algorithms have been proposed for solving the RNP problem, where the deployment environments targeted are discretized. Those approaches include the use of Genetic Algorithms (GA) [17], Simulated Annealing (SA) [40] and Tabu Search [41]. In these cases, a hierarchical approach to optimization is used to prioritize within the defined bi-objective function. The network connectivity, defined as the size of the largest component in the topology graph, is considered as a primary objective while user coverage is secondary. User positions are assumed to be given, and several models for generating client positions in experimental evaluations are used that include normal, uniform, exponential, and Weibull distributions. Extensive

studies have been conducted to compare and evaluate those approaches for problem domains with different sizes and parameters [42, 43], including the use of a Friedman test [44] and evaluations using the ns-3 simulation framework [45]. The results show the strengths and weaknesses of these approaches relative to factors such as communication ranges and the models used for generating client positions.

Another group of meta-heuristic algorithms has been proposed that uses a different set of assumptions. One of the main differences is that rather than using a hierarchical optimization approach, both objectives are optimized at the same time. In this case, the bi-objective function is a normalized sum of both objectives, i.e. network connectivity and client coverage. In a manner similar to the other meta-heuristic algorithms discussed above, these approaches also assume knowledge of client positions beforehand. A Particle Swarm Optimization (PSO) [25] algorithm applied to the RNP problem is proposed in [13]. In this case, as opposed to the previously discussed evolutionary algorithms, optimization in continuous state space is used. That is, rather than representing the deployment area as a grid of finite size, the positions of the router nodes are computed as floating-point values. This approach gives more flexibility as to where the router nodes can be positioned, and it is an important feature when considering practical deployment. The authors also formulate a model for dynamic network scenarios where both router nodes and clients can change their locations. The evaluation includes both static and dynamic cases. Several extensions have been proposed based on the PSO approach. In [18], the authors define a mobility scheme based on a social community structure to dynamically adjust the router node positions in the WMN. In [19], an SA approach with momentum terms was proposed to additionally deal with the service priority for client access control, although the deployment environment considered there is discretized.

Two very strong assumptions are made in the meta-heuristic approaches considered that limit their applicability to practical applications. This is ameliorated in this paper by proposing a new extended definition of the RNP problem with additional constraints.

One of the strong assumptions used by others is that of knowing client positions beforehand. The network clients (rescuers or victims) in rescue scenarios are highly mobile, and the responders are usually required to perform an exhaustive search of disaster areas in question. Instead of assuming knowledge of client positions in order to maximize coverage of the WMN network, the approach considered in this paper uses an alternative network coverage performance measure that is expressed as the total area covered by the WMN.

The second strong assumption relates to network connectivity metrics where obstacles in environments are not considered at all. The signal strength of an RN depends not only on the distance to another router, but also on any physical structures such as obstacles, present in the space between them. Therefore, to fully utilize the range of a mesh router, obstacles in the environment have to be taken into account when calculating the RNP and they are used in the approach considered in this paper.

Additionally, meta-heuristic methods that use a grid representation of the environment are not directly well suited for solving the extended RNP problem when obstacle constraints are taken into account. Simple grid discretization strategies applied to existing or acquired real-world models can easily limit the number of possible solutions. This results in finding sub-optimal solutions or in not finding any solutions at all. Consequently, the experimental evaluations used for comparison of the performance of the RRT-WMN algorithm proposed in this paper are based on two optimization techniques capable of solving problems in continuous domains: the PSO [16] and CMA-ES [24].

2.2 Gateway node placement

The GNP problem has also been proven to be in the class of NP-hard problems [14]. Consequently, the main focus in previous work has been placed on providing efficient heuristic-based algorithms to find near-optimal solutions to the problem. Typically, the GNP problem is defined in terms of an Integer Linear Program (ILP) defined to find a minimum number of gateway nodes for a given WNM backbone network that satisfies a given set of QoS constraints. Three QoS constraints are commonly considered that influence the network performance. These include the communication delay, relay load, and gateway throughput.

Several heuristic-based algorithms for solving the GNP that use a single objective of minimizing the number of gateway nodes given the three QoS constraints mentioned, have been proposed in [14, 21, 22, 37]. Other work has considered a second objective of minimizing the number of hops between an RN and its assigned gateway [46, 47]. One of the algorithms proposed in this paper for solving the combined RNP-GNP problem utilizes the heuristic graph clustering technique proposed in [22].

2.3 Combined router and gateway node placement

Research that focuses on solving the combined RNP-GNP problem fully, or at least to some degree, is very limited. In [16], a PSO approach was applied to the RNP problem that includes a set of GNs with their positions fixed beforehand. The experimental evaluation was limited to a case with four static gateways. Work presented in [48] proposed a heuristic algorithm for solving the RNP

problem that assumes one gateway is placed in the center of the calculated network topology. Gateway capacity and frequency channel assignment were taken into account when calculating the solution. The candidate RN positions considered during the solution search were generated using hexagonal tessellation over the circle centered around the gateway node.

In [10, 15], a WMN Topology Planning–Maximum Coverage (WMNTP-MC) model is proposed that targets solving an ILP to find solutions to the combined RNP and GNP problem. The ILP is solved using a CPLEX branch-and-bound technique. There are two main limitations to the WMNTP-MC approach which makes it ill-suited to the applications discussed in this paper. First is the required calculation time and lack of scalability to large problem instances. Since the approach relies on solving an ILP program directly by an LP solver, the computation times vary, and even for small problem instances take a considerable amount of time. In the experimental evaluation of problems that involved placing between 5–80 RNs and 1–16 GNs, the reported computation times varied between 96s and 9+ days, with an average time of 77.9h. In the context of time-critical applications, such as emergency rescue, algorithms with long execution times are not applicable, and more efficient approaches that can handle larger problem instances are required.

The second limitation is that in the WMNTP-MC approach, the RNP-GNP problem is modeled as a discrete combinatorial problem, where candidate locations for both RN and GN need to be predefined. This means that continuous environment models typically available in GIS systems cannot be handled directly, and an additional adequate selection or discretization process needs to be provided. Similar limitations exist in earlier work presented in [49]. In this case, the proposed ILP model focused on minimizing WMN deployment costs while dealing with problems of channel assignment. Experimental evaluations were presented with up to 105 nodes in total, including router and gateway nodes. In comparison, the two approaches proposed in this paper, for solving the combined RNP-GNP problems, utilize continuous models directly, take into account obstacle constraints to maximize the utilization of an RN's radio signal, and efficiently provide solutions to large-scale problems involving hundreds of nodes.

3 Router node placement

3.1 Problem definition

In this section, a mathematical model of the RNP problem is provided that is specified as a constrained optimization problem. A WMN is composed of a set of mesh router nodes (RNs) and mesh clients. RNs serve as access points for mesh clients (rescuers or victims in our domain)

and are interconnected with point-to-point wireless links creating a backbone for the network.

In Fig. 4, an example of a deployed WMN is shown consisting of five mesh routers r_i , with their respective radio coverage Υ (green circles), and ten mesh clients c_j (orange), deployed in the area P with six obstacles O (blue polygons).

For each WMN generated, a network topology graph, $G = (V, E)$, can be constructed and used for analysis. In the literature for RNPs [13, 16], the set of vertices (V) in a graph consists of mesh clients and mesh routers. The edges (E) represent connectivity, where: 1) for two router nodes to be connected they have to be in their communication range, and 2) the client/router edge is added when the client is in the communication range of a router. In the approach considered in this paper, the definition of edge connectivity is extended by introducing additional constraints that relate to the handling of obstacles and line-of-sight requirements.

Formally, a WMN for the RNP problem (Fig. 4) [17], is defined as a set of interconnected devices in a universe $U = R \cup C$, where $R = \{r_1, r_2, \dots, r_n\}$ is a set of n mesh router nodes and $C = \{c_1, c_2, \dots, c_m\}$ is a set of m mesh clients. Each r_i represents the i th mesh router and consists of a tuple $\langle r_i^X, \gamma_i, \Upsilon_i \rangle$ where $r_i^X = (x_i, y_i) \in \mathbb{R}^2$, is the position of the router node, γ_i is its nominal communication range, and Υ_i is the circle representing its radio coverage centered at the r_i^X position with radius γ_i for $i \in \{1, 2, \dots, n\}$. C is the client set, where each c_j for $j \in \{1, 2, \dots, m\}$, represents the j th client. In the problem formulation, it is assumed that the client positions are unknown, so $C = \emptyset$. Consequently, the network topology graph $G = (R, E)$ will consist of vertices from R , with a graph edge connectivity constraint on members of E defined as:

$$e(r_i, r_j) \in E(G) \iff \left(\overline{r_i^X r_j^X} \cap O = \emptyset \right) \wedge \left(\|r_i^X - r_j^X\| < \min(\gamma_i, \gamma_j) \right) \quad (1)$$

This states that for any edge in $E(G)$, between a pair of router nodes r_i and r_j in R , the line segment between their positions, $\overline{r_i^X r_j^X}$, is not in collision with any obstacle, and the distance between them is smaller than the minimum of their radio coverage radius (e.g. r_1 and r_2 in Fig. 4). The latter connectivity constraint used is different from previous work where two nodes were considered to be connected when their coverage circles overlap, $\Upsilon_i \cap \Upsilon_j \neq \emptyset$.

The formulation used in this paper is motivated by pragmatic constraints, since most commercially available devices have the same radio range for both inter-router mesh connections as well as clients. Therefore, a pair of connected routers will always have an overlap in their respective coverage areas, but additional constraints may be needed for the connectivity to actually work in practice.

The deployment area for an ad hoc network is defined as a set P specified by a polygon. Within the area, an obstacle set is the union of all obstacle regions specified by polygons, for m obstacles $O = O_1 \cup O_2 \cup \dots \cup O_m$.

Based on the definitions introduced so far, the extended RNP problem considered in this paper can be specified as the following non-convex constrained optimization problem. Find n mesh router positions r_i^X where $i \in \{1, 2, \dots, n\}$, that maximize a cost function $f(\mathcal{X})$ given a target deployment set P , obstacle set O , and a network topology graph $G = (R, E)$. The objective function is defined as the proportion between the geographical area covered by the WMN and the deployment set P without obstacles:

$$f(\mathcal{X}) = \psi(G, P, O) = \frac{\text{Area} \left(\left(\left(\bigcup_{i=1}^n \Upsilon_i \right) \cap P \right) \setminus O \right)}{\text{Area}(P \setminus O)}, \quad (2)$$

where $f: R^n \mapsto \mathbb{R}$ (each $x \in \mathcal{X}$ is a tuple $\langle r_i^X, r_{i+1}^X, \dots, r_n^X \rangle$). Intuitively, when $\psi(G, P, O) = 1$, the WMN defined by the graph G fully covers the entire deployment area excluding obstacles, and at $\psi(G, P, O) = 0$, there is no coverage of

the area. Using *network coverage* $\psi(G, P, O)$, as an objective function rather than *client coverage* used in other approaches, allows for searching for solutions that are not dependent on knowledge of client positions. Details of the area coverage calculations are presented in Section 3.2.

Another important measure of WMN performance is *network connectivity* which quantifies how well the routers are interconnected. Since the connections between router nodes depend on their radio ranges and line-of-sight constraints, a network topology graph G for a particular set of RN positions may not be fully connected. This means it may consist of several subgraph components as shown in the example WMN in Fig. 4. There the graph G consists of 3 subgraphs (G_1, G_2, G_3) in the WMN backbone case, excluding client connections. The size of the greatest subgraph component in $G = G^1 \cup G^2 \cup \dots \cup G^h$ is defined as:

$$\phi(G) = \max_{i \in \{1, 2, \dots, h\}} \{|G^i|\}. \quad (3)$$

Typically, in other work [13, 17], the *network connectivity* (also called *size of the greatest component*) is considered as part of the objective function. In this case, there is always a trade-off between finding solutions that maximize both objectives. Such problems are often solved using the bi-objective optimization technique, where each objective is multiplied with a weight parameter emphasizing the importance of each separate objective.

Generally, *network connectivity* is considered more important than *client or network coverage*. Consequently, in this formulation it is considered as a very strong requirement that the RNP solution guarantees a fully connected network. For that reason, network connectivity is treated as a constraint rather than an objective to be maximized.

The general optimization problem formulation for the extended RNP problem, which includes the objective function and all the constraints considered in this approach, is defined as:

$$\max_{\mathcal{X}} f(\mathcal{X}) \quad (4a)$$

$$\text{s.t. } r_i^X \in P \setminus O \quad \forall i \in \{1, 2, \dots, n\}, \quad (4b)$$

$$e(r_i, r_j) \in E(G) \iff$$

$$\overline{r_i^X r_j^X} \cap O = \emptyset \wedge (\|r_i^X - r_j^X\| < \min(\gamma^i, \gamma^j)) \quad \forall i \neq j; i, j \in \{1, 2, \dots, n\}, \quad (4c)$$

$$\phi(G) = n. \quad (4d)$$

The goal of the optimization problem is to maximize the geographical area which is covered by the WMN (Eq. 4a) and the constraints enforce three important features of

solutions. First, all router positions should be within the deployment area P with exclusion of obstacle regions (Eq. 4b). Second, connections between any two routers that are considered as valid have a direct line-of-sight (Eq. 4c). Finally, the topology graph G is fully connected where it consists of a single graph (Eq. 4d).

In practice, solving this non-convex, non-smooth, high-dimensional constrained optimization problem directly is infeasible. An alternative to solving such a problem is to use penalty function methods in which the constraints are replaced with a penalty function that is a cumulative measure of constraint violations. The penalty function is added to the objective function, thus discouraging the search algorithm from visiting infeasible parts of the search space.

Penalty methods with binary constraints still require parameter tuning and there are no guarantees for convergence to the global optimum. In an attempt to alleviate this problem, a penalty function is used that also includes a distance measure indicating how infeasible a particular solution is. The objective function, including penalties for different constraint violations, is calculated as follows:

$$\max_{\mathcal{X}} (f(\mathcal{X}) - \pi(\mathcal{X})); \quad \pi(\mathcal{X}) = \sum_{p=1}^4 (\pi_c \cdot \pi_p(\mathcal{X}))^2, \quad (5)$$

where π_c is a penalty coefficient chosen empirically. Four π_p penalty functions are used in the RNP problem formulation, where in the case of no constraint violations the value of $\pi(\mathcal{X}) = 0$.

$$\pi_1(\mathcal{X}) = \sum_{i=1}^n \begin{cases} r_i^X \in P \rightarrow 0, \\ r_i^X \notin P \rightarrow d(P, r_i^X), \end{cases} \quad (6)$$

$$\pi_2(\mathcal{X}) = \sum_{i=1}^n \begin{cases} r_i^X \notin O \rightarrow 0, \\ r_i^X \in O \rightarrow \exists k (r_i^X \in O^k) \rightarrow d(O^k, r_i^X), \end{cases} \quad (7)$$

$$\pi_3(\mathcal{X}) = \sum_{e(r_i, r_j) \in E(G)} \begin{cases} \overline{r_i^X r_j^X} \cap O = \emptyset \rightarrow 0, \\ \overline{r_i^X r_j^X} \cap O \neq \emptyset \rightarrow 1, \end{cases} \quad (8)$$

$$\pi_4(\mathcal{X}) = \sum_{\substack{j>1 \\ r_i \in G^1 \\ r_j \in G^j}}^h \min \|r_i^X - r_j^X\| - \min(\gamma_i, \gamma_j). \quad (9)$$

Penalties π_1 and π_2 correspond to the first constraint (Eq. 4b) and if violated they are calculated as a distance measure to the closest border of the deployment area or obstacle, respectively. Line-of-sight constraints (Eq. 4d) are imposed by penalty π_3 . Full graph connectivity constraints are enforced by penalty π_4 which in the case of a violation, where graph G consists of several subgraphs, it will be equal to the sum of the minimum distances between the largest subgraph component G^1 in the graph G and the remaining subgraphs. This penalty will decrease

as the number of subgraphs decreases, or the remaining smaller subgraphs are moved closer to the largest subgraph component G^1 .

Note that although solving the extended RNP problem using penalty functions bears some similarities to bi-objective optimization, where constraints are a part of the objective function with much higher negative values, in this case, a feasible solution is one that meets all the constraints. Thus the penalty in the objective function is $\pi(\mathcal{X}) = 0$.

3.2 Network coverage calculation

Network coverage calculations are an essential part of the RNP algorithm described in Section 3.3. These calculations have to be done efficiently. Thus an approach based on grid representations allowing for fast computation of covering areas is used. Two types of network coverage calculations are used in the algorithm. The first, $\psi(G, P, O)$, assesses the total geographical area covered by a WMN over a given deployment set P , excluding obstacles O . It is used as the objective function (Eq. 2) when applying optimization algorithms. The second variant, $\mu(G, P, O, r_{new})$, assesses an area coverage improvement due to a particular router node r_{new} , when selecting a new router node position in the process of finding a final network topology graph incrementally.

Both area calculations are approximated using grid representations encoded as three two-dimensional matrices. This allows us to leverage very efficient image processing algorithms to calculate network coverage. An example for a sample environment is depicted in Fig. 5.

3.3 RRT-WMN algorithm

In this subsection, a description of the proposed RRT-WMN algorithm is presented. The algorithm incorporates an extension to the Rapidly Exploring Random Trees

(RRT) [20] motion planning technique that is capable of finding efficient solutions to the motion planning problem in multi-dimensional state spaces that take into account state constraints. The original version of the RRT algorithm iteratively builds a tree rooted in an initial state by extending it towards a randomly selected state. This expansion procedure is biased towards covering unexplored parts of the state space, and that is the key feature exploited in the RRT-WMN algorithm when applied to the RNP domain. The RRT and its variants are predominantly applied to finding motion plans that connect an initial state to a goal state.

Although the standard RRT algorithm can be applied to finding solutions to the extended RNP problems, it has a significant limitation. There is no notion of cost or objective in the standard RRT algorithm variants that could be used to optimize towards a given objective. The extended RNP problem is an optimization problem where the goal is to maximize the network coverage. As such, the standard RRT would provide a single solution generated randomly without considering the main objective formulated in the RNP problem. Therefore, the objective of maximizing the network coverage would not be realized. Additionally, the standard RRT use would be limited to RNP problems where all router nodes are homogeneous.

The RRT-WMN algorithm uses the same state exploration principle for calculating router node positions as the standard RRT. However, instead of generating a single graph, the algorithm creates multiple graphs in parallel, each of which is a feasible RNP solution. A recursive process is used to compute these graphs where, at each step, a number of potential new router positions are considered for addition to one of the existing partial graphs. All router position candidates are ranked according to an area coverage improvement score, i.e. $\mu(G, P, O, r_{new})$.

Fig. 5 Area coverage calculation for a sample environment with 0.5 m/grid cell resolution. Overview of the router node placement problem with the new candidate node r_{new} highlighted in yellow (problem environment), E_M grid representation of the environment, E_G grid representing the topology graph G , and final area coverage grid representation E_T

The pseudo-code for the algorithm is presented in Algorithm 1 and Algorithm 2. The input to the RRT-WMN is a graph G (current partial solution) and a set R (the list of router nodes as tuples), (see Section 3.1). The parameters used during the calculations include polygons specifying the deployment area P and obstacles O , a limit for the number of candidates considered at each iteration N_B , a *timeout* constant and a minimum area coverage value ψ_{min} . The last parameter is used for finding a minimum set of router nodes that yields at least ψ_{min} network coverage.

Algorithm 1: RRT-WMN

```

Input:  $G, R, timeout$ 
Parameters :  $P, O, N_B, \psi_{min}$ 
1 if  $G = \emptyset$  then
2    $r_{init} \leftarrow \text{removeFirst}(R)$ 
3   if  $r_{init}^X = \emptyset$  then
4      $r_{init}^X \leftarrow \text{randomState}(P, O)$  ▷ select random state s.t.:  $r_{init}^X \in P \setminus O$ 
5      $G \leftarrow G \cup r_{init}$ 
6      $G_{best} \leftarrow \emptyset$ 
7 if  $timeout$  then return  $G_{best}$ 
8 if  $|R| > 0 \wedge \psi(G) \leq \psi_{min}$  then
9    $candidates \leftarrow \text{getConstrainedStates}(G, R, N_B)$  ▷ rejection sampling
10  sort  $candidates$  according to score
11  for  $(r_{new}, r_{score})$  in  $candidates$  do
12     $G \leftarrow G \cup r_{new}$ 
13     $\text{remove}(R, r_{new})$ 
14     $G_{best} \leftarrow \text{RRT-WMN}(G, R)$  ▷ recursive call
15 else
16   if  $\psi(G)/\phi(G) > \psi(G_{best})/\phi(G_{best})$  then
17      $G_{best} \leftarrow G$  ▷ the current graph is a better solution
18 return  $G_{best}$ 

```

An initialization step is performed (in Lines 1-6) that either assigns a predefined router node position, already given as an input to the algorithm, where $r_1^X \neq \emptyset$ in $r_1 \in R$, or it selects the first position by random state sampling, subject to the constraints in Eq. 4b. A *timeout* termination condition is checked in Line 6, and if true, the current best solution is returned. In the main loop of the algorithm, a condition for a feasible solution is evaluated in Line 8. It checks if there are any more nodes in R to be added to the solution graph, or if the current solution has reached a specified minimum network coverage (ψ_{min}). If any of these two conditions are true, a comparison between the current solution graph and the best graph found so far is performed (Lines 16-17). Otherwise, a set of N_B candidate nodes are generated by performing rejection sampling according to Algorithm 2.

Algorithm 2: GETCONSTRAINEDSTATES

```

Input:  $G, R, N_B$  - number of candidates,
Output:  $candidates = \{(r_{cand}, r_{score})\}$ 
Parameters :  $P, O$ 
1  $candidates \leftarrow \{\emptyset, \emptyset\}$ 
2 while  $|candidates| \leq N_B$  do
3    $x_{rand} \leftarrow \text{randomState}(P, O)$  ▷ sampling for positions to expand towards
4   if  $\text{inRange}(x_{rand}, G)$  then continue ▷ already in WMN range
5    $r_{near} \leftarrow \text{NearestVertex}(x_{rand}, G)$ 
6    $r_{new} \leftarrow \text{random}(R)$ 
7    $d = \min(\gamma_{near}, \gamma_{new})$  ▷ find minimum distance to expand
8    $x_{new} \leftarrow \text{NewState}(x_{rand}, r_{near}^X, d)$ 
9   if  $\neg \text{CollisionFree}(P, O, x_{new}, r_{near}^X)$  then continue
10   $r_{new}^X \leftarrow x_{new}$ 
11   $r_{score} \leftarrow \mu(G, P, O, r_{new})$  ▷ calculate area coverage improvement
12   $candidates \leftarrow candidates \cup \{(r_{new}, r_{score})\}$ 
13 return  $candidates$ 

```

The candidate nodes are then sorted according to their score and the one with the highest score is added first to the solution graph G and removed from the set R . The

updated solution graph G and the set of unassigned router nodes R are then used in the recursive call to the algorithm (Line 14).

The rejection sampling algorithm (Algorithm 2) generates a random state x_{rand} , in Line 3, subject to the constraints specified in Eq. 4b. In case the selected random state is already in range of the current solution graph G , a new random state is generated. Next, the closest router r_{near} in the graph to the random sample is selected (Line 5).

In order to enforce the full network connectivity constraint specified in Eq. 4d, a distance d used for graph extension is calculated. Its value is based on the minimum radio coverage radiuses of r_{near} and one of the unassigned router nodes in R that is selected randomly (Lines 6-7). A new state is then calculated (x_{new}) as a point on a line between the positions of x_{new} and r_{near}^X located at the distance d from r_{near}^X (Line 8). A line-of-sight check, according to the edge connectivity constraint specified in (Eq. 4c), between the points is then performed (Line 9). If successful, the new state is used as a candidate with its score calculated using the area coverage improvement $\mu(G, P, O, r_{new})$. The procedure is repeated until N_B candidates are created. Figure 6 presents an example of two successfully generated candidates (r_{new1} and r_{new2}) and some rejected state samples, x_{rand3} , which is already in range of the WMN (rejected in Line 4), and x_{rand5} , where the potential new extended state would not have a line-of-sight to its nearest node r_{near2} (rejected in Line 9).

3.3.1 Algorithm enhancement

The main objective of the RRT-WMN algorithm is to maximize the deployed network coverage given a set of router nodes to be distributed. Thus, by default, the distance d at which a new candidate node is generated in the

Fig. 6 Generation of candidate router positions in the RRT-WMN algorithm

rejection sampling step (Algorithm 2), is calculated as the maximum allowed value which ensures two router nodes are still connected.

The difficulty of finding a solution for the RNP depends on the structure of the environment, the structure and density of the obstacles, and the ranges of router nodes selected for the deployment of the network in set R . For practical reasons of in-air deployment, it is assumed that the network should be constructed from identical routers. This implies that the distance d between two or more connected nodes is the same. In environments where the value of d is smaller than typical distances between obstacles, finding new candidate node positions by rejection sampling is relatively straightforward because there are many pairs of positions that meet the constraints.

Such problem instances are considered to be *sparse* relative to the value of d . Finding RN positions using this strategy can be problematic in *dense* cases where the obstacles can limit valid router placements considerably and 1) a solution may not even exist given the router node range parameters or 2) the random sampling is considerably more difficult, thus convergence to any solution is much slower. This is similar to the problem of finding a valid path through a narrow passage in motion planning in robotics. Typically, an environment will have both types of sparse and dense characteristics, such as sparse regions associated with city suburbs and denser regions closer to the city center. In this case, the difficulty of finding valid router node placement will vary.

Due to the variation of density in the environments studied, an extension to the basic version of the RRT-WMN algorithm is proposed, where the distance d (Line 7, Algorithm 2), at which a candidate node x_{new} is generated, is additionally adjusted by a factor λ depending on how difficult it is for the algorithm to generate new

candidate nodes. This may result in a higher number of router nodes used in the solution, but it will also improve both convergence and computation time of the algorithm.

A simple strategy can be used where a timeout is added for the rejection sampling step. Each time a new candidate generation fails in the allocated time, the next value of distance d is reduced by a factor such as 10%.

Whenever a new candidate is generated within the specified time, the λ factor is increased until it reaches a value of 1. The λ factor is specified as:

$$\lambda_i = \begin{cases} (\lambda_{i-1} + \delta), & \text{if } (T_i - T_{i-1}) > 0, \\ (\lambda_{i-1} - \delta), & \text{if } (T_i - T_{i-1}) = 0, \end{cases} \quad (10)$$

where i is the iteration number of the rejection sampling algorithm, T_i denotes the number of timeouts which occur until step i , and δ is a fixed constant defining a step size for reduction or increase of λ . The value of λ is always limited to the range $\lambda \in [\lambda_{min}, 1.0]$. Initial value of $\lambda_0 = 1.0$.

For example, with $\delta = 0.1$ and $\lambda_{min} = 0.5$ each timeout occurrence will effectively reduce the distance d by 10% compared to a nominal value calculated in Line 7 of Algorithm 2, and decrease down to 50% of the original value. An illustrative example is depicted in Fig. 7.

3.3.2 Convergence analysis

In this section, an informal proof of global convergence for the RRT-WMN algorithm is provided. A thorough proof for a global convergence for a conceptual random search algorithm has been presented in [50]. Although the RRT-WMN is not identical to the conceptual algorithm, it does include its key features and structure. These features include both algorithms starting from an initial random solution, generating random new solutions, and selecting the best solution at each execution step. The RRT-WMN

Fig. 7 Simple illustration of adaptive step size enhancement. In step 1 the distance d_1 is selected based solely on r_1 and r_{new} router ranges which makes it impossible to find a valid new candidate position. After a timeout, the step size parameter is decreased to $\lambda = 0.9$, thus $d_2 = 0.9 \cdot d_1$ allowing to find a valid candidate node position r_{new}

algorithm additionally includes a mechanism of recursion, use of heuristics, and constraints. Because both algorithms follow the same structure, the same framework for proving a global convergence used in [50] can be applied when analyzing the convergence properties of the RRT-WMN algorithm.

In short, the global convergence proof proposed in [50] considers the following problem. Given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $S \subset \mathbb{R}^n$. The goal is to find a point $x \in S$ which minimizes f on S or at least which yields an acceptable approximation of the infimum of f on S . Global convergence for an algorithm is achieved if the following two properties, (H1) and (H2), are satisfied:

$$(H1) \quad f(D(x, \xi)) \leq f(x) \quad \text{and if} \quad \xi \in S, f(D(x, \xi)) \leq f(\xi), \quad (11)$$

where ξ is the newly generated solution at any step and x is the current best solution. S is the space of acceptable solutions. Function $D(x, \xi)$ selects x or ξ depending on which solution is better. Essentially when an algorithm fulfills the (H1) property, the algorithm generates solutions that are improving.

$$(H2) \quad \forall A \subset S : v(A) > 0, \quad \text{we have that} \quad \prod_{k=0}^{\infty} (1 - \mu_k(A)) = 0, \quad (12)$$

where μ_k is the sequence of random numbers generated by the algorithm. A is a subset of the solutions S . $v(A)$ is the volume of A , and $v(A) > 0$ means A is not empty. Essentially the (H2) property states that for an infinite amount of draws, the probability of missing the elements of set A must be equal to zero. In practice, when an algorithm fulfills the (H2) property, the algorithm is guaranteed to find globally optimal solutions after performing an infinite number of draws.

The RRT-WMN is searching for an RNP solution for $n = |R|$ router nodes by recursively generating a number of WMN topology graphs G . At each recursion level, the RRT-WMN solves a problem of a single router node placement by using a random search approach (Line 9, Algorithm 1) and recursively calls the algorithm to place the remaining routers. The placement of one router node is done by the rejection sampling algorithm (Algorithm 2) that generates a N_B number of router configurations, one of which is chosen by the RRT-WMN algorithm according to heuristics. All configurations satisfy the RNP problem constraints and include the router's position and what router node is selected from the initial set of routers R .

Fulfilling the (H1) property is guaranteed by the RRT-WMN algorithm in Lines 16-17 of Algorithm 1, where a newly generated complete solution G (i.e. containing all router nodes from R or exceeding a network coverage threshold ψ_{min}) is compared to the existing best-known solution G_{best} and the best of both solutions is kept.

Let $RRT-WMN^i$ denote the i -th level of the recursive process in the RRT-WMN algorithm, so that the $RRT-WMN^i$ calls $RRT-WMN^{i+1}$. The recursion stops when $i = n$. Thus, $RRT-WMN^n$ is the last recursive step of the algorithm at which all router node configurations are generated, and the construction of a new solution is completed. The $RRT-WMN^{n+1}$ step will therefore not be executed. Note that in (Line 8, Algorithm 1), there is an optional condition to check if the current solution has reached the specified minimum network coverage.

At level n , all the router nodes have been placed and $RRT-WMN^n$ only checks if the current solution is better than the absolute best-achieved solution so far (G_{best}).

The proof that the RRT-WMN algorithm satisfies the (H2) property is done using induction. We show that for all i the $RRT-WMN^i$ satisfies the (H2) property. In the base case, at level $i = n - 1$, the algorithm calculates the placement of the last (n -th) router node using a random search. Assuming the number of candidate locations $N_B \rightarrow \infty$, the full space of feasible solutions will be explored (Line 9, Algorithm 1). Consequently, any subset A is explored. Thus, the $RRT-WMN^{n-1}$ satisfies the (H2) property.

In the general case, at the i -th level of the algorithm, the G_i graph represents the set of router nodes placed at previous levels. The algorithm generates random r and executes a recursive call $RRT-WMN^{i+1}(G_i \cup r)$ to calculate configurations of the remaining router nodes. The induction hypothesis is that $RRT-WMN^{i+1}(G_i \cup r)$ satisfies the (H2) property.

To prove the hypothesis consider a subset of solutions $A \subset S$, and a solution $a = (r_0, r_1, \dots, r_{i-1}, r_i, \dots, r_n)$. The probability that the router configuration r_i of a is drawn is equal to 1.0 when the RRT-WMN (Line 9, Algorithm 1) is executed using an infinite N_B number of draws. Executing the RRT-WMN at the next level $i+1$ generates the remaining router configurations. Since the $RRT-WMN^{i+1}(G_i \cup r_i)$ satisfies the (H2) property, the probability that the other router configurations $(r_{i+1}, r_{i+2}, \dots, r_n)$ of a are drawn is equal to 1.0. Thus, the probability that $RRT-WMN^i$ draws all existing configurations in a is equal to 1.0, proving that $RRT-WMN^i$ satisfies the (H2) property.

By induction, for all i , the $RRT-WMN^i$ satisfies the (H2) property. Since $RRT-WMN^0$ is the RRT-WMN full algorithm, the RRT-WMN satisfies the (H2) property. In conclusion, the RRT-WMN algorithm satisfies both (H1) and (H2) properties. Therefore, it will converge toward a global optimum.

3.4 Experimental evaluation for solving RNP problems

In this section, results of a comparative experimental evaluation are provided for the extended RNP problem. First, the performance of the RRT-WMN algorithm is compared with two optimization techniques, Particle Swarm

Optimization (PSO) [25] and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [24] using randomly generated environments. The criteria used for comparison between the algorithms are based on the computational time and quality of the solutions measured in terms of the coverage of the WMN. The second part of the evaluation focuses on the performance evaluation of the RRT-WMN algorithm when applied to actual environment models based on real GIS data.

3.4.1 Comparative evaluation using randomly generated problems

The PSO algorithm variant used for comparison, proposed in [16], includes the use of a constriction coefficient and local search. It has been successfully applied to a different RNP problem specification. The CMA-ES algorithm is an evolutionary optimization technique that has grown in popularity in recent years due to its success in solving many problems where the objective function is non-convex, multi-modal, non-smooth, discontinuous, noisy, or ill-conditioned. All the evaluated algorithms were implemented in the Python programming language, and the experiments were run on a computer equipped with an Intel Core i7 (I7-6820HQ) processor.

Three cases of different complexity were designed for the evaluation. The first two are based on setups previously used in the evaluation of other approaches [13, 16, 51], and include a relatively small limit on the number of router nodes used. These cases are used for comparison between the two optimization techniques, PSO and CMA-ES, and the RRT-WMN algorithm. The third case has a significantly larger complexity and is used to study the robust scalability of the RRT-WMN algorithm in comparison to the other approaches.

- Case 1: 16 mesh router nodes with a circular radio coverage range with a radius of 6m are to be deployed in an area of size 32m×32m. The area includes 10 randomly distributed obstacles with size 3m×3m.
- Case 2: 32 mesh router nodes with a circular radio coverage range with a radius of 7.5 are to be deployed in an area of size 64m×64m. The area includes 10 randomly distributed obstacles with size 6m×6m.
- Case 3: 200 mesh router nodes with a circular radio coverage range with a radius of 183m are to be deployed in the area of size 4000m×4000m. The area includes 320 randomly distributed obstacles with size 50m×50m.

Parameters used for the algorithms in the comparative evaluation were chosen empirically and were set as follows.

- PSO: 100 particles, 50 iteration limit, with $v_{max} = 3.5$ for the particle velocity.
- CMA-ES: 15 000 iteration limit, $\sigma = 0.01$, with no adaptive σ strategy used.
- RRT-WMN: *timeout* = 10s [30s], $N_B = 10$ [20] and $\psi_{min} = 0.99$ (values in the brackets are for Case 3).

Additionally, since all three algorithms can be run iteratively and in anytime fashion, the first valid solution with its execution time was recorded as well, and it is marked in the results by a capital letter F at the end of an algorithm name (e.g. PSO-F).

For each of the cases, 10 instances of an environment were generated using a uniform distribution. The algorithms were executed 10 times per each instance. Figure 8 presents statistics for Cases 1-2. The coverage measurement is the percentage of the overall deployment area covered i.e. $f(\mathcal{X})$ (see Section 3.1).

In general, for Case 1, the best results were obtained using the CMA-ES and RRT-WMN algorithms, where solutions reached the average coverage of 94.7% and 94.5%, respectively. The CMA-ES algorithm reached a maximum of 97.3% coverage on average which is around 2% better than the best coverage in the RRT-WMN case. However, the average calculation time was slower by a factor of 12, where average CMA-ES times were 62s compared to 4.8s for the RRT-WMN. Interestingly, the first solutions calculated by the RRT-WMN are quite close in quality to the final ones and are computed at an average of 0.2s. The PSO algorithm performed quite poorly in the extended RNP domain as compared to the other two algorithms, providing solutions with an average coverage of 69.6%.

For Case 2, the RRT-WMN outperformed both the CMA-ES and PSO algorithms with average solutions reaching 80% coverage while still maintaining very short average calculation times of 4.8s. In this case, the CMA-ES and PSO algorithms require 16-30 times more computational time than RRT-WMN and provide solutions with lower coverages between 31% and 62%. A representative set of solutions generated by the three algorithms for Case 2 is presented in Fig. 9. The figure shows that the node positions calculated by the RRT-WMN algorithm are farther apart, thus reaching more significant network coverage. The RRT-WMN performed even better on more complex problems. In Case 3, it reached an average coverage of $80.9 \pm 0.5\%$ with computation times at $20.1 \pm 3.4s$. The two other algorithms, CMA-ES and PSO, were not anywhere near feasible for this case due to the amount of time required for finding a solution. A possible reason for the poor performance and scalability of the CMA-ES and PSO approaches when applied to the extended RNP problems could be attributed to the use of a penalty function for constraint handling. Unfortunately, publicly

Fig. 8 Experimental evaluation results for Case 1 and Case 2

available implementations of the CMA-ES and PSO do not offer a better strategy for handling constraints. Thus, the use and evaluation of other variants of these algorithms (e.g. [52, 53]) are left to be investigated in future work.

3.4.2 Evaluation on real-world GIS models

The second part of the evaluation was performed using models based on real-world GIS data from the Swedish government institution Lantmäteriet [54], which offers easy access to a large amount of data. Three cities, shown in Fig. 10, were selected with different area sizes and building structure densities. The goal was to find RNP solutions that cover at least 99% of selected deployment areas using router nodes with a fixed range of 183m (600 feet), with the following locations and parameters:

- Case 4: the city of Gamleby (Fig. 10a) with an approximate deployment area selected equal to 8.2km², including 2157 building structures (obstacles) - maximum of 200 router nodes (i.e. $|R| = 200$), $timeout = 60s$, $N_B = 3$ and $\psi_{min} = 0.99$.

- Case 5: city of Västervik (Fig. 10b) with an approximate deployment area selected equal to 13.05km², including 8763 building structures (obstacles) - maximum of 350 router nodes (i.e. $|R| = 350$), $timeout = 400s$, $N_B = 3$ and $\psi_{min} = 0.99$.
- Case 6: city of Linköping (Fig. 10c) with an approximate deployment area selected equal to 22.1km², including 7801 building structures (obstacles) - maximum of 550 router nodes (i.e. $|R| = 550$), $timeout = 400s$, $N_B = 3$ and $\psi_{min} = 0.99$.

The real-world GIS models and the RNP problems of Cases 4-6 are considerably more complex than in Case 3 described in the previous subsection. As the CMA-ES and the PSO optimization algorithms were not feasible for solving Case 3, these algorithms were also not applicable to problems included in Cases 4-6. Thus, only the RRT-WMN algorithm was applied to these extended cases. Parameters used for the execution of the RRT-WMN algorithm for different environments were chosen experimentally, and alternative selection approaches are left to be considered in future work.

Fig. 9 Representative example solutions generated by the CMA-ES (left), PSO (middle) and RRT-WMN (right) algorithms for Case 2

Fig. 10 Experimental environments based on real geographical data for 3 selected cities in Sweden obtained from the governmental institution Lantmäteriet [54]. Orthophoto with building structures highlighted in orange (left). Overview of problems shown with scale in meters (right): selected area (red), obstacles (blue)

In the case of sparse environments, such as Case 4, it is generally easier to find RNP solutions where each router node range is maximally used. Cases 5 and 6 are more complex and parts of the environments have dense regions. This is especially the case for city centers. There may not be many positions for the router nodes that have a line-of-sight for a given fixed range (i.e. 183m in our experimental setup). Thus, in order to calculate the RNP solution one can place some router nodes closer to each other to increase the RRT-WMN convergence, using adaptive range selection. Adaptive RRT-WMN will place more router nodes in denser parts of the environment.

For each case, 100 runs of the base and adaptive version of the RRT-WMN algorithm were performed to

gather statistics for the algorithm performance evaluation. A summary of the results as averages over all runs is presented in Table 1 and Fig. 11.

In Case 4, the base RRT-WMN algorithm consistently found accepted solutions (solutions that have more than 99% coverage). There was no need to use adaptive step size enhancement in this case. The average time to calculate a solution was 9.72s with these solutions consisting of an average of 170.27 router nodes.

In denser environments such as Cases 5 and 6, there was a considerable difference in the quality of solutions generated by both versions of the RRT-WMN algorithm. Figures 12a, b depict statistics comparing results of applying both algorithm versions for Cases 5 and 6, respectively.

Table 1 Summary of results for experimental evaluation in Case 4-6. Values for coverage, #nodes and time are averages for all 100 runs

Case	Area [km ²]	#obs	RRT-WMN			Adaptive RRT-WMN			Difference		
			Coverage[%]	#nodes	Time[s]	Coverage [%]	#nodes	Time[s]	Coverage [%]	#nodes [%]	Time [%]
4	8.2	2157	99.0741	170.27	9.72	-	-	-	-	-	-
5	13.05	8763	98.3902	291.65	293	98.9557	302.61	196	+0.5655	+3.78	-33.11
6	22.1	7801	98.4428	476	361	98.9632	497.04	246	+0.5204	+4.41	-31.86

The plots on the right show average solution coverage as a function of the RRT-WMN algorithms run time, demonstrating that generally, the adaptive RRT-WMN reaches the desired solution quicker. The plots on the left side depict histograms for solution coverage as a function of the algorithms' runs.

In Case 5, the base RRT-WMN performed quite well, finding 52 solutions with more than 99% coverage. The average coverage for all solutions was 98.39%. Two solutions had a coverage of 95%, and the rest were between 96.5% and 99%. An average of 291.65 nodes were used with an average computation time of 293s. The Adaptive RRT-WMN performed much better, delivering 87 solutions with a greater than 99% coverage level. The rest of the solutions had a coverage between 97.5% and 99%. The average computation time was 196s (33% less), with an average of 302 nodes (3.78% more).

Case 6 turned out to be the most difficult as parts of the environment are quite dense compared to the selected ranges of the router nodes.

The base RRT-WMN algorithm returned 35 solutions with 99% coverage, and 1 solution as low as 95%. This was as expected. The average number of router nodes used was 476, and the average running time was 361s. The adaptive RRT-WMN, on the other hand, consistently found solutions (83 times out of 100), with the average

number of nodes used at 497 (4.41% more). Again as in Case 5, the average computation time decreased to 246s (31.86% less).

4 Router node and gateway placement

The extended RNP problem defined in the previous section, together with the proposed RRT-WMN algorithm, provide a means to efficiently calculate the RN placements that maximize overall network coverage while maintaining its connectivity. Solutions to RNP problems provide WMN backbone network configurations consisting of RNs, which is one of the steps required in the calculation process of the hybrid WMN networks considered in this paper that consist of both router and gateway nodes.

This section considers the Gateway Node Placement (GNP) problem and the combined problem of dealing with both Router and Gateway Node Placement simultaneously. In short, the GNP problem deals with finding a minimum number of GNs and their placements to ensure a sufficient level of QoS based on criteria directly influencing the network performance measures such as communication delay, relay load, and capacity limits of GNs. In the next section, a formal model for the GNP problem, using an Integer Linear Program, is presented. This is followed by descriptions of two approaches used to solve the

Fig. 11 Experimental evaluation results for Cases 4-6

Fig. 12 Results of experimental evaluation for Case 5 and 6

combined RNP and GNP problems. A detailed empirical evaluation of the two approaches then follows.

4.1 Gateway node placement problem definition

The GNP problem formulation used in this paper adopts the model presented in [22]. There the goal is to find a minimum number of GNs and their placements for a given WMN backbone network that satisfies three QoS constraints. The constraints considered include: 1) maximum communication delay (R_{QoS}); 2) maximum relay load for each RN (L_{QoS}); 3) and gateway throughput (S_{QoS}).

Recall that the network topology graph (Section 3.1) is represented as $G = (V, E)$ where V is a set of vertices each representing a router node with its physical x_i, y_i position with $i \in \{1, 2, \dots, n\}$. The set E contains edges between all vertices in G that meet range connectivity and obstacle constraints (Eqs. 4b-4d).

The GNP problem can be formalized as an Integer Linear Program (ILP). Gateway nodes are represented by a set $I, I \subset V$ and each mesh router $v_i \in V$ is a potential location of a gateway. A binary variable a_i is used to indicate if a corresponding mesh router v_i has been selected as a gateway.

$$a_i = \begin{cases} 1, & \text{if router node } v_i \in V \text{ is selected as a gateway,} \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

In order to represent a router node assignment to a gateway, a binary variable $b_{i,j}$ is used. Data traffic generated by each router node will be served by a single gateway.

$$b_{i,j} = \begin{cases} 1, & \text{if router node } v_j \text{ is assigned to gateway } v_i \\ & v_j \in V, v_i \in I, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

The hop count between a router $v_j \in V$ and its associated gateway $v_i \in I$ is represented by $h_{i,j}$. Additionally, a binary variable $z_{i,j}^k$ is used to define whether the shortest path between two nodes v_i and v_j passes through node v_k .

$$z_{i,j}^k = \begin{cases} 1, & \text{if a path from } v_i \text{ to } v_j \text{ passes through node } v_k. \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

Finally, the objective function for the GNP problem is defined as:

$$\min \sum_i^n a_i \quad (16)$$

subject to:

$$\forall v_j \in V : \sum_{v_i \in I} b_{i,j} = 1, \quad (17)$$

$$\forall v_j \in V, v_i \in I : b_{i,j} \leq a_i, \quad (18)$$

$$\forall j \in V : \sum_{v_i \in I} h_{i,j} \cdot b_{i,j} \leq R_{QoS}, \quad (19)$$

$$\forall v_i \in I, v_k \in V : \sum_{v_j \in V} z_{i,j}^k \cdot b_{i,j} \leq L_{QoS}, \quad (20)$$

$$\forall v_i \in I : \sum_{v_j \in V} b_{i,j} \leq S_{QoS}, \quad (21)$$

$$\forall v_i \in I : a_i \in \{0, 1\}, \quad (22)$$

$$\forall v_j \in V, v_i \in I : b_{i,j} \in \{0, 1\}, \quad (23)$$

$$\forall v_j \in V, v_k \in V, v_i \in I : z_{i,j}^k \in \{0, 1\}. \quad (24)$$

The ILP searches for the optimal value for gateway assignment, minimizing their number as defined in Eq. 16, subject to eight constraints. The condition defined in Eq. 17 ensures that each router node is only assigned to one gateway. The inequality specified in Eq. 18 guarantees that a router node v_j is only assigned to a node that has been selected as a gateway. The limit of maximum hop distance, R_{QoS} , between a router node v_j and its assigned gateway v_i is enforced by the inequality specified in Eq. 19. Effectively, this constraint enforces an upper bound on each graph cluster radius. The upper bounds on the remaining two QoS constraints, *relay load* and *cluster size*, are enforced in Eqs. 20 and 21, respectively. The constraints defined in Eqs. 22–24 indicate that a_i , $b_{i,j}$ and $z_{i,j}^k$ are binary variables.

This ILP formulation for the GNP problem is NP-hard since it can be reduced to the minimum set cover problem [14]. Consequently, solving it directly by using popular LP solvers (e.g. Matlab or CPLEX) for larger problem instances is infeasible due to memory requirements. Instead, a number of heuristic approaches have been proposed in the literature (e.g. [14, 21, 22, 37]).

Fundamentally, finding solutions to the GNP problems assumes the router node placements are known. Thus, the RNP-GNP problem is defined as a combination of both RNP and GNP problems described in Section 3.1 and this section, respectively. In the following sections, two algorithms for solving combined RNP-GNP problems are presented. The first approach solves each problem in a

sequence, thus providing solutions to the combined RNP-GNP problems. The second algorithm applies a divide-and-conquer strategy, where the processes of solving RNP and GNP problems are interleaved.

4.2 The sequential RNP-GNP algorithm

One straightforward approach toward solving the RNP and GNP problems simultaneously, that is proposed in this paper, uses a combination of the RRT-WMN algorithm together with one of the heuristic graph clustering approaches. An overview of the Sequential RNP-GNP algorithm is depicted in Fig. 3 (left). In this scheme, the two placement problems are considered separately. The algorithm first solves the RNP problem by using the RRT-WMN algorithm. Then the resulting network topology graph, together with the QoS constraints, are used as input to a graph clustering approach. As a result, the Sequential RNP-GNP algorithm provides a minimum set of clusters (sub-networks) with gateway locations that satisfy the given QoS constraints. As discussed in Section 2.2, many heuristic-based graph clustering approaches have been proposed in the literature. In the approach proposed in this paper, the *Weighted Recursive Dominating Set* algorithm, presented in [22], has been integrated.

The first step in the algorithm ensures that the calculated router node placements for a given deployment region meet the goals of the RNP problem and its constraints. This means that the resulting WMN backbone network configuration maximizes the network coverage, maintaining its connectivity while minimizing the number of RNs used. The second step guarantees that the number of gateway nodes assigned is minimal and the division of the network topology graph into a set of disjoint clusters (sub-networks) satisfies the three QoS constraints R_{QoS} , L_{QoS} , and S_{QoS} .

Although the Sequential RNP-GNP algorithm can efficiently generate solutions to the combined RNP-GNP problems, these solutions may be sub-optimal. This is mainly because the two placement problems are considered separately. Network topology graphs generated by the RRT-WMN algorithm maximize network coverage while utilizing a minimum number of RNs. Thus the generated graphs are *sparse* in nature. Independent of the graph clustering technique choice, the solutions to GNP problems defined for *sparse* network topologies may require more gateways, considering the given QoS constraints. A more elaborate algorithm that interleaves processes of solving RNP and GNP problems is presented in the next section.

4.3 RNP-GNP by area decomposition algorithm

An overview of the proposed algorithm based on area decomposition is depicted in Fig. 13. The main idea of the algorithm is based on a divide-and-conquer strategy

of breaking down the RNP-GNP problem into a set of smaller and simpler RNP problems which are then solved concurrently. Within each RNP problem solution, a single gateway location is optimally chosen.

The RNP-GNP problem specification used by the algorithm includes a deployment area with obstacles, a set of RNs to be distributed, minimum coverage, and QoS constraint requirements. The algorithm starts by dividing the deployment area into several equally sized sub-regions. Router nodes in the input set are then assigned to be distributed within each sub-region. Based on the region division and router assignments, several RNP problems are instantiated (Step 1).

Considering that these RNP problems are independent, they can be solved concurrently by running several RRT-WMN algorithm instances. Solutions generated for each RNP problem are evaluated locally according to minimum coverage requirements in their respective sub-regions. Additionally, one gateway position within each sub-network is selected that minimizes the value of communication delay and relay load in the sub-network topology graph, which is subject to the L_{QoS} and R_{QoS} QoS constraints (Step 2).

The number of RNs necessary to fulfill the network coverage requirements in each sub-region may vary depending on the environment complexity, for example, the obstacle density. Thus, solutions to some RNP problems may require fewer RNs than have been initially assigned, while others may require more RNs to achieve local sub-network coverage goals. The algorithm keeps track of unused RNs for sub-regions where the RNP problem has been successfully solved. These RNs are redistributed among sub-regions where an RNP solution meeting the specified sub-network coverage requirements has not yet been found.

While each sub-network coverage is evaluated locally with respect to its sub-region specification (Step 2), its actual coverage may overlap with neighboring sub-regions contributing to the global coverage over the entire deploy-

ment region specified in the initial RNP-GNP problem. Thus, after all RNP problems are solved individually, the combined coverage over the entire deployment area is evaluated (Step 3). This check provides the opportunity to return solutions as soon as all calculated sub-networks collectively meet the minimum coverage requirement specified by the RNP-GNP problem.

The process of solving individual RNP problems (Step 2), evaluation of generated solutions with respect to coverage over the entire deployment region, and redistribution of unused RNs (Step 3), is repeated until either all generated RNP solutions fulfill the minimum coverage requirements individually or collectively.

Effectively, the number of gateways used in the solution to the RNP-GNP problem is equal to the number of sub-regions created by the area division (Step 1). Network topology graphs including gateways created in each sub-region are evaluated according to the QoS constraints (Step 4). In the case of a constraint violation, the number of sub-regions used during the area division is increased, and the process is repeated (Steps 1-4). Otherwise, the result is reported.

The WMNbyAreaDecomposition algorithm is presented in Algorithm 3. The algorithm is recursive and requires one input parameter which is a number of sub-regions/gateways m . Constant parameters that are used during the calculations that specify the combined GNP-RNP problem include: 1) a set of router nodes R with their range specification; 2) a specification of the deployment area P ; 3) a set of obstacles O ; 4) a minimum target coverage to reach ψ_{min} ; 5) three QoS constraints R_{QoS} , L_{QoS} , and S_{QoS} ; and 6) a maximum number of attempts for solving individual RNP problems N_{opt} . Solutions generated by the algorithm are reported as a set of network topology graphs G_{sol} where each member corresponds to one sub-network configuration. These sub-network configurations include RN positions and GN placements that fulfill minimum coverage requirements and satisfy connectivity and QoS constraints.

Algorithm 3: WMNbyAreaDecomposition

Input: m - number of sub-regions/gateways
Output: $G_{sol} = \{G_1, \dots, G_m\}$
Parameters: $R, P, O, \psi_{min}, R_{QoS}, L_{QoS}, S_{QoS}, N_{opt}$

```

1  $\Omega \leftarrow \text{DivideArea}(m, P, O)$ 
2  $\Gamma \leftarrow \text{AssignRouters}(m, R, \Omega)$ 
3  $G_{sol} \leftarrow$  empty list of RNP graph solutions
4  $U \leftarrow \{i \in \mathbb{Z} : 1 \leq i \leq m\}$ 
5 for  $j = 1$  to  $N_{opt}$  do
6    $R_{free} \leftarrow \emptyset$ 
7   for  $i$  in  $U$  do
8      $G_i = \text{RRT-WMN}(\Gamma_i)$ 
9     if  $\psi(G_i) > \psi(G_{sol}[i])$  then
10        $\text{labelGW}(\text{SelectGW}(G_i))$ 
11        $G_{sol}[i] \leftarrow G_i$ 
12     if  $\psi(G_i) \geq \psi_{min}$  then
13        $R_{free} = R_{free} \cup (R_i \setminus V(G_i))$ 
14        $U \leftarrow U \setminus \{i\}$ 
15   if  $\psi(G_{sol}) \geq \psi_{min}$  then
16     break
17    $\Gamma \leftarrow \text{DistributeRFree}(R_{free}, U, \Gamma)$ 
18 if  $\text{isQoSsSatisfied}(G_i, R_{QoS}, L_{QoS}, S_{QoS})$ 
19    $\forall G_i \in G_{sol}$  then
20   return  $G_{sol}$ 
21 return WMNbyAreaDecomposition( $m + 1$ )
```

The execution of the algorithm starts with a decomposition of the deployment area, RNs assignment, and initialization of individual RNP problem instances together with auxiliary data structures (Step 1, Lines 1-4). The *DivideArea* function (Line 1) divides the deployment area P into a set of m disjoint sub-regions with equal sizes, which can be denoted as:

$$\bigcup_{i=1}^m P_i = P, \quad (25)$$

$$\forall i \neq j \quad P_i \cap P_j = \emptyset,$$

$$\text{Area}(P_i) = \frac{\text{Area}(P)}{m}.$$

Additionally, the function creates m number of O^P sets that represent obstacles intersecting with a sub-region P_i to optimize steps required for line-of-sight connectivity constraint checking. The set generation is defined as:

$$O_i^P = \{\forall O_j \in O, O_j' = O_j \cap P_i : O_j' \neq \emptyset\} \quad (26)$$

The result of the *DivideArea* function is a list of tuples Ω , where each tuple contains a sub-region polygon specification and a set describing obstacles associated with it, $\langle P_i, O_i^P \rangle$ for $\forall i \in \{1, 2, \dots, m\}$.

The polygon partitioning defined in Eq. 25 is known as the *area partitioning problem* [55]. In the general case of the area partitioning problem, area sizes of sub-regions are defined by a set of weights so that the division can be made with varying area proportions. The sub-region sizes used in the *DivideArea* function are assumed to be equal; hence weight values are equal. Initially, the polygon decomposition proposed in [55] was used to divide a deployment area.

Unfortunately, the sub-polygon regions generated by the algorithm were sub-optimal when applied in the RNP-GNP application domain due to their shape characteristics. The algorithm guarantees that the area size constraints are satisfied, but no other constraints are considered. Generated sub-region shapes most often included

polygons with small angles between vertices (sharp corners). Therefore, the resulting network topology graph solutions to the RNP-GNP problems based on such divisions produced sub-optimal network configurations with longer relay chains that increase communication delays and RN relay loads.

Instead, an alternative algorithm presented in [56] is used. The algorithm takes into account both area and shape constraints and maximizes sub-polygon compactness based on the compactness score proposed in [57]. Generally, by imposing the shape constraints, the algorithm avoids generating sub-polygons with sharp corners. Consequently, solutions to RNP problems defined over compact deployment areas provide better quality network configurations with improved performance characteristics.

After the deployment area has been decomposed, the *AssignRouters* function is executed (Line 2). This deals with the allocation of RNs to specific sub-regions to define a set of independent RNP problems. The set of routers R is divided and assigned to be used in each sub-region deployment area defined in Ω . The number of routers assigned to each sub-region is equal, and their selection is random.

Other heuristic-based methods could be considered in this step, such as considering router node range specification or other properties. This, however, is left to be investigated in future work.

Based on the area divisions defined in Ω and the RN assignments, m number of RNP problems are defined and stored in a list Γ . Each RNP problem specification Γ_i contains all parameters required by the RRT-WMN algorithm. These parameters include a deployment area specification P_i , a set of obstacles intersecting with the area O_i^P and a set of routers, R_i , to be distributed, where $i \in \{1, 2, \dots, m\}$.

Two auxiliary data structures are initialized in Lines 3-4. The first is a list G_{sol} that keeps track of network topology graphs G_i generated by the RRT-WMN algorithm for each RNP problem Γ_i . The Γ_i problem is considered to be *solved* if a solution G_i is found that satisfies the minimum coverage requirement Ψ_{min} for its sub-region. All solutions generated by the RRT-WMN algorithm are stored whether they meet the minimum coverage requirement for a sub-region or not. The second data structure is a set U (*Unsolved*) that is used to keep track of the status for each solution. The set contains values of indices i corresponding to RNP problems Γ_i that remain unsolved. Initially, the set U contains all indices $i = [1, 2, \dots, m]$ as none of the RNP problems have been solved yet.

The main loop of the algorithm (Steps 2-3, Lines 5-17) executes the RRT-WMN algorithm for each RNP problem instance in Γ , evaluates resulting network topology graphs according to their sub-regions, selects the

gateway locations within each sub-network, and applies RN redistribution among unsolved RNP problem instances. The N_{opt} parameter defines the limit on the number of iterations of the loop.

As discussed previously, the difficulty of finding a solution for an RNP problem Γ_i that satisfies coverage requirements in its sub-region depends on the deployment area characteristics and the density of obstacles. Some instances of RNP problems may require fewer RNs than the initial random equal assignment (Line 2), and some may need more in order to fulfill the given minimum coverage. The algorithm tries to solve each RNP problem up to N_{opt} times to rebalance the initial assignment of the RNs. In between each run, the RNs that were redundant in successfully solved RNP problem instances are redistributed among the remaining unsolved problems. A set R_{free} is initialized in Line 6, and it is used to keep track of RNs that have been redundant whenever a solution meeting the coverage criteria is found (Lines 12-13).

In every loop iteration, values of indices contained in the set U determine which of the RNP problem instances are executed using the RRT-WMN algorithm (Line 8). Each new solution to the RNP problem represented as a network topology graph G_i , is evaluated (Line 9) as to whether its quality (i.e. coverage ψ) has improved over the solution calculated in previous iterations ($G_{sol}[i]$). If an improved RNP solution has been found, the optimal gateway location is selected, labeled in the solution graph (Line 10), and stored in the list of current solutions (Line 11). Additionally, whenever the newly calculated solution fulfills the local coverage requirements, the problem index is removed from the set U (Line 14). Therefore, solved RNP problems will not be selected for execution with the RRT-WMN algorithm in the next iteration.

Recall the definitions of the three QoS constraints introduced previously. A gateway throughput (S_{QoS}) considers the bandwidth capacity of gateway nodes since the traffic within each sub-network (G_i) has to be potentially served by a gateway. Effectively, the S_{QoS} is an upper bound on the size of the network topology graph ($|G_i|$) calculated for a particular sub-region. The resulting sub-network sizes generated during the algorithm execution depend on the performance/effectiveness of the interleaved process of solving RNP problems with the RRT-WMN algorithm combined with the RN redistribution strategy.

Satisfying the other two QoS constraints, however, depends on the selection of gateway locations within the calculated sub-network topology graphs G_i . The first constraint to consider puts bounds on the maximum communication delay (R_{QoS}) and it is formulated as the maximum number of hops information has to travel in between a router node and the gateway to which the RN is assigned to, as each re-transmission adds to the total delay. The second constraint limits the maximum relay load for each

RN (L_{QoS}) and it is defined as an upper bound on the number of router nodes that rely on re-transmission in shortest paths that lead to a gateway from all RNs.

The gateway placement calculation that minimizes values of sub-network performance measures related to both constraints is performed by the *SelectGW* function. The selection of the gateway location within each solution graph G_i is done in a two-step process defined by the following expression:

$$\operatorname{argmin}_{c_j \in C(G_i)} \operatorname{Calc}L_{QoS}(\operatorname{BreadthFirstSearch}(c_j, G_i)). \quad (27)$$

First, a set of vertices that are centers of the solution graph $C(G_i)$ are calculated. Each of these vertices (c_j) are potential gateway locations that guarantee minimal communication delay. The graph centers are defined as a set of all vertices u where the greatest hop distance $d(u, v)$ to other vertices v in the graph is minimal. Selecting any of these vertices as a gateway location minimizes the delay, subject to the R_{QoS} constraint.

In the second step, the relay load for all alternatives is calculated by generating a set of shortest-distance trees rooted in each candidate location using Breadth-First-Search (BFS). The location which yields the minimum value of relay load is selected. This heuristic-based approach is based on the work presented in [21].

An example illustration of this process for a simple example is presented in Fig. 14. In this case, a solution graph to the RNP Problem G_i consists of six vertices $v_1, v_2, \dots, v_6$. The set of graph centers includes vertices v_2 and v_3 as the hop distances between those and the remaining nodes are minimal and equal to 2. Selecting either of them will result in the same network delay. The choice between v_2 and v_3 will, in fact, influence the maximum relay load required. The shortest-path trees rooted in both candidate gateway locations generated by the BFS search presented in the figure show that the maximum relay load equals 3 and 1 for v_2 and v_3 candidates, respectively. Thus the *SelectGW* selects the v_3 as the gateway location.

The execution of the main loop (Lines 5-17) terminates when one of two conditions occurs. First, the iteration limit N_B has been reached. Second, solutions to all RNP problems generated so far, collectively meet the minimum coverage requirements over the whole deployment area (Line 15). Since the RNP problem solved in the main loop is independent, the RRT-WMN algorithm can be executed in parallel to decrease the total calculation time.

In the final step of the algorithm (Step 4, Lines 18-20), the three QoS constraints are checked, and if satisfied the list of solution graphs G_{sol} is returned. Otherwise, the WMNbyAreaDecomposition algorithm is executed with an increased number of sub-networks ($m = m + 1$).

The choice of the input parameter m influences the execution time of the algorithm. In the general case, one can

start with the initial value of m equal to 1. However, in many cases, it may be possible to estimate the lower bound on the number of sub-networks the network requires given QoS constraints. Thus, the algorithm execution can be initialized with higher values of m , cutting down the execution times. For example, assume the hybrid WMN network that is being calculated uses homogeneous RNs that have equal range specifications. The number of RNs required to reach the minimum coverage requirements for a given deployment area can be obtained by setting the number of RNs in R sufficiently high and running the RRT-WMN algorithm once. The resulting number of RNs in the solution graph ($|G|$) and the cluster size QoS constraint (S_{QoS}) specifying the maximum number of nodes in each sub-network can be used as a lower bound for the parameter value m as follows:

$$m = \text{floor} \left(\frac{|R|}{S_{QoS}} \right). \quad (28)$$

The convergence of the WMNbyAreaDecomposition algorithm highly depends on the polygon partitioning algorithm selected to be used in Line 1. Unfortunately, the partitioning algorithms available in the literature do not allow for seamless integration of the RNP-GNP problem objectives (i.e. maximizing the network coverage while minimizing the number of gateways) while the sub-region shapes are generated. Therefore, when the WMNbyAreaDecomposition algorithm uses the partitioning algorithms considered in this work (i.e. [55, 56]), the algorithm is not guaranteed to converge to a global optimum. The development of new polygon partitioning techniques suited for the RNP-GNP problems will be considered in future work.

4.4 Experimental evaluation for solving combined RNP-GNP problems

This section provides a comparative evaluation between the two proposed algorithms for solving the combined RNP-GNP problems performed using the three real-world models (Gamleby, Västervik and Linköping) presented in

Section 3. Other approaches to solving the RNP and GNP problems simultaneously [10, 49] are not directly applicable to continuous environment models and, as such, have not been included in the comparison.

Generally, the results showed that the WMNbyAreaDecomposition outperforms the Sequential RNP-GNP algorithm in all considered cases. The generated network topology graphs, on average, have a much lower number of gateways required for the same QoS constraints across all R_{QoS} , L_{QoS} , S_{QoS} choices. At the same time, the generated solutions maintain the specified minimum coverage, satisfy the network connectivity constraints and utilize a minimum number of router nodes. For that reason, the following strategy for comparison was used. First, experiments were run on the three environments using the WMNbyAreaDecomposition algorithm with relaxed values of $R_{QoS} = L_{QoS} = S_{QoS} = NaN$ to find the minimum values for these constraints. The resulting QoS values were recorded and used to set up a set of problems to be executed with the Sequential RNP-GNP algorithm for evaluation. The number of gateways in each solution, given the same QoS constraint values, was used to compare the solutions generated by both algorithms.

In the first set of experiments, the following set of parameters were used during the WMNbyAreaDecomposition algorithm execution:

- Case 4: the city of Gamleby (Fig. 10a, 8.2km² deployment area including 2157 building structures) - maximum of 170 router nodes ($|R| = 170$), $timeout = 60s$, $N_B = 3$, $\psi_{min} = 0.99$, $N_{opt} = 5$ and $m \leq 12$ (up to 12 gateway nodes).
- Case 5: city of Västervik (Fig. 10b, 13.05km² deployment area including 8763 building structures) - maximum of 302 router nodes ($|R| = 302$), $timeout = 400s$, $N_B = 3$, $\psi_{min} = 0.99$, $N_{opt} = 5$ and $m \leq 17$ (up to 17 gateway nodes).
- Case 6: city of Linköping (Fig. 10c, 22.1km² including 7801 building structures) - maximum of 497 router

nodes ($|R| = 497$), $timeout = 400s$, $N_B = 3$, $\psi_{min} = 0.99$, $N_{opt} = 5$ and $m \leq 25$ (up to 25 gateway nodes).

The RNs are assumed to be homogeneous, with their maximum ranges specified as 183m (600 feet). The number of nodes to be used in each problem $|R|$ is set according to the average results obtained during the RRT-WMN algorithm experimental evaluations for respective environments presented in Section 3.4.2. Limits on the number of gateways in each environment were selected based on the total number of RNs to obtain sub-networks with reasonably small sizes between 14–20 router nodes. Other parameters used during the algorithm execution were chosen empirically. A total of 10 runs of the algorithm per environment were performed, and the average values of the QoS measures were recorded.

The results of the experimental evaluation of the WMNbyAreaDecomposition algorithm for all environments are depicted in Fig. 15. Plots present the three QoS measures as a function of the number of gateways assigned. The communication delay expressed as the maximum number of hops between a router and a gateway node (subject to R_{QoS}) is shown on the left. The relay load (subject to L_{QoS}) calculated as the maximum number of router nodes that rely on re-transmission in shortest paths that lead to a gateway from all RNs is shown in the middle. The last of the QoS measures considers the gateway throughput (subject to S_{QoS}), expressed as the maximum number of RNs assigned to a gateway, is shown on the right.

In general, as expected, the QoS measures related to the network performance decrease (less is good) when the number of gateways/sub-networks increases. This trend is independent of the environment complexity since the network configurations with a higher number of gateways will naturally exhibit better performance. In Case 4, where the simplest of the three environments is considered, the algorithm finishes with network configurations that include 12 gateways with the following characteristics: the maximum number of RNs in generated sub-network configurations is 15; the maximum relay load of RNs is decreased to 8; and the maximum distance between RNs and GNs is reduced to 5 hops.

In the remaining more complex environments, the QoS measures for final WMN network configurations are higher. In Case 5, sub-networks including up to 19 RNs are generated with the maximum relay load for RNs of 10 and the maximum distance between RNs and GNs equal to 7 hops. Similarly, in Case 6, generated sub-networks consist of up to 22 RNs with the maximum values of relay loads and distances between RNs and GNs equal to 13 and 7, respectively.

Naturally, these QoS measure values cannot be compared directly as the problems are different in terms of

the deployment area size and the number of RNs and GNs. However, a general observation from the experimental evaluations in the cases considered showed that the average values of relay load and RN-GN distances in most generated sub-networks were lower.

In Cases 5 and 6, a small number of sub-networks typically require higher bounds on R_{QoS} , L_{QoS} and S_{QoS} constraints. These are associated with dense parts of the environments such as city centers. Correlations between environment model characteristics such as obstacle density and the difficulty of sub-network calculations could potentially be used to improve the proposed algorithm's performance further. However, this is left for future work.

The experiments were run on a computer equipped with an Intel Core i7 (I7-6820HQ) processor with the algorithm executed on a single core. The average algorithm run-time for one value of m was recorded as 278.3s, 619.4s, and 887.1s for Cases 4–6, respectively. In the general case, the total execution times when considering an RNP-GNP problem and specific values of QoS constraints will depend on the selection of the starting parameter m . Additionally, the execution times will decrease when using concurrent (multi-core) RRT-WMN algorithm execution.

The final part of the evaluation involves a comparison between the results obtained from both the WMNbyAreaDecomposition and the Sequential RNP-GNP algorithms. The QoS constraints given by solutions found by the first algorithm were used to set up a set of problems to be executed with the Sequential RNP-GNP algorithm. The number of gateways in each solution, given the same QoS constraint values, was used as a metric to compare the solutions generated by both algorithms. Results are presented in Fig. 16.

The WMNbyAreaDecomposition algorithm provided WMN network configurations with an average of 78.3%, 73.7%, 92.5% fewer gateway nodes required for Case 4 (Gamleby), Case 5 (Västervik) and Case 6 (Linköping), respectively.

5 Field deployment of ad hoc wireless mesh networks

This paper has proposed a number of complex algorithms for determining the efficient generation of ad hoc wireless mesh networks that meet QoS constraints in diverse geographical environments. The long-term goal is to autonomously deploy such networks using teams of UAVs, where the algorithms proposed generate placement of router and gateway node positions and planning techniques generate motion plans to deliver and deploy physical routers and gateways. This would result in an end-to-end solution to the problem targeted in this paper. In this section, some previous work to that effect [4] is summarized, where limited field experiments have been done. UAV functionality for rapidly delivering router and

Fig. 15 Results of experimental evaluation of the WMNbyAreaDecomposition algorithm for Cases 4-6 with relaxed QoS constraints (i.e. $R_{QoS} = L_{QoS} = S_{QoS} = NaN$)

gateway nodes for ad hoc wireless mesh networks has been developed that uses deployable *CommKits* as network nodes. These *CommKits* can be delivered based on the output of the algorithms described in this paper. In initial experimentation, smaller autonomous DJI quadrotor systems have been used (Fig. 17). These systems are extended to deliver communication equipment to operational environments autonomously through in-air deployment and even serve as airborne routers to link local ad hoc networks that have been already deployed. Eventually,

larger autonomous RMAX systems could be used to carry and deploy several *CommKits* at once.

The design of the *CommKit* has been driven by a number of requirements. First, it should be small and lightweight to allow for in-air deployment even from smaller UAV platforms, while at the same time it should be robust to withstand a certain amount of shock when hitting the ground if deployed from air. The solution to this integrates a parachute for deployment that minimizes forces generated at landing.

Fig. 16 Comparison between 1) the Sequential RNP-GNP approach and 2) the WMNbyAreaDecomposition placement algorithms. Plots visualizing a number of additional gateways required when using approach 1) instead of 2) given the same values of R_{QoS} , L_{QoS} and S_{QoS} QoS constraints are depicted on the left. Results shown in percent are presented on the right

Fig. 17 The DJI Matrice 100 system with a CommKit attached

Fig. 18 The CommKit hardware architecture

Fig. 19 The two CommKit prototypes

Secondly, CommKits should be stand-alone devices containing their own electrical power source, computational resources, and sensors such as a GPS receiver and Inertial Measurement Unit (IMU). These components are essential for implementing a number of required functionalities such as independent release of CommKits from a carrying platform, optimization and repair of a deployed ad hoc mesh network, and self-monitoring and safe in-air delivery. This sophistication is required since the goal is for any deployment mission to be 100% autonomous.

The CommKit hardware architecture is depicted in Fig. 18. The central component of the system is a WiFi Access Point (AP) which is used for the creation of an ad hoc wireless network between multiple CommKits. Two variants of the UniFi APs from Ubiquiti¹ have been chosen for evaluation and integrated into the design: the AC Lite and AC Mesh with 100m and 183m advertised communication ranges, respectively.

Two prototype CommKit units (see Fig. 19) have been built with their specifications summarized in Table 2.

The internal components of a CommKit are mounted on a stack which is then put in the enclosure lined with protective foam in order to ensure that the components can withstand the forces of the impact when the box touches down.

Promising experimental evaluation using CommKits as dynamically deliverable, environment aware, router and gateway nodes, for the generation of ad-hoc wireless mesh networks in the field has been performed [4]. The

algorithms in this paper, together with the physical functionality, provide the basis for an end-to-end system for generation and deployment of ad hoc wireless networks using teams of UAVs.

6 Conclusions

This paper proposed an end-to-end solution to the challenging problem of autonomously generating ad hoc Wireless Mesh Networks in operational environments where emergency rescue teams are intended to be deployed. Teams of specially equipped UAVs are used to deploy communication kits from the air that include routers and battery supplies [4]. This paper has targeted a solution to three fundamental problems involved in this endeavor: the Router Node Placement (RNP) problem, the Gateway Node Placement (GNP) problem, and the combined simultaneous RNP-GNP problem.

New and novel algorithms were proposed to solve these problems. The RRT-WMN algorithm was used as a basis to solve all three problems. It appeals to the use of Rapidly Exploring Random Tree algorithms used in motion planning. Two alternative algorithms were proposed to solve the combined RNP-GNP problem, the first using a combination of the RRT-WMN algorithm with a pre-existing graph clustering algorithm and the second using a new WMNbyAreaDecomposition algorithm based on a divide-and-conquer strategy that reduces the problem to a number of concurrently solved RNP problems.

Empirical results have shown that these algorithms are not only scalable to real-life emergency rescue scenarios

¹ubnt.com

Table 2 Prototype CommKit box specifications

Name	Parachute system	Access point	Weight [g]	Dimensions [cm]
CommKit0	Internal Mars Mini V2	AC Lite	1050	15×20×5.5
CommKit1	External Mars P2 Lite	AC Mesh	990	16×16×6

but are also state of the art in terms of efficiency and speed, as shown by comparative analyses with existing algorithms. It is important to note that the generalization of these problems proposed in the paper which introduces additional constraints to deal with obstacles and lack of knowledge as to where clients are positioned in an operational region, makes these problems not only more difficult for existing algorithms, but also often makes them out of reach in terms of scalability criteria.

The algorithms proposed in this paper specifically target deployment scenarios with these real-life complexities making them more appropriate for use in the field. This is shown to be the case based on the empirical analysis of the approach and limited field robotics experiments that demonstrate the deployment concept.

Authors' contributions

M. Wzorek conceived the original ideas described in the contribution section. The author also performed presented experimental evaluations and took lead in writing of the manuscript. Both C. Berger and P. Doherty provided critical feedback and contributed to the final version of the manuscript. P. Doherty supervised the project. The authors read and approved the final manuscript.

Funding

Supported by the ELLIIT Network Organization for Information and Communication Technology, Sweden; the Swedish Foundation for Strategic Research SSF (Smart Systems Project RIT15-0097); and the Wallenberg AI, Autonomous Systems and Software Program: WASP WARA-PS Project. The 3rd author is also supported by an RExperts Program Grant 2020A1313030098 from the Guangdong Department of Science and Technology, China and a Sichuan Province International Science and Technology Innovation Cooperation Project Grant 2020YFH0160.

Availability of data and materials

The geographical data used in the evaluation are available from the governmental institution Lantmäteriet [54] but restrictions apply to the availability of these data, which were used under license for this study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of Lantmäteriet.

Code availability

No custom code submitted, available from the authors upon reasonable request.

Declarations

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author details

¹Department of Computer and Information Science, Linköping University, Linköping, Sweden. ²School of Intelligent Systems and Engineering, Jinan University (Zhuhai Campus), Zhuhai 519070, China.

Received: 10 May 2021 Accepted: 7 October 2021

Published online: 02 December 2021

References

- P. J. Denning, Hastily formed networks. *Commun. ACM*. **49**(4), 15–20 (2006). <https://doi.org/10.1145/1121949.1121966>
- United Nations Office for Coordination of Humanitarian Affairs (UNOCHA), 2010 Haiti Earthquake Response: An After Action Review of Response. INSARAG Haiti Earthquake After-Action Review Meeting. Geneva (2010). https://www.insarag.org/wp-content/uploads/2016/04/Printed_Haiti_Book_Low_Definition_Version.pdf. Accessed Oct 2021
- C. Berger, P. Doherty, P. Rudol, M. Wzorek, Hastily formed knowledge networks and distributed situation awareness for collaborative robotics. *arXiv preprint arXiv:2103.14078* (2021)
- M. Wzorek, C. Berger, P. Rudol, P. Doherty, in *International Electrical Engineering Congress (IEEECON)*. Deployment of ad hoc network nodes using uavs for search and rescue missions, (2018). <https://doi.org/10.1109/IEEECON.2018.8712230>
- P. Gupta, P. R. Kumar, The capacity of wireless networks. *IEEE Transactions on information theory*. **46**(2), 388–404 (2000). <https://doi.org/10.1109/18.825799>
- J. Li, C. Blake, D. S. De Couto, H. I. Lee, R. Morris, in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. Capacity of ad hoc wireless networks, (2001), pp. 61–69. <https://doi.org/10.1145/381677.381684>
- J. Jun, M. L. Sichitiu, The nominal capacity of wireless mesh networks. *IEEE Wirel. Commun.* **10**(5), 8–14 (2003)
- P. Kyasanur, N. H. Vaidya, in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*. Capacity of multi-channel wireless networks: impact of number of channels and interfaces, (2005), pp. 43–57. <https://doi.org/10.1145/1080829.1080835>
- B. He, B. Xie, D. P. Agrawal, in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. Optimizing the internet gateway deployment in a wireless mesh network (IEEE, Pisa, 2007), pp. 1–9. <https://doi.org/10.1109/MOBHOC.2007.4428603>
- L. Shillington, D. Tong, Maximizing wireless mesh network coverage. *Int. Reg. Sci. Rev.* **34**(4), 419–437 (2011)
- H. Suzuki, Y. Kaneko, K. Mase, S. Yamazaki, H. Makino, in *IEEE Vehicular Technology Conference*. An ad hoc network in the sky, skymesh, for large-scale disaster recovery (IEEE, 2006), pp. 1–5. <https://doi.org/10.1109/VTFC.2006.496>
- R. B. Dilmaghani, R. R. Rao, Hybrid wireless mesh network with application to emergency scenarios. *J. Softw.* **3**(2), 52–60 (2008)
- C.-C. Lin, Dynamic router node placement in wireless mesh networks: A pso approach with constriction coefficient and its convergence analysis. *Inf. Sci.* **232**, 294–308 (2013). <https://doi.org/10.1016/j.ins.2012.12.023>
- Y. Drabu, H. Peyravi, in *Seventh International Conference on Networking (Icn 2008)*. Gateway placement with QoS constraints in wireless mesh networks (IEEE, Cancun, 2008), pp. 46–51. <https://doi.org/10.1109/ICN.2008.89>
- G. Lee, A. T. Murray, Maximal covering with network survivability requirements in wireless mesh networks. *Comput. Environ. Urban. Syst.* **34**(1), 49–57 (2010). <https://doi.org/10.1016/j.compenvurbsys.2009.05.004>
- C.-C. Lin, T.-H. Chen, H.-H. Chin, Adaptive router node placement with gateway positions and qos constraints in dynamic wireless mesh networks. *J. Netw. Comput. Appl.* **74**, 149–164 (2016). <https://doi.org/10.1016/j.jnca.2016.05.005>
- F. Xhafa, C. Sánchez, L. Barolli, in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. Genetic algorithms for efficient placement of router nodes in wireless mesh networks (IEEE, Perth, 2010), pp. 465–472. <https://doi.org/10.1109/AINA.2010.41>
- C.-C. Lin, P.-T. Tseng, T.-Y. Wu, D.-J. Deng, Social-aware dynamic router node placement in wireless mesh networks. *Wirel. Netw.* **22**(4), 1235–1250 (2016). <https://doi.org/10.1007/s11276-015-1036-7>
- C.-C. Lin, L. Shu, D.-J. Deng, Router node placement with service priority in wireless mesh networks using simulated annealing with momentum terms. *IEEE Syst. J.* **10**(4), 1402–1411 (2016). <https://doi.org/10.1109/JSYST.2014.2341033>
- J. J. Kuffner, S. M. LaValle, in *Proc. ICRA*. RRT-connect: An efficient approach to single-query path planning, (2000). <https://doi.org/10.1109/ROBOT.2000.844730>
- Y. Bejerano, Efficient integration of multihop wireless and wired networks with qos constraints. *IEEE/ACM Trans. Netw.* **12**(6), 1064–1078 (2004). <https://doi.org/10.1109/TNET.2004.838599>
- B. Aoun, R. Boutaba, Y. Iraqi, G. Kenward, Gateway placement optimization in wireless mesh networks with qos constraints. *IEEE J. Sel. Areas Commun.* **24**(11), 2127–2136 (2006). <https://doi.org/10.1109/JSAC.2006.881606>
- M. Wzorek, C. Berger, P. Doherty, in *PRICAI 2019: Trends in Artificial Intelligence*, ed. by A. C. Nayak, A. Sharma. Router node placement in wireless mesh networks for emergency rescue scenarios (Springer, Cham, 2019), pp. 496–509. https://doi.org/10.1007/978-3-030-29911-8_38

24. N. Hansen, *The CMA Evolution Strategy: A Comparing Review*. (Springer, Berlin, Heidelberg, 2006), pp. 75–102. https://doi.org/10.1007/3-540-32494-1_4
25. R. Eberhart, J. Kennedy, in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. A new optimizer using particle swarm theory, (1995), pp. 39–43. <https://doi.org/10.1109/MHS.1995.494215>
26. D. B. Johnson, D. A. Maltz, J. Broch, et al, Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc Netw.* **5**(1), 139–172 (2001)
27. C. Perkins, E. Belding-Royer, S. Das, RFC3561: Ad hoc on-demand distance vector (AODV) routing. RFC Editor (2003). <https://doi.org/10.17487/RFC3561>
28. R. Ogier, F. Templin, M. Lewis, *RFC3684: Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)*. (RFC Editor, USA, 2004). <https://doi.org/10.17487/RFC3684>
29. A. Neumann, C. Aichele, M. Lindner, S. Wunderlich, *Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)* (Internet Engineering Task Force, 2008), p. 24. <https://datatracker.ietf.org/doc/html/draft-openmesh-b-a-t-m-a-n-00>
30. P. Kyasanur, N. H. Vaidya, in *IEEE Wireless Communications and Networking Conference, 2005*. Routing and interface assignment in multi-channel multi-interface wireless networks, vol. 4 (IEEE, New Orleans, 2005), pp. 2051–2056. <https://doi.org/10.1109/WCNC.2005.1424834>
31. K. N. Ramachandran, E. M. Belding-Royer, K. C. Almeroth, M. M. Buddhikot, in *Infocom*. Interference-aware channel assignment in multi-radio wireless mesh networks, vol. 6, (2006), pp. 1–12. <https://doi.org/10.1109/INFOCOM.2006.177>
32. R. Ramanathan, in *Proceedings of the 2nd ACM International Symposium on Mobile Ad-hoc Networking & Computing*. On the performance of ad hoc networks with beamforming antennas, (2001), pp. 95–105. <https://doi.org/10.1145/501426.501430>
33. S. D. Blostein, H. Leib, Multiple antenna systems: their role and impact in future wireless access. *IEEE Commun. Mag.* **41**(7), 94–101 (2003). <https://doi.org/10.1109/MCOM.2003.1215645>
34. T. Naeem, J. K.-K. Loo, Common Security Issues and Challenges in Wireless Sensor Networks and IEEE 802-11 Wireless Mesh Networks. *J. Dig. Content Technol. Appl.* **3**, 88–93 (2009). <https://doi.org/10.4156/jdcta.vol3.issue1.naeem>
35. S. Khan, J. L. Mauri, *Security for Multihop Wireless Networks*. (Crc Press, Boca Raton, 2014)
36. A.-S. K. Pathan, *Security of self-organizing networks: MANET, WSN, WMN, VANET*. (CRC press, Boca Raton, 2016)
37. T. Maolin, et al, Gateways placement in backbone wireless mesh networks. *Int. J. Commun. Netw. Syst. Sci.* **2**(01), 44 (2009). <https://doi.org/10.4236/ijcns.2009.21005>
38. A. A. Franklin, C. S. R. Murthy, in *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*. Node placement algorithm for deployment of twotier wireless mesh networks (IEEE, Washington DC, 2007), pp. 4823–4827. <https://doi.org/10.1109/GLOCOM.2007.915>
39. J. Camp, J. Robinson, C. Steger, E. Knightly, in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*. Measurement driven deployment of a two-tier urban mesh access network, (2006), pp. 96–109. <https://doi.org/10.1145/1134680.1134691>
40. F. Xhafa, A. Barolli, C. Sánchez, L. Barolli, A simulated annealing algorithm for router nodes placement problem in wireless mesh networks. *Simul. Model. Pract. Theory.* **19**(10), 2276–2284 (2011). <https://doi.org/10.1016/j.simpat.2010.08.014>
41. F. Xhafa, A. Barolli, M. Takizawa, et al., in *2011 Third International Conference on Intelligent networking and collaborative systems*. A tabu search algorithm for efficient node placement in wireless mesh networks (IEEE, Fukuoka, 2011), pp. 53–59. <https://doi.org/10.1109/INCoS.2011.44>
42. S. Sakamoto, E. Kulla, T. Oda, M. Ikeda, L. Barolli, F. Xhafa, A comparison study of simulated annealing and genetic algorithm for node placement problem in wireless mesh networks. *J. Mob. Multimedia.* **9**(1&2), 101–110 (2013)
43. T. Oda, Y. Liu, S. Sakamoto, D. Elmazi, L. Barolli, F. Xhafa Xhafa, Analysis of mesh router placement in wireless mesh networks using friedman test considering different meta-heuristics. *Int. J. Commun. Netw. Distrib. Syst.* **15**(1), 84–106 (2015). <https://doi.org/10.1504/IJCNDS.2015.070289>
44. M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **32**(200), 675–701 (1937). <https://doi.org/10.1080/01621459.1937.10503522>
45. A. Barolli, T. Oda, M. Ikeda, L. Barolli, F. Xhafa, V. Loia, Node placement for wireless mesh networks: Analysis of wmn-ga system simulation results for different parameters and distributions. *J. Comput. Syst. Sci.* **81**(8), 1496–1507 (2015). <https://doi.org/10.1016/j.jcss.2014.12.024>
46. B. He, B. Xie, D. P. Agrawal, Optimizing deployment of internet gateway in wireless mesh networks. *Comput. Commun.* **31**(7), 1259–1275 (2008). <https://doi.org/10.1016/j.comcom.2008.01.061>
47. M. Seyedzadegan, M. Othman, B. M. Ali, S. Subramaniam, Zero-degree algorithm for internet gateway placement in backbone wireless mesh networks. *J. Netw. Comput. Appl.* **36**(6), 1705–1723 (2013). <https://doi.org/10.1016/j.jnca.2013.02.031>
48. J. Wang, B. Xie, K. Cai, D. P. Agrawal, in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. Efficient mesh router placement in wireless mesh networks (IEEE, Pisa, 2007), pp. 1–9. <https://doi.org/10.1109/MOBHOC.2007.4428616>
49. E. Amaldi, A. Capone, M. Cesana, I. Filippini, F. Malucelli, Optimization models and methods for planning wireless mesh networks. *Comput. Netw.* **52**(11), 2159–2171 (2008). <https://doi.org/10.1016/j.comnet.2008.02.020>
50. F. J. Solis, R. J.-B. Wets, Minimization by random search techniques. *Math. Oper. Res.* **6**(1), 19–30 (1981)
51. F. Xhafa, C. Sánchez, A. Barolli, M. Takizawa, Solving mesh router nodes placement problem in wireless mesh networks by tabu search algorithm. *J. Comput. Syst. Sci.* **81**(8), 1417–1428 (2015). <https://doi.org/10.1016/j.jcss.2014.12.018>
52. D. V. Arnold, N. Hansen, in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. A (1+ 1)-cma-es for constrained optimisation, (2012), pp. 297–304
53. M. Kohler, L. Forero, M. Vellasco, R. Tanscheit, M. A. Pacheco, in *2016 IEEE Congress on Evolutionary Computation (CEC)*. Pso+: A nonlinear constraints-handling particle swarm optimization, (2016), pp. 2518–2523. <https://doi.org/10.1109/CEC.2016.7744102>
54. Lantmäteriet, Lantmäteriet open geodata (2020). <https://lantmateriet.se/en/maps-and-geographic-information/open-geodata/>. Accessed Dec 2019
55. S. Hert, V. Lumelsky, Polygon area decomposition for multiple-robot workspace division. *International Journal of Computational Geometry and Applications.* **8**, 437–466 (1998). <https://doi.org/10.1142/S0218195998000230>
56. M. Wzorek, C. Berger, P. Doherty, Polygon Area Decomposition using a Compactness Metric. *arXiv preprint arXiv:2110.04043* (2021)
57. J. E. Schwartzberg, Reapportionment, gerrymanders, and the notion of compactness. *Minn. L. Rev.* **50**, 443 (1965)

Publisher's Note

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