Spatial Prediction of Wireless Links and Its Application to the Path Control of Mobile Robots

Eduardo Feo Flushing, Michal Kudelski, Luca M. Gambardella and Gianni A. Di Caro
Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland
Email: {eduardo,michal,luca,gianni}@idsia.ch

***Abstract*—We consider the path planning problem of a mobile robot that has to travel towards a given target location. The robot shares the environment with other mobile robots, altogether forming a wireless mobile ad hoc network relaying data in a multi-hop fashion. In this scenario, the robot's path planner has to optimally balance two potentially conflicting goals: keep the traveled distance within an assigned maximum value while letting the robot effectively communicate with the other robots in the network. We propose a solution method relying on the use of a link quality predictor built offline through a supervised learning approach. Together with the information gathered online from the other robots, the predictor allows to adaptively build a spatial map of expected communication quality, for both local and distant areas. In turn, the map is used by the path planner, based on a mixed integer linear formulation and an intelligent strategy for discretizing the environment, to iteratively find the best network-aware path to follow. The proposed approach is evaluated in various realistic simulation scenarios, showing the effectiveness of using the link quality map and the robustness to different restrictions regarding available information and computational resources.**

I. Introduction

We consider the general scenario in which a *mobile autonomous robot* $\mathcal{R}$ has to reach an assigned target location. The location could be, for instance, a place where the robot is expected to carry out some specific tasks, or an intermediate station where to take a further decision for action, or even the final destination where to rest and recharge the batteries for a while. It does not matter what is the specific purpose for reaching the target location, robot $\mathcal{R}$ aims to arrive as soon as possible and minimize energy expenditure. This implies moving along the *shortest path*, either in time or in distance. At the same time the robot might require to *communicate* with other robots in the surrounding areas, and/or with an external control center. For instance, communications could serve to implement coordinated behaviors for the multi-robot system as a whole, or could be used to let a human operator teleoperate or monitor the behavior of $\mathcal{R}$. In the most general case, these communications can be provisioned by the wireless *mobile ad hoc network* (MANET) formed by the robots, which can be used to collectively forward data in a multi-hop manner.

The simple but at the same time quite general scenario considered above, points out that an autonomous mobile robot traveling towards a given destination has to address both *navigation* and *communication* issues in order to effectively *plan* its path. In this work we tackle this problem of planning the path to an assigned destination, aiming to provide a solution that optimally balances robot's traveled distance and the ability to effectively communicate along the path with other mobile robots and/or with a control center using the wireless ad hoc network formed by the whole multi-robot system. In order to select such a *network-aware path*, the core issue becomes how to evaluate and predict the *communication quality* associated to each one of the possible feasible paths that $\mathcal{R}$ could follow to reach its destination. In terms of navigation, the quality is directly related to the *length* of the path (i.e., the traveling distance to the destination). Assuming that a common reference system is accessible to the robots (e.g., the GPS system), this length is immediately computable.

Deriving a measure of the expected communication quality (e.g., in terms of bandwidth or reliability) that a robot could experience while moving *along* a defined path in the environment is a difficult task. In fact, at each point along the path, the communication quality provided by the robot MANET depends on a number of factors, related both to the environment (e.g., the presence of walls, reflections, temperature of the air, electromagnetic disturbances, etc.), and to the local deployment and transmission activities of the other robots. In the literature, a common approach to this kind of problems consists in relying on complex equations for modeling the radio environment and its propagation and interference phenomena, coupled with some online sampling for communication quality, and derive an estimate of what it could be expected at a certain point in the environment (e.g., see the exemplary work in [1]). Still, it would be very difficult to capture and model the effects of the concurrent transmissions and mobility of the other robots, which have a large impact on the communication quality in a MANET.

In this work, we tackle the problem of estimating the expected quality of the communications that would be experienced in a distant point by relying on the use of *machine learning*, rather than explicit mathematical modeling. This way of proceeding is based on the results of our previous works [2], [3], [4], where we developed LQL, a framework for building effective *link quality estimators* through *supervised learning*. In the framework, we first identify a set of (easily) measurable network features that jointly define the *local network configuration* of a wireless link and, in turn, have a major impact determining the quality of the link (examples of the used features include local topology and local traffic characteristics in the neighborhood of both sender and receiver nodes). The LQL framework includes a distributed protocol for the controlled gathering of network data. For a given link, these data associate the information about the local network configuration of the link to the measured value of its quality, expressed in terms of *packet reception ratio* (PRR). The pairs *(local network configuration, link quality)* are then used as training data to learn the *regression mapping* between the set of features identifying local configurations and links' PRRs. The final outcome of the whole process is a *link quality estimator*, that, once built, can be used *online* in the environment where the training data has been gathered: based on the online measure of the values of the selected network features, a robust prediction about the expected quality of the associated links can be issued at a very low computational cost. The set of network measurements can be gathered prior task execution [2] or during the course of a mission [3]. In both cases, once enough training data have been collected, the link quality estimator can be built and used for the rest of the time.

Once the estimator is learned, we can use it to *predict* the quality of the existing local links or, as we do in this paper, the quality of *prospective* wireless links. As a result, we can endow the robot with the ability to answer the question: "What will be the quality of the wireless links with my neighbors when I move to that given point in space?". That is, given that $\mathcal{R}$ can gather information about the relative positions and traffic activity of the other nodes in the region towards its target destination, it can build a *spatial map of communication quality*, that associates to each point in the space a value of expected link quality. We term this the *network reward*: the local provisioning of network capability in terms of both connectivity and bandwidth expressed through the prediction of the links' PRR. The *total reward* associated to a full path is computed in an additive way, as the cumulative reward that can be collected along the path.

In practice, using the spatial map the robot can perform *network-aware path planning* over a long time/distance *horizon* (i.e., up to the destination): given that $\mathcal{R}$ is given a maximum distance it can travel, its planner has to compute the path that allows it to gather the maximal cumulative network rewards while satisfying the bounded distance constraint. This is equivalent to solve a single objective constrained optimization problem, which we model through a *mixed integer linear programming* (MIP) formulation (see Section V). The solution consists of a path, expressed as a *finite sequence of waypoints*, from the current location of the robot to its final destination point. The maximum distance budget is a parameter defined according to problem-specific strategic reasoning for limiting the consumption of robot's energy and time. It defines a bounded extra traveling length compared to the shortest path.

In order to deal with *dynamic environments*, where all deployed robots are potentially mobile, the calculation of the network-aware path is implemented as a *multi-stage* scheme using a *rolling horizon*. Online, while advancing towards the destination, $\mathcal{R}$ iteratively *replans* its path based on the newly gathered information about positions and traffic patterns of the other robots, that allows to issue new and up-to-date spatial predictions. In any case, predictions about points that are relatively far away from the current location might be not extremely accurate, due to the changes that will happen in the network before $\mathcal{R}$ could get there. To tackle this issue, we adopted an original approach based on the use of a *variable resolution* (termed, *adaptive grid*) to define the density of the feasible waypoints for path selection, setting a progressively decreasing density with the increase of the distance from the current location. In this way, we progressively decrease the impact of far away actions when calculating a plan. The adaptive grid strategy also allows to empirically control the *size* of the problem being solved in order to keep it bounded within acceptable limits for the computation time (e.g., for real-time applications and to favor iterative replanning).

Overall, the *contributions of the paper* consist in: (i) showing the advantages of learning and using spatial maps of communication quality for explicitly connecting path planning with network optimization in the same control model of a mobile robot; (ii) introducing a formal model for the optimization of both motion and communications which explicitly accounts for the presence of multiple mobile robots; (iii) proposing a variable resolution strategy for defining the structure of the problem to effectively address the computational challenges related to the dynamic and uncertain aspects of the scenario; (iv) reporting an extensive computational study, in simulation, that considers different application scenarios of practical interest for MANETs and addresses various limitations in terms of accessible information and computational resources,

The rest of the paper is organized as follows. Related work is discussed in Section II. The network aware path-planning problem is introduced in Section III. Our previous work on link quality learning and its novel use to derive spatial quality maps, is discussed in Section IV-A, while Section IV-B reports the definition of network rewards. The mathematical model for network-aware planning of a mobile robot is introduced in Section V. Results from simulation experiments considering different application scenarios are shown in Section VI, while Section VII summarizes the paper and outlines future work.

II. Related Work

A common requirement in multi-robot systems is that of maintaining or providing ad hoc communications [5]. Thus, the problem of planning and coordinating robot actions has to account for two potentially conflicting objectives. First, since application-related tasks have to be carried out at well defined locations in the environment (e.g., performing sensing tasks), a robot has to compute paths and navigate to the specified locations. However, in order to enable the formation of local network topologies that permit the required flows of information, the robot also needs to support creation, maintenance, and improvement of wireless links in order to enable data exchange. In practice, supporting wireless networking imposes constraints to the way robots can move throughout the environment. The challenges arising by the interplay between communication and mobility have been addressed in different domains such as search [6], [7], coordination and planning [8], [9], [1], surveillance [10], pursuit and evasion [11].

A common way to address the problem is through the dedicated use of a group of robots as *communication providers*, whose only objective is to enable data communications. These approaches include building and maintaining a communication infrastructure that enables the robots to communicate with any other robot in the team [12], or allows data exchange between two specific robots [13], [14]. In other works the robots simultaneously play the role of communication providers and task executors. To this end, the provisioning of communications and task planning are commonly considered as integrated issues, usually implying the enforcement of hard proximity constraints (e.g., to establish permanent communication paths between a base station and a robot team [15]). For the case of a single robot system, the co-optimization of motion and communications is considered in [1], focusing on adjusting the speed along an assigned path based on a formal model.

In this paper, we deal with the problem of fully planning the trajectory of a single mobile robot moving from one location to another, with the goal of supporting multi-hop wireless communications with other mobile robots. No restrictions are imposed to the other robots, while the planning robot has only to respect limits on the maximum traveling distance and/or time. This setting is much more general than those considered in the mention works and allows to fully exploit all robots for practical tasks rather than supporting communications.

In other domains, the purpose of the mobile robotic agent is to *collect* data generated by a number of static nodes (e.g., a sensor network) scattered over a defined area [16]. Typically, in these scenarios, data collection occurs when a static node is able to establish a wireless data link to one of the mobile, collector agents, also called *data mules*. Therefore, the problem becomes that of planning the trajectories of the data mules in order to *visit* and collect the data from all the static nodes and deliver these data to the end-user (e.g., a base station). Existing approaches commonly formulate this problem as a Vehicle Routing Problem (VRP), and make use of exact methods such as *mixed integer linear programming* (MIP) [17], [18], or computationally-efficient heuristic and approximation algorithms [19], [20]. Similarly, in this work, we formulate the trajectory planning problem as a variant of VRP, and make use of a MIP model to find optimal solutions. However, differently from the mentioned works, the problem we are dealing with is intrinsically dynamic, such that we propose techniques to deal with the inherent variations and uncertainties of the environment, and to compute real-time solutions using the MIP formulation in an iterative manner.

The existence and the quality of a wireless link depend on the interplay of a number of factors, related both to presence of concurrent wireless transmissions [21] and *hardware*, *software*, and *traffic generation* aspects (e.g., antenna characteristics, PHY and MAC layer protocols). Despite this well understood fact, most of the previous works have considered simplified communication models such as the *disk model* [7], [20]. More recently, some works have highlighted the importance of considering realistic communication models to control the trajectory of a mobile robot, based on probabilistic channel predictions [22] or using online estimators of wireless link capacity [23]. However, so far these approaches have been limited to the spatial predictions of the *single* wireless link between the mobile robot and a stationary base station. Moreover, the adopted models cannot really take into account the interference caused by the concurrent transmissions of nearby nodes. In this work, we use a machine learning approach for making spatial predictions of both existing and prospective links. Learning is based on a set of measurable network features that precisely account for the presence of *multiple* robots/nodes and implicitly capture the effects of complex radio phenomena (as supported by extensive realistic simulations and by real-world experiments carried out in our previous related works [3], [4]). Using the learned model, we build spatial maps of communication quality, that we use for making real-time spatial predictions and iteratively plan and adapt the path of the mobile robot accordingly, coping in this way with dynamics changes in multi-robot environments.

III. The Network-Aware Path Planning Problem

Given a multi-robot scenario where each robot is mobile and equipped with a wireless network interface, we consider the problem of planning the path for a specific robot $\mathcal{R}$ that has to travel from a starting location s to an ending location e. The objective is to find a path, expressed through a discrete sequence of waypoints, that maximizes a selected measure of *network performance* while keeping the traveled distance $\mathcal{D}$ within an assigned maximum limit value $\mathcal{D}_{max}$. $\mathcal{D}_{max}$ is strategically defined as an extra percentage of the shortest distance $\mathcal{D}_{min}$ for traveling from s to e (e.g., $\mathcal{D}_{max} = k\mathcal{D}_{min},\ k = 1.2$). The other robots are assumed to have their own motion plans, which do not adapt or change to favor $\mathcal{R}$.

For the sake of simplicity, we assume that motion happens on a *discretized* 2D plane. This means that the feasible paths between s and e are restricted to a numerable set $\mathcal{N}$ of candidate *waypoints* placed on a (non homogeneous) 2D grid covering the area between s and e. Without loss of generality, we consider a rectangular area, with s and e positioned at the opposite corners of the area. Furthermore, the movement of the robot between waypoints is restricted to a numerable set of *feasible transitions* $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ (directed arcs). The union of $\mathcal{N}$ and $\mathcal{E}$ constitute a directed *traversability graph* $G = (\mathcal{N}, \mathcal{E})$ that defines all the possible movements of the agent that can be considered when computing a solution path, which consists of a sequence of connected waypoints. The traversability graph accounts for the presence of (static) obstacles. We assume that the local navigation (and possible dynamic obstacle/agent avoidance) between waypoints is demanded to anyone of the many specialized algorithms available from the literature (e.g. [24]). It is important to remark that the discretization of the environment, that restricts the computation of a path to a selection among the feasible paths in the traversability graph, is a common way of proceeding. First of all, because the use of paths expressed as sequences of waypoints can accommodate for the *uncertainties* intrinsic to real-world scenarios, which would hardly justify the use of paths specified as continuous trajectories. Moreover, the environment discretization allows to reduce the *complexity* of the problem, and in our case enables the use of different levels of granularity to address computational constraints, which is shown in Section V-A.

We assume that some relative or global *positioning system* is in place, and that the robots *periodically broadcast short messages* including information about their position and expected traffic loads. These messages are spread throughout the ad hoc network by the robots themselves in a multi-hop manner, avoiding the uncontrolled proliferation of multiple copies and/or the transmission of out-of-date information. Based on these cooperative mechanisms, the planning robot $\mathcal{R}$ at any time can have access to information about expected traffic activity and current positions of the other robots (but not about their future positions).

This information is the basis to perform *network-aware path planning*: when planning the path of $\mathcal{R}$ from its current location to e, the path should be selected such that the traveled distance is minimized while at any point along the path communications with other robots can be effectively established in the robot MANET. In order to reach the objective, it is necessary to assign to each point of $\mathcal{R}$'s navigation space a sort of score, expressing how good communications would be at that (potentially distant) point. At this aim, based on the current deployment of all the agents in the system, and given the characteristics of their wireless network interfaces, to each waypoint i in $\mathcal{N}$ we associate a *networking reward* R_i. This indicates the expected *gain*, in terms of provisioning of networking capability, for passing through point i. At each point, the networking reward is derived from a *spatial map of link quality*, which, in turn, is built from the position and traffic load information spread by the robots. We discuss this process in Section IV. By default, both the start and the end locations have assigned zero reward: $R_s = R_e = 0$.

Given the starting and ending positions s and e, and an upper bound $\mathcal{D}_{max}$ on the maximum distance that the $\mathcal{R}$ is allowed to travel, the objective of $\mathcal{R}$'s planner is to define an elementary path p in G (i.e., a sequence of waypoints) from s to e, such that $|p| \leq \mathcal{D}_{max}$ and the total network reward collected along the path is maximized. The optimization problem is therefore set in a *lexicographic* way. First, an upper bound $\mathcal{D}_{max}$ on the maximum allowed traveling distance is derived from the calculation of the shortest path and of $\mathcal{D}_{min}$, the associated minimum traveling distance between s and e. The shortest path calculation is performed directly on the traversability graph, based on the map of the environment. Second, the $\mathcal{D}_{max}$ is used as a bound for the problem of defining a path that maximizes the collected network reward.

We envisage the situation in which the path planning process is *periodically iterated*, with each *stage* corresponding to a different starting location: $\mathcal{R}$ calculates its path to the destination from its starting position, executes it for a certain time duration, and then calculates new path starting from the new current position. The process is iterated over time until the final destination e is reached.

IV. From Spatial Link Quality Predictions to Network Rewards

In order to define network rewards, we first need to be able to quantify the *quality of a link*, that is, how good or bad a link is for communication. The presence of links of good quality is a core requirement for the proper functioning of a MANET. Therefore, our goal is to predict the value of the quality of the links that would be established along the path of robot $\mathcal{R}$, and employ these predictions to plan $\mathcal{R}$'s trajectory aiming to optimize both communications and distances. In the following, we first discuss the LQL framework (developed in previous work) which we use to learn link qualities, and then we show how we use it to operatively define network rewards.

A. LQL framework for spatial link quality predictions

In a MANET, the quality of a wireless link depends on several factors, such as *nodes' deployment* and the complex relations and interplay that they have with *hardware*, *software*, *traffic generation*, and *environmental* aspects. In order to effectively capture their combined effects in previous work [2], [3], we proposed LQL, a *supervised learning* framework to learn the mapping between some of these factors and the expected link quality, adopting as *link quality metric* the expected PRR (i.e., the $[0, 1]$ ratio between received and sent data packets). The framework allows to predict the quality of a wireless link on the basis of its *local network configuration*. Based on the literature and our experience, we represented this configuration by the following vector of features that are relatively easy to measure and play a major role determining the quality of a link: the *distance* between the two end-points of the link, and the number, relative positioning, and traffic characteristics of the *neighbor robots* (two robots are neighbors, and share a link, if they are separated by a distance less or equal to the transmission range of the network). Figure 1 illustrates the concept of local network configuration.

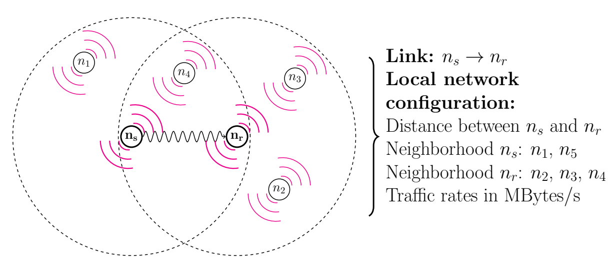


Fig. 1: Local network configuration of a link ($n_s \rightarrow n_r$). The neighborhood of the two end-points n_s, n_r of the link are described in terms of the relative positions of the surrounding robots and are depicted as dotted circles.

In order to build and use the link quality mapping, the first step prescribed by the LQL framework consists of using the robots/nodes of the network to collect a set of labeled link quality samples (i.e., pairs composed of a feature vector describing the local network configuration of a link and the corresponding PRR value). To this end, either *offline* or *online data gathering* procedures can be adopted. In the offline one [2], a group of mobile robots is deployed in the field prior the task operation of the network. Robots move in a controlled way, trying to maximize the number and the diversity of the observed local network topologies. At the same time, robots generate probing messages at variable rates. Each robot measures the PRR of the probing packets together with the values of the corresponding features describing the local network configuration of all links to its neighbor robots. Instead, in online data gathering [3], the mapping is learned during the normal course of operations of the network. All nodes passively monitor incoming and outgoing network traffic, and exchange the minimal amount of information that is required to assemble feature vectors and compute their corresponding PRR. Over time, each node incrementally records a set of link quality samples which is used for training. In this work we consider the offline procedure, assuming that an initial data gathering phase is executed before starting the main task of the system, or that the samples are already available from past operations. Note that the samples can be collected using any number of devices, at any time. Therefore, the same set of samples can be used in different situations and with a different number and type of robots, as long as the hardware/software parameters of the network interfaces remain the same (i.e., PHY-MAC protocols, transmission range, bandwidth).

After collected, the samples are used as training data to learn a link quality model in the form of a regression mapping from the space of the network features to PRR values. The effectiveness of the selected features and of the learning process has been validated in extensive experiments in simulation [2], sensor networks [3] and mobile robots [4] in various environments, both open space and cluttered ones, showing excellent accuracy and the ability to automatically capture the effects of complex radio propagation phenomena in the environment. Once trained, the model can be installed on robots and used to issue predictions for the expected PRR of a link, providing its local network configuration as input.

B. Network rewards

In this work, for network-aware path planning we make use of LQL's results to let $\mathcal{R}$ issue predictions of quality estimates for local and distant links, both existing and prospective ones. At this aim, in order to calculate the values of feature vectors, $\mathcal{R}$ requires information about relative positioning and envisaged traffic loads of other robots. As mentioned in Section III, this information is made available to $\mathcal{R}$ through local multi-hop communications in the robot MANET. Based on this information, for each waypoint i in the traversability graph to the target destination, $\mathcal{R}$ calculates the corresponding feature vector, which, in turn, allows to immediately derive the estimated PRR quality of all (incoming and outgoing) links at i using the learned link quality model.

The expected quality of communications that $\mathcal{R}$ would experience at point i (that can be quite distant from $\mathcal{R}$'s current location) depends on both the quality and the number of the individual (and possibly prospective) links that $\mathcal{R}$ could be able to establish in i. This notion is used to define the *network reward* R_i at point i, as the value that represents the *attractiveness* of that location in terms of communications. Quantitatively, the network reward can be defined in many different ways as a function of the number and the quality of the links at the location. In practice, it should be related to the communication performance goals of the specific application scenario. We propose a function that at any point i seeks for a balance between quality and number of connections:

$$R_i = \alpha LQ_i + (1-\alpha)Conn_i(l), \tag{1}$$

where $\alpha \in [0,1]$, LQ_i indicates the average of the predicted PRR values of all links in i, and $Conn_i$ denotes the connectivity component, related to the number l of links at point i. Both LQ_i and $Conn_i$ take values in $[0,1]$. We use the following form to define $Conn_i(l)$:

$$Conn_i(l) = \begin{cases} \dfrac{l}{max_l} & l \leq max_l \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where max_l is a parameter controlling the value of each additional connection. In the experiments we set $max_l = 6$.

V. MIP for Network-Aware Path Planning

We formulate the network-aware path planning problem as an instance of an *Orienteering Problem* (OP) [25]. An OP is specified through a graph where each vertex is associated to a score, and the edges define the adjacency relations between the vertices. The goal is to determine a path in the graph such that the total score collected along the path is maximized. In our case, vertices correspond to waypoints and the score is the networking reward obtained for passing by a waypoint.

The MIP *decision variables* are the following:

x_{ij}: binary, equals 1 if arc $(i,j) \in \mathcal{E}$ belongs to the path;

y_i: binary, equals 1 if way-point $i \in \mathcal{N}$ belongs to the path.

$$\text{maximize} \sum_{i\in\mathcal{N}} R_i y_i \tag{3}$$

subject to

$$y_s = y_e = x_{es} = 1 \tag{4}$$

$$\sum_{(i,j)\in\mathcal{E}} x_{ij} = \sum_{(j,i)\in\mathcal{E}} x_{ji} = y_i \quad i \in \mathcal{N} \tag{5}$$

$$t_i - t_j + 1 \leq (|\mathcal{N}|-1)(1-x_{ij}) \quad (i,j)\in\mathcal{E},\ i,j \notin \{s,e\} \tag{6}$$

$$1 \leq t_i \leq |\mathcal{N}| \quad i \in \mathcal{N} \tag{7}$$

$$\sum_{(i,j)\in\mathcal{E}} x_{ij} d_{ij} \leq \mathcal{D}_{max} \tag{8}$$

$$x_{ij},\ y_i \in \{0,1\} \quad i,j \in \mathcal{N} \tag{9}$$

Fig. 2: MIP formulation of path planning.

The MIP formulation for the path planning problem is shown in Figure 2. It refers to one generic planning stage. Constraints (4) ensure that paths start and end at the selected initial and ending points, where the initial point changes over the planning stages. Path continuity is guaranteed by constraints (5). Constraints (6)-(7) eliminate sub-tours [25]. Constraint (8) restricts the total distance that can be traveled.

Finally, constraints (9) set the binary requirements on the model variables.

A. Variable resolution for defining the traversability graph

The defined MIP builds on the traversability graph, that sets the search space of the problem. In practice, in order to build the traversability graph we assume that a detailed map of the environment is available to the planner. In fact, the waypoints are defined on the basis of the locations accessible to $\mathcal{R}$, while the transitions between waypoints depend both on the environment, and on the geometry and dynamics of the robot. To define the traversability graph, below we introduce an original method that achieves two objectives: minimize the computational load, and account for the dynamic characteristics of the problem at hand.

The traversability graph G defines the feasible waypoints and feasible waypoint transitions that are considered for $\mathcal{R}$'s path planning. In principle, the larger is the number of waypoints (i.e., the discretization of the environment is fine grained), the higher is the spatial accuracy in planning. However, the *computational complexity* of the problem is mainly affected by the size, in terms of cardinality $|\mathcal{E}|$ of the set of the directed arcs, and by the structure of the traversability graph G. In practice, if $\mathcal{R}$ is a robot with limited computational capabilities, the size of the problem should be defined according to these limits, to allow the rapid computation of the plans and make it possible to iterate the computations over different planning stages. In Section VI-B we perform a computational study precisely to derive practical estimates for the computational effort vs. problem's size. From this study we can roughly derive what should be the value of $\max_{\mathcal{E}}$, representing the maximum possible size for the edge set $\mathcal{E}$ that makes the computation staying within reasonable limits. Given $\max_{\mathcal{E}}$, and defined δ^+ as a parameter indicating the maximum number of outgoing arcs that any node $i \in \mathcal{N}$ can have (which impacts onto the structural complexity of the graph) and which is also, by construction, the most likely number of outgoing arcs of a node, we can also derive an estimate of the maximum allowed cardinality for $\mathcal{N}$, as $\max_{\mathcal{N}} = \max_{\mathcal{E}} / \delta^+$. Imposing the limits $\max_{\mathcal{N}}$ and $\max_{\mathcal{E}}$ it ensures that the resulting planning problem is computationally light/affordable for the robot.

In order to deal with the *dynamic* aspects of the multi-robot scenario, we must take into account that the spatial predictions of far away points are subject to a potentially large error due to the mobility of the robot. In planning, a common way to deal with this aspect consists in the use of a *discount factor*, weighting less the rewards expected in the far future. However, the application of a time-dependent discount factor in the formulation of the objective function would result into an explosion of the size of the traversability graph, whose number of nodes (and arcs) would be multiplied by the number of time steps in the planning horizon. Therefore, in order to achieve a discounting over the collected rewards while maintaining low the computational requirements, we make use of a variable resolution strategy (*adaptive grid strategy* henceforth), in which the density of the waypoints (and, accordingly, of the edges), diminishes with the increasing of the distance from the current starting point. Since the network reward is additively incremented for each traversed edge, when planning the increased density automatically results in giving more weight to the decisions about the initial path segments compared to the (sparse) decisions related to the final part of the path. The rationale is that it does not make much sense to perform a fine-grained planning in areas that will be reached at a time $t \gg t_{current}$. At the current time $t_{current}$ when the plan is issued, the positions of the robots in those areas that will be reached at time t are very uncertain. This is due to the mobility of the robots, as well as to the fact that the planner does not known about their future movements. Since the positions of all robots have an impact for building the spatial maps and, therefore, on network rewards, the reward itself becomes uncertain in the long horizon.

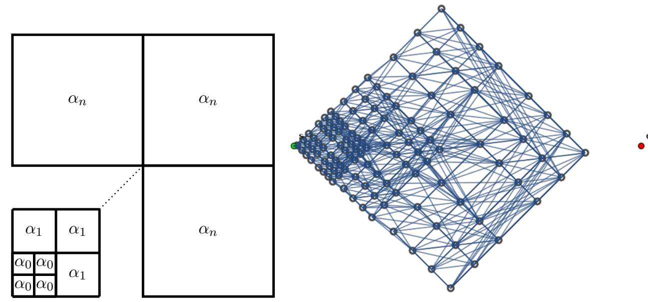


Fig. 3: (Left) Illustration of the adaptive grid strategy. (Right) Example of resulting traversability graphs. The arcs ending at e have been omitted.

In order to construct the traversability graph taking into account for bounds $\max_{\mathcal{E}}, \max_{\mathcal{N}}$, and δ^+, and implementing the variable resolution strategy, the following methods are used to define waypoints and edges.

1) Waypoints: The squared area defined by s and e is first uniformly divided into four squares of equal area. The sub-area containing the point s is further divided into four. The recursive process is repeated n times, finally obtaining $3\,(n-1)+4$ sub-areas. After this process, areas of equivalent size will contain the same number of waypoints. Figure 3 shows an example of construction, where α_i indicates the number of waypoints that will be generated inside a sub-area, which will be calculated depending on the value $\max_{\mathcal{N}}$.

2) Edges / Transitions: Having defined the set of waypoints, the issue becomes that of establishing the possible transitions between them. This translates into selecting $\max_{\mathcal{E}}$ arcs from the set $\mathcal{N} \times \mathcal{N}$, where each node $i \in \mathcal{N} \setminus \{e\}$ has an out-degree less or equal to δ^+. In the general case of open space, we propose a simple procedure to construct the set $\mathcal{E}$, that we used for the experiments.[1] For each node i, we select the closest δ^+ nodes that represent a movement towards the destination. Given two waypoints i and j, the arc

[1]In case of cluttered environments an analogous but more elaborate procedure can be followed, taking into account also robot's geometry and dynamics.

$i \rightarrow j$ represents a movement towards the destination if the angle between the vectors $\vec{ij}$ and $\vec{se}$ falls inside the interval $[-90°, 90°]$. Additionally, all directed arcs from $i \rightarrow e$, $\forall\, i \in \mathcal{N} \setminus \{e\}$, are also included in the set $\mathcal{E}$, and are not taken into account regarding the out-degree limit δ^+ since directly bring to the destination. In the experiments we set $\delta^+ = 8$.

Fig. 4 shows an example of the possible transitions considered in the traversability graph. In the example, the arc defined from i to j may be included in the graph since the angle between $\vec{se}$ and $\vec{ij}$ (γ_{ij} in the figure) is within the interval $[-90°, 90°]$, while the arc $i \rightarrow k$ (describing angle γ_{ik}) is forbidden.

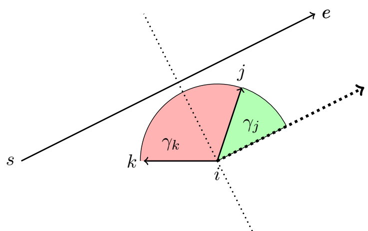


Fig. 4: Traversability graph: only movements describing a deviation less or equal to 90° with respect to the straight line joining s and e are allowed (e.g, the γ_j sector).

B. *Iterative path replanning*

In addition to the adaptive grid strategy, in order to effectively deal with the challenges of dynamic scenarios, the uncertainties of the environment, and the lack of complete information, we make use of an *iterative replanning strategy*, in which the planning process is periodically iterated. Each planning *stage* corresponds to a different starting location, from which the robot calculates its path to the destination considering the current deployment of other mobile agents, and the derived spatial prediction map. The robot then executes the path for a certain time, and recalculates a new path starting from the new current position. The process is iterated over time until the destination is reached.

Figure 5 illustrates the iterative replanning procedure. A robot (depicted as a triangle) moves with speed $2m/s$, traveling from $(0, 0)$ (leftmost bottom point) to $(600, 600)$ (rightmost top point), with $\mathcal{D}_{max} = 1.5\mathcal{D}_{min}$. The robot computes an initial trajectory at time 0, and performs replanning at time steps $t = 120s, 240s, 360s$. Each figure shows the spatial prediction map together with the deployment of other mobile robots (depicted as circles), at each replanning stage. It is possible to appreciate how the trajectory is adapted at each stage, based on the updated link quality map.

VI. Experimental Evaluation

A. *Simulation scenario*

We use the *ns-3* network simulator considering the following configuration: 802.11a Wi-Fi networks, transmission rate of 6 Mbps, log-distance propagation loss model with default parameters (path loss exponent set to 3.0), which corresponds to a transmission range (tx_r) of roughly 120 m. Robots are deployed over an area of 600× 600 m^2. In the basic path planning scenario, the planning robot $\mathcal{R}$ starts at the top-left corner of the area and moves towards the destination at the bottom-right corner (with a constant speed depending on the scenario). All other robots, 30 in all the experiments, are initially randomly placed in the simulation area and they autonomously follow individual random mobility models. Each robot generates a constant bit rate traffic in the form of 1-hop broadcast transmissions, according to one of the three possible rates: 80 Kbps, 400 Kbps, or 1.5Mbps, (which correspond to three *traffic profiles*, "low, medium, high", randomly assigned at the beginning of the simulation). Packet size is set to 1000 bytes. By default, we run simulations until $\mathcal{R}$ reaches its destination.

B. *Computational study*

As a first evaluation step, we use the described general scenario, to perform a computational study to assess the computational requirements for solving the planning model and to estimate the maximal problem size that allows to obtain optimal solutions within specified time limits. In particular, we assume that a computation time of less than one second is a desirable target (e.g., for real-time applications).

We consider 20,000 problem instances using random time-varying $2D$ functions to mimic the temporal variations of spatial predictions. We use different sets of parameters to construct the traversability graphs. and performed re-planning every 10 moves. To compute solutions to the MIP model we use Gurobi Optimizer 5.0.2 solver, with default parameters on a 3.4 Ghz Intel Core i7 processor and 8GB RAM. In order to analyze the results, we considered the resources need by the solver to find the optimal solution to the MIP model. Figure 6 shows the computational load, in terms of *CPU time* and *memory* usage, versus the size of the traversability graph (number of nodes and arcs) for all the model instances solved.

Results, as expected, show an exponential increase of both metrics as the size of the graph increases. The y-axis is a logarithmic scale for both metrics. Results also indicate that the use of graphs with less than 2500 arcs enables the solver to compute optimal solutions in less than one second, and with minimal memory requirements. Therefore, in the following experiments, this is the value we set for parameter $\max_{\mathcal{E}}$.

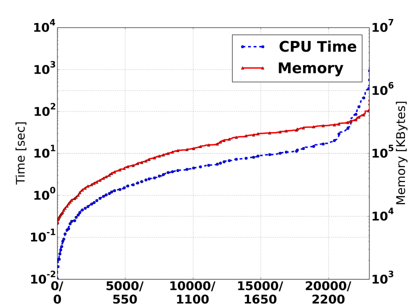
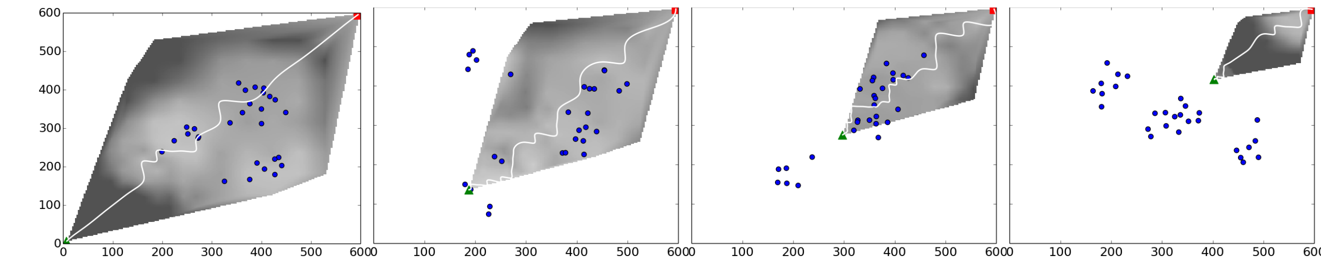


Fig. 6: CPU time and amount of memory required to solve the MIP over different sizes for the traversability graph.



(a) $t = 0$ (b) $t = 120$s (c) $t = 240$s (d) $t = 360$s

Fig. 5: Illustration of multi-stage replanning. A mobile robot (depicted as a triangle) travels across the area. At each stage, the deployment of the other robots (depicted as circles) determines the spatial prediction map used for planning. A dark gray color indicates a low reward. Light gray indicates a high reward. The computed trajectory at each step is indicated as a white line. White areas represent regions not covered by the traversability graph, and therefore not considered for planning.

C. *Maximization of local data exchange*

We evaluate the use of the adaptive grid strategy to tackle the dynamic nature of the multi-robot scenario. To this end, we consider the simple scenario where the randomly moving mobile robots periodically broadcast data packets according to their traffic profile. The path of robot $\mathcal{R}$ has to be adaptively planned in order to maximize the amount of data *received* from them. As network performance metric, we therefore consider the amount of data $\mathcal{R}$ could receive along its path.

In each experiment set we performed 100 runs, each time using different random trajectories for the data generating robots. We considered two strategies to place the set of possible waypoints in the traversability graph. The first, called *Uniform*, defines the waypoints using a uniform grid, where the points are uniformly spaced. The second, called *Adaptive*, defines the waypoints with varying density as explained in Section V-A. The results obtained by the use of the two strategies are shown in the box plots of Figure 7 considering different speeds (same for all the robots, including $\mathcal{R}$), given that speed has a major impact making dynamic the scenario.

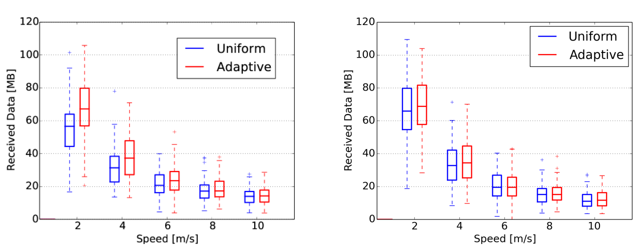


Fig. 7: The amount of data received by robot $\mathcal{R}$ vs. speed using different strategies for placing the waypoints. Re-planning intervals of 10 seconds (left) and 60 seconds (right) were used.

The results show that using the adaptive grid provides better performance (in terms of amount of received data) in comparison to the uniform case. However, the difference is less significant in very dynamic scenarios (i.e., with higher speeds). Nevertheless, it can be seen that combining the adaptive grid with short re-planning intervals it is effective to cope with dynamic scenarios, while the decrease of re-planning intervals on a uniform grid does not produce significant improvements.

As pointed out in section IV-A, $\mathcal{R}$ requires information about position and data rate of the other robots in order to build the link quality map. Therefore, here we analyze the impact of the *amount of information available to* $\mathcal{R}$ about the other robots. We consider a general robot speed of 2 m/s and a re-planning interval of 30 seconds. We restrict the amount of available information to by only considering the subset of robots that are located within a certain distance from $\mathcal{R}$. In particular, we considered distances of $k \cdot tx_r$, for $k = 1, \ldots, k$. This simulates the scenario where a simple controlled flooding mechanism allows to reliably propagate information, within the $k-$hop neighborhood. The results in Figure 8 (left) show that the difference in performance between the 1-hop and k-hop cases, with $k > 1$, is small, at least in terms of the median value, while in terms of range of variability the 1-hop case seems to be more subject to large variations in performance. However, overall, the results show that the use of the adaptive grid, together with the replanning strategy, allows to exploit at the best even the most restricted amount of information.

In the last set of experiments we evaluate the two grid strategies under a another restriction, this time in terms of the *allowed size for the traversability graph*. We consider a general robot speed of 2 m/s and a re-planning interval of 30 s. Results in Figure 8 (right) indicate that the adaptive strategy in practice does not suffer from reducing graph size (i.e., reducing planning resolution, increasing computational speed), while this has a negative impact on the uniform strategy.

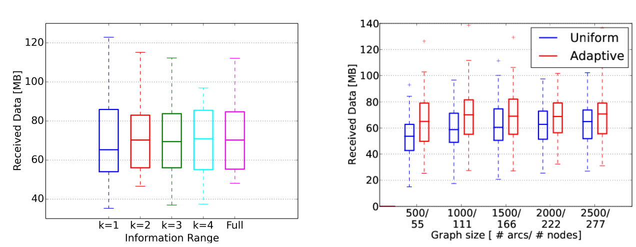


Fig. 8: (Left) Performance of computed solutions with incomplete information. The robot only uses the information about other robots within distance $k \cdot tx_r$. 'Full' means information about all robots. (Right) Comparison between adaptive and uniform grid strategies under different limits for graph size.

D. Multi-hop data session in irregular networks

In the previous experiments, we have shown that our approach can improve the local data exchange with other robots in the network. In this section, we demonstrate how this can directly improve the quality of a multi-hop communication session between an agent and a *base station (BS)*. We consider

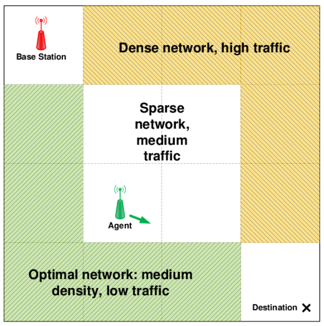


Fig. 9: Scenario for $\mathcal{R}$ moving through an irregular network.

the following scenario, which is an extension of the basic one from the previous experiments. Robot $\mathcal{R}$ still has to move towards the destination, but this time it also needs to constantly *send data back to the base station*. The base station (BS) is static and located at $\mathcal{R}$'s starting position. This scenario could represent a teleoperation situation, in which a frequent data exchange between the robot and a control center is required.

In order to emphasize the usefulness of the path planning approach, we consider for the other robots a spatial distribution that results in a network topology which is *irregular in terms of connectivity* (see Figure 9), namely: (a) the middle part of the network is very sparse (implying low connectivity), (b) upper and right parts are dense (high connectivity), but also the network load is very high there (which determines a lower quality of the wireless links), and (c) lower and left parts provide optimal communication conditions: they are well connected, but not very dense and with low traffic. All networking robots implement random waypoint mobility (speed: 1.5 m/s, pause time: 5 secs), but only within their local areas (the network is divided into 16 cells as in Figure 9). $\mathcal{R}$ moves with speed of 2 m/s. For data routing we use our *AntHocNet* algorithm [26].

The choice for such an irregular setting is twofold: some initial experiments showed that in perfectly uniform networks (uniform connectivity and traffic profiles) on average the trajectory does not affect the quality of local communications; most of the real applications naturally lead to a certain level of irregularity determined by the requirements of the performed tasks. In particular, the irregularity in the proposed scenario emphasizes the situation where the trajectory should be carefully chosen in order to achieve a successful communication.

In the experiments, we compare three approaches: (i) following the shortest path (i.e., a straight line in our scenarios), which we refer to as SP, (ii) path planning based on maximizing the number of local connections (CONN) and (iii) our approach based on spatial link quality predictions (LQ). In particular, the approach CONN can be considered as a representative approach of many other works in the literature, in which the local decision-making aiming to optimize the network performance is based exclusively on the number of neighbors, ignoring the quality of the resulting connections (e.g., ignoring the negative effects of interference). That is, the simple behavior dictated by CONN is to locally move towards highly populated areas.

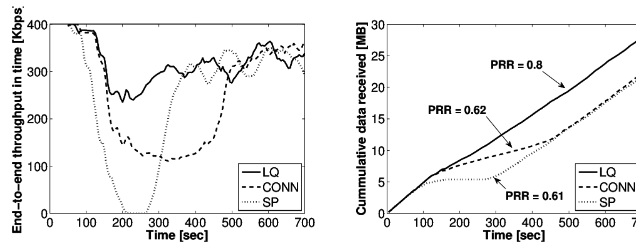


Fig. 10: The amount of data exchanged with the BS. (Left) End-to-end throughput averaged over a moving window. (Right) The cumulative amount of received data.

First we analyze the amount of data exchanged with the BS during the experiment. We measure the end-to-end throughput (Figure 10, left) between the robot and the BS, averaged in a moving window (window size 5 sec.). We also show the cumulative amount of data received in the BS from the beginning of the experiment (Figure 10, right), and the final *average PRR*. All results are averaged over 10 simulations (700s each).

We can observe that choosing our LQ approach provides the highest throughput and ensures continuous connectivity with the BS. Choosing the shortest path implies going through the sparse area of the network, and consequently loosing connectivity with the BS. At the same time, simply moving towards the dense and highly connected areas is not enough, as traffic congestion may lead to frequent packet losses, such that obtained throughput is significantly lower compared to LQ (although the connectivity with the BS is usually preserved). The advantage of using the shortest path is that the destination is reached earlier (after approximately 425 seconds), and than the connectivity is reestablished. In the case of the explicitly controlled trajectories (LQ and CONN), the destination is reached after approximately 550 seconds. This is particularly disadvantageous for the CONN approach, as the agent spends more time in the congested areas of the network. As a result, the total amount of data received at the BS at the end of the simulation is comparable for both SP and CONN approaches. Nevertheless, the results show the advantages of spatially predicting the quality of wireless links and moving towards the areas of expected good communication quality.

In order to better support our claims, we also analyze the distribution of the average end-to-end packet delay (Figure 11, left) and of the average packet delay jitter (Figure 11, right). In both cases, we consider values averaged in a moving time window of 5 seconds over all simulation runs (thus, each box plot represents 1400 measurements, 140 measurements

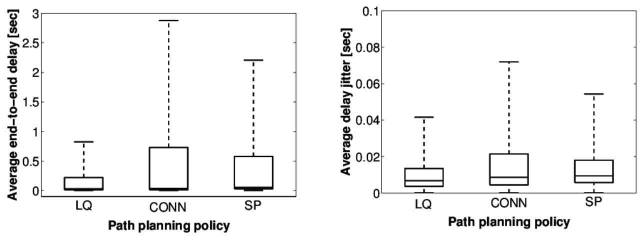


Fig. 11: (Left) Distribution of the average end-to-end packet delay. (Right) Distribution of the average packet delay jitter.

for each 700s simulation run). The results show another important advantage of following good quality links: our approach achieves significantly lower delays, and delays are also more stable in time according to the time jitter. This could be beneficial in applications such as video or voice streaming.

VII. Conclusions and Future work

The paper tackled the problem of performing network-aware online path planning for a mobile robot $\mathcal{R}$ moving towards an assigned location and sharing the environment with other autonomous mobile robots. Altogether the robots form a multi-hop mobile ad hoc network. $\mathcal{R}$'s planner exploits both information gathered online regarding other robots' positions and traffic loads, and a machine-learning based link quality estimator developed in previous work, to build spatial maps of communication quality. The maps are used at planning time to score spatial locations in terms of provisioning of networking and find the path that optimally balances traveled distance and expected quality of communications along the path. The other robots cooperate by sharing information but do not change their behavior to ease $\mathcal{R}$'s task. A mixed integer formulation is adopted to model and solve the planning problem, and a variable resolution approach is adopted to discretize $\mathcal{R}$'s navigation space and cope with dynamic and uncertain issues.

The effectiveness of the network-aware planner has been demonstrated through a set of simulation experiments under various realistic scenarios of one and multi-hop communication and routing, and studying the impact of speed, other robots' information, and computational resources.

Ongoing work focuses on extending the model to include collision avoidance and management of information about other robots' mobility [27], and on the implementation on a swarm of mobile robots all employing the same planner.

Acknowledgments

This research has been partially funded by the Swiss National Science Foundation (SNSF) Sinergia project SWARMIX, project number CRSI22_133059.

References

[1] Y. Yuan and Y. Mostofi, "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1562–1572, 2013.

[2] E. Feo Flushing, J. Nagi, and G. Di Caro, "A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks," in *Proc. of the International Conference on Computing, Networking and Communications (ICNC)*, 2012.

[3] G. A. Di Caro, M. Kudelski, E. Feo Flushing, J. Nagi, I. Ahmed, and L. Gambardella, "On-line supervised learning of link quality estimates in wireless networks," in *Proc. of Med-Hoc-Net*, 2013, pp. 69–76.

[4] M. Kudelski, L. Gambardella, and G. A. Di Caro, "A mobility-controlled link quality learning protocol for multi-robot coordination tasks," in *Proc. of IEEE ICRA*, 2014 (to be published).

[5] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 111–131, 2008.

[6] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, 2007.

[7] Y. Pei, M. W. Mutka, and N. Xi, "Connectivity and bandwidth-aware real-time exploration in mobile robot networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 9, pp. 847–863, 2011.

[8] G. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, 2012.

[9] D. Tardioli, A. Mosteo, L. Riazuelo, J. Villarroel, and L. Montano, "Enforcing network connectivity in robot team missions," *Int. Journal of Robotics Research*, vol. 29, no. 4, pp. 460–480, 2010.

[10] A. Ghaffarkhah and Y. Mostofi, "Path planning for networked robotic surveillance," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3560–3575, 2012.

[11] J. Thunberg and P. Ögren, "A mixed integer linear programming approach to pursuit evasion problems with optional connectivity constraints," *Autonomous Robots*, vol. 31, no. 4, pp. 333–343, 2011.

[12] S. Gil, D. Feldman, and D. Rus, "Communication coverage for independently moving robots," in *Proc. of IEEE IROS*, 2012, pp. 4865–4872.

[13] Y. Yan and Y. Mostofi, "Robotic router formation in realistic communication environments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 810–827, 2012.

[14] E. I. Grøtli and T. A. Johansen, "Path planning for UAVs under communication constraints using SPLAT! and MILP," *Journal of Intelligent Robotics Systems*, vol. 65, no. 1-4, pp. 265–282, 2011.

[15] J. Fink, A. Ribeiro, and V. Kumar, "Robust control of mobility and communications in autonomous robot teams," *IEEE Access*, vol. 1, pp. 290–309, 2013.

[16] E. Ekici, Y. Gu, and D. Bozdag, "Mobility-based communication in wireless sensor networks," *IEEE Communications*, vol. 44, no. 7, pp. 56–62, 2006.

[17] Y. Yan and Y. Mostofi, "Communication and path planning strategies of a robotic coverage operation," in *Proc. of the American Control Conference (ACC), 2013*, 2013, pp. 860–866.

[18] R. Sugihara and R. K. Gupta, "Path planning of data mules in sensor networks," *ACM Trans. on Sensor Netw.*, vol. 8, no. 1, pp. 1–27, 2011.

[19] O. Tekdas, V. Isler, J. H. Lim, and A. Terzis, "Using mobile robots to harvest data from sensor fields," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 22–28, 2009.

[20] D. Bhadauria, O. Tekdas, and V. Isler, "Robotic data mules for collecting data over sparse sensor fields," *Journal of Field Robotics*, vol. 28, no. 3, pp. 388–404, 2011.

[21] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[22] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 964–978, 2012.

[23] M. Lindhe and K. Johansson, "Adaptive exploitation of multipath fading for mobile sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1934–1939.

[24] J. Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Springer, 2011, vol. 70, pp. 3–19.

[25] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[26] G. Di Caro, F. Ducatelle, and L. Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.

[27] G. A. Di Caro, E. Feo Flushing, and L. Gambardella, "Use of time-dependent spatial maps of communication quality for network-aware multi-robot path planning," in *Proc. of the 8th Intl. Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN)*, 2014.