

Distributed Scheduling of Network Connectivity Using Mobile Access Point Robots

Nikolaos Chatzipanagiotis, *Student Member, IEEE*, and Michael M. Zavlanos, *Member, IEEE*

Abstract—In this paper we consider scenarios where mobility can be exploited to enable reliable communications in wireless networks with scarce resources that are unable to concurrently service their nodes. Specifically, we consider cases where a team of robots operate as mobile Access Points (APs) that provide service, namely sufficient end-to-end communication routes, to a multi-hop network of static source nodes which generate data. We introduce the *connectivity scheduling* problem, a novel framework that combines motion planning of the APs with service scheduling of the source nodes and network routing control so that integrity of communications is guaranteed over time. We formulate the connectivity scheduling problem as a multi-stage Mixed Integer Programming (MIP) problem, where path planning, service scheduling, and routing decisions are all jointly optimized over a discrete-time horizon. Since MIP problems can grow intractable quickly, we further consider a continuous convex reformulation of the problem and employ sparse optimization techniques, specifically the re-weighted ℓ_1 regularization scheme, to recover the desired integrality structure of the solution. We propose a decentralized method to solve the above relaxation that is based on the recently developed Accelerated Distributed Augmented Lagrangians (ADAL) algorithm. Specifically, we modify ADAL by incorporating in the algorithm the re-weighted ℓ_1 scheme, which enables us to recover the desired sparsity structure of the original MIP at the final solution. Numerical results are presented that validate the effectiveness of the proposed framework.

Index Terms—Robotic teams, wireless communications, network optimization, convex optimization, distributed algorithms.

I. INTRODUCTION

WIRELESS networks are ubiquitous nowadays, since they can provide good coverage and service at low costs. They typically involve multiple agents positioned in geographically disparate locations that communicate in a multi-hop fashion, i.e., information reaches its destination following a path of intermediate nodes that are selected based on their ability to relay data. Due to the nature of wireless transmissions, the reliability of such communications is significantly affected by the availability of resources, as well as environmental factors. In this paper we propose the use of mobility as an effective way to compensate for scarce resources and environmental factors and improve on the reliability and efficiency of wireless communications.

Early approaches that consider mobility in multi-hop wireless networks employed proximity graphs and studied the problem of preserving graph connectivity while the wireless

agents move. Such approaches involve, for example, maximization of the algebraic connectivity of the graph [1, 2], potential fields that model loss of connectivity as an obstacle in the free space [3], and distributed hybrid approaches that decompose control of the discrete graph from continuous motion of the robots [4]. Distributed algorithms for graph connectivity maintenance have also been implemented in [5, 6]. A comprehensive survey of this literature can be found in [7].

The advantage of using graph-based approaches to control the communications in networks that contain mobile agents lies in their simplicity. However, it is this simplicity that also limits their practical applicability, since graph-based methods cannot fully capture the highly complex nature of wireless transmissions. Moreover, graph-based methods do not account for the actual routing decisions of the nodes. The problem of establishing reliable routing schemes for mobile ad-hoc networks has been studied extensively; see [8]–[13] for some indicative examples. However, these works consider mobility essentially as an external parameter that can cause changes to the topology of the communication network, and they focus on designing appropriate routing schemes that take into account this time-varying nature of the problem. In other words, they consider mobility to be unpredictable, or predictable but uncontrollable.

To address these shortcomings, a scheme combining mobility and routing control for teams of mobile robots was recently proposed in [14, 15]. In these works, the robots' trajectories are designed to maintain the connectivity of the network, using a more realistic communication model that takes into account requirements such as packet deliverability, queue stability, the randomness of wireless links, and satisfaction of desired end-to-end rates. Related methods for the control of wireless robot networks are proposed in [16] and [17, 18], where the wireless channels are modeled using path loss, shadowing, and multi-path fading, or evaluated using on-line learning techniques, respectively.

In this paper we propose a new viewpoint on the problem of designing reliable mobile communication networks. Specifically, we consider communication networks involving static wireless source nodes that generate data and assume that these networks do not possess the necessary resources, i.e., sufficient numbers of reliable end-to-end routes, in order to support the rates of information generated by the sources. In mathematical terms, this translates to situations where for given network configurations, the problem of determining optimal communication rates and routes is infeasible. To address this problem, we propose a new framework to allocate the available network resources to subsets of nodes over time,

This work is supported by NSF under grant CNS #1261828, and by ONR under grant #N000141410479.

Nikolaos Chatzipanagiotis and Michael M. Zavlanos are with the Dept. of Mechanical Engineering and Materials Science, Duke University, Durham, NC, 27708, USA, {n.chatzip,michael.zavlanos}@duke.edu.

so that end-to-end connectivity is ensured, not for all time, but over time for all subsets of nodes. We call this framework *connectivity scheduling*. Specifically, we consider teams of robotic data sinks, henceforth referred to as Access Points (APs), that move in the area in which the network nodes are deployed and service those nodes that fall within their end-to-end connectivity range. Nodes that fall outside the APs' connectivity range or have been serviced before, either remain idle or act as relays, facilitating the service of new nodes. More formally, connectivity scheduling is the problem of determining AP trajectories and associated sequences of connected subnetworks whose union forms a connected network. Here we say that a subnetwork (resp., network) is connected if there exist end-to-end paths, determined by the existence of routing decisions, from a subset of (resp., all) sources to a set of APs. Also, we define the union of connected subnetworks by the union of end-to-end paths between the corresponding sources and APs. Therefore, compared to graph-theoretic formulations, "connectivity over time" here is not defined in terms of unions of links, but in terms of unions of end-to-end paths between sources and APs.

This work points to a new direction in the control of mobile communication networks. Unlike most recent literature that designs network topologies so that point-to-point or end-to-end connectivity is always guaranteed, i.e., so that feasible communication variables always exist, connectivity scheduling presents a more flexible approach that allows for temporary disconnections of nodes, while ensuring that all network nodes are reliably serviced over time. To the best of our knowledge, this is the first time that communication control in mobile networks is formulated as a connectivity scheduling problem. A relevant problem is considered in [19] where clustering methods are used to partition the set of nodes and determine associated clusterheads that an Unmanned Aerial Vehicle (UAV) can visit. Clustering respects energy, bit error rates (BER), and UAV flight time constraints, but the communication model in [19] does not guarantee end-to-end rates. Similarly, in [20] a heuristic algorithm is proposed to plan the path of a single UAV serving clustered nodes such that the power consumption of said nodes is minimized. Another pertinent work is [21] where the authors propose variants of the Traveling Salesman Problem (TSP) to plan the path of a single mobile sink node such that it maximizes data collection from a set of clustered sensors. The main differences between [20, 21] and the current paper are that we allow for multiple mobile data gathering nodes, while at the same time exploiting the benefits of multi-hop routing schemes.

We formulate the connectivity scheduling problem as a single multi-stage Mixed Integer Programming (MIP) problem [22], where path planning of the APs, service scheduling of the source nodes, and network routing decisions are all jointly optimized over a given discrete-time horizon. In the proposed MIP all the integer variables are binary, representing the status of the nodes, e.g., currently being serviced or not, and the presence of the mobile access points at given discrete locations in space. Since MIP problems are computationally challenging to solve, we further propose a convex relaxation of the problem where the integrality constraints are relaxed into

continuous ones. In order to recover the 0–1 structure for the, now relaxed, binary variables, we utilize sparse optimization techniques [23]–[25]. In particular, we employ the recently proposed re-weighted ℓ_1 regularization approach [23] that has proved to perform well in practice.

The proposed relaxation of the MIP problem is more efficient to solve, however, it still requires a central processing unit that gathers all the problem parameters, computes a solution, and then broadcasts the optimal decisions to the network nodes. Such centralized methods suffer from certain well known disadvantages in practice. Specifically, they do not scale well with the number of network nodes, can incur large communication costs, entail significant delays, are vulnerable to failures and changes in the problem parameters, and also give rise to privacy and security concerns. To address these issues, we propose a distributed solution, wherein the relaxed problem can be decomposed into smaller subproblems that are solved iteratively and in parallel at every node in the network. The proposed distributed approach is based on the Accelerated Distributed Augmented Lagrangians (ADAL) algorithm [26]–[28], that was recently developed by the authors. ADAL is based on the *augmented Lagrangian* (AL) framework, a regularization technique that is obtained by adding a quadratic penalty term to the ordinary Lagrangian of a problem [29, 30]. The AL framework has proven to be the most efficient and popular approach for distributed optimization recently, cf. [31] for a more detailed discussion. In comparison, *dual decomposition* [30], a standard technique used for distributed optimization, suffers from exceedingly slow convergence rates and requires strict convexity of the objective function. These drawbacks are alleviated by the AL framework due to the addition of the regularization terms [30].¹ It was shown in [26] that, for a number of different applications, ADAL exhibits a significant improvement in convergence speed compared to existing distributed AL techniques, such as the *Alternating Directions Method of Multipliers* (ADMM) [31, 42], and the *Diagonal Quadratic Approximation* (DQA) [43]. In this paper, we modify ADAL by incorporating into the algorithm the re-weighted ℓ_1 regularization heuristic. Numerical results are provided to verify that the algorithm converges to the desired solution, while also recovering the 0–1 structure for the, now relaxed, binary variables.

The rest of this paper is organized as follows. In section II we discuss the model that is used to capture the multi-hop wireless communications between nodes in the network. Then, in section III we present the proposed multi-stage planning approach for the mobile APs, and formulate the problem as a single MIP problem. In section IV we present a convex reformulation of the MIP problem, which entails relaxing the integrality constraints into continuous ones and employing the re-weighted ℓ_1 regularization scheme to recover the sought integrality structure of the solution. Moreover, we propose a distributed algorithm to solve this convex relaxation, which is based on combining ADAL with re-weighted ℓ_1 regularization.

¹Alternative, recently proposed algorithms for distributed optimization include Newton methods [32]–[34], projection-based approaches [35, 36], Nesterov-like algorithms [37, 38], online methods [39, 40], and even continuous-time approaches [41].

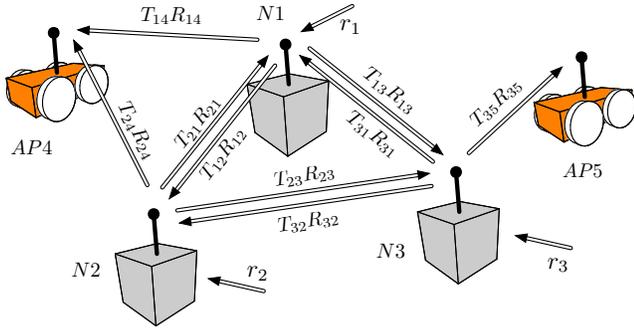


Fig. 1: Wireless network consisting of two APs and three source nodes. Shown are the packet rates r_j generated by every source as well as the rates $T_{jn}R_{jn}$ sent from source j and successfully decoded by node n , where T_{jn} is the probability that node j routes packets to node n , and R_{jn} is the reliability of the channel between nodes j and n .

Finally, in section V we present numerical results to verify the validity of the proposed approach.

II. THE COMMUNICATION MODEL

The wireless communication scheme employed in this paper is based on a multi-hop stochastic routing model, whereby each node selects a neighbor to forward a data packet according to a given probability distribution, cf. [14, 44]. All nodes are assumed to be stationary during the communication stage.

In particular, consider a wireless network consisting of J source nodes that need to route data towards a set of K Access Points (APs). We denote the set of source nodes by $\mathcal{J} = \{1, \dots, J\}$, the set of APs by $\mathcal{K} = \{J+1, \dots, J+K\}$, and the combined set containing all the nodes of the network by $\mathcal{N} = \mathcal{J} \cup \mathcal{K} = \{1, \dots, J+K\}$. We assume that the APs only receive and do not transmit any data.

Point-to-point connectivity is modeled by a rate function R_{jn} that determines the amount of information that is transmitted from node $j \in \mathcal{J}$ and is correctly decoded by node $n \in \mathcal{N}$. Specifically, denoting by R_0 the transmission rate of the nodes' radios and by $P_{jn} \in [0, 1]$ the probability of successful decoding over the link from j to n , the rates R_{jn} can be constructed as $R_{jn} = R_0 P_{jn}$. In what follows, for simplicity, we work with normalized rates by making $R_0 = 1$ that is the same for all nodes. Therefore, $R_{jn} \in [0, 1]$. We also assume that direct communication with the APs is not always possible, so the source nodes need to route data to the APs in a multi-hop fashion. Routing of packets is due to routing decisions T_{jn} that represent the fraction of time that node j selects node n as its intended destination. Note that, since the routing variables T_{jn} represent time slot shares they need to satisfy $0 \leq T_{jn} \leq 1$ and $\sum_{n \in \mathcal{N}} T_{jn} \leq 1$. Then, the products $R_{jn}T_{jn}$ denote the point-to-point rate of information that is transmitted by node j and is correctly received by node n .

Between their generation or arrival from another node and their transmission, packets are stored in a queue. The total rate at which packets arrive at the queue of node j is $\sum_{i \in \mathcal{J}} T_{ij}R_{ij}$. Similarly, the total rate at which packets leave the queue of node j is $\sum_{n \in \mathcal{N}} T_{jn}R_{jn}$. We can express the end-to-end

information rate r_j at node j as

$$r_j = \sum_{n \in \mathcal{N}} T_{jn}R_{jn} - \sum_{i \in \mathcal{J}} T_{ij}R_{ij}. \quad (1)$$

This is the rate at which node j can add data in the network. Integrity of the communication network requires that the end-to-end rates r_j exceed minimum thresholds $r_j^{\min} \geq 0$ that capture the rate of data that is directly generated at nodes, i.e., $r_j \geq r_j^{\min}$ for all $j \in \mathcal{J}$. This condition is necessary to ensure stability of the queues and deliverability of information packets to the APs. As before, we assume normalized minimum thresholds so that $r_j^{\min} \in [0, 1]$.

Introducing optimality criteria $f_j(r_j)$ and $g_{jn}(T_{jn})$ that measure the utility associated with the end-to-end rates r_j and routing decisions T_{jn} , respectively, the optimal operating points can then be selected as

$$\begin{aligned} \min_{\mathbf{r}, \mathbf{T}} \quad & \sum_{j \in \mathcal{J}} f_j(r_j) + \sum_{j \in \mathcal{J}} \sum_{n \in \mathcal{N}} g_{jn}(T_{jn}) \\ \text{s.t.} \quad & r_j = \sum_{n \in \mathcal{N}} T_{jn}R_{jn} - \sum_{i \in \mathcal{J}} T_{ij}R_{ij}, \quad \forall j \in \mathcal{J}, \\ & \sum_{n \in \mathcal{N}} T_{jn} \leq 1, \quad \forall j \in \mathcal{J}, \\ & 0 \leq T_{jn} \leq 1, \quad \forall j \in \mathcal{J}, n \in \mathcal{N}, \\ & r_j \geq r_j^{\min}, \quad \forall j \in \mathcal{J}. \end{aligned} \quad (2)$$

Here, we use $\mathbf{r} \in \mathbb{R}^J$, and $\mathbf{T} \in \mathbb{R}^{J(J+K)}$ to denote the vectors stacking all the end-to-end data rates r_j and the routing decisions T_{jn} , respectively.

The utility functions f_j and g_{jn} are assumed to be convex for all $j \in \mathcal{J}$ and $n \in \mathcal{N}$. Typically, the requirement on the utility functions $f_j(r_j)$ is that they are monotonically non-increasing expressing preference to larger transmission rates [44], i.e., an increase in the rate of one node does not increase the value of the total utility function to be minimized. Example utilities for (2) are linear $\sum_{i \in \mathcal{J}} f_j(r_j) = -\sum_{j \in \mathcal{J}} w_j r_j$ or logarithmic $\sum_{j \in \mathcal{J}} f_j(r_j) = -\sum_{j \in \mathcal{J}} \log(r_j)$. Linear utilities yield larger rates and favor nodes that are close to the APs. Logarithmic utilities yield fairer operating points because they penalize small rates. This choice is typical in Network Utility Maximization (NUM) problems [45]. Similar arguments hold for the g_{jn} functions, also. Note that formulations with non-convex functions are certainly possible as well; see [46] for more details on this subject.

III. MULTI-STAGE CONNECTIVITY SCHEDULING WITH MOBILE APs

In this section we consider situations where problem (2) is infeasible due to the lack of available resources, i.e., available end-to-end communication routes in the network, that can support the desired communication rates r_j^{\min} . We address this issue by introducing a discrete time horizon in problem (2) over which all source nodes can be serviced. Each time step of the horizon is a single communication stage as described in section II. The access points are allowed to move during this horizon in order to service the source nodes in a sequential manner. At each communication stage the source nodes can be in one of three states: i) active, wherein they generate

data according to their minimum rate requirements r_j^{min} , ii) operate only as relays, wherein they generate no data, but help relay the data from other sources, and iii) inactive, wherein they remain idle and do nothing for the duration of the communication stage.

The key idea that motivates this work is that even though the APs can not service the whole network at once, it is possible that they can still service parts of the network that are located in their vicinity so that, over time, as they move, they service the whole network. The challenge then is to determine a sequence of AP locations and associated subsets of nodes so that eventually all nodes are serviced over time. We call this problem *connectivity scheduling*, a multi-stage communication and mobility control framework that jointly determines appropriate AP trajectories and time-stamped communication routes so that end-to-end communication is guaranteed between all nodes and APs over time. The advantage of the proposed multi-stage framework is that it allows us to combine the multi-modal operation capabilities of the nodes (source, relay, inactive) with the mobility of APs in order to address the infeasibility of the original problem in a more efficient manner.

A simple example illustrating the need for mobility in connectivity scheduling is shown in Fig. 2, while a more elaborate example is shown in Fig. 3. This example showcases a scenario where the r_j^{min} are too large to be satisfied all at once for that specific network structure, cf. Fig. 3a. More specifically, the R_{jn} are such that the middle nodes have to set their routing decisions towards the AP to 1 and route only their own data directly to the AP. Hence, the top node cannot satisfy its $r_j^{min} = 0.9$ by using the middle nodes as relays, which renders the optimization problem (2) infeasible. To resolve this, we can introduce two communication stages and utilize a mobile AP. At the first stage, only the middle nodes are active and they route directly to the AP, cf. Fig. 3b, while the top node is inactive. At the second stage, the AP services only the top node, since the middle nodes have been serviced already. To achieve this, the AP moves appropriately so as to establish a stronger link with the rightmost middle node, which in turn allows the top node to relay its data successfully, while the left middle node remains inactive. Note that the key factor to render the problem feasible here is the mobility of the AP; if the AP did not move, then the two-stage formulation of the static problem would still be infeasible, since the channel reliabilities R_{jn} are not large enough to support a multi-hop routing scheme.

In what follows we present a Mixed Integer Programming (MIP) formulation to solve the connectivity scheduling problem over a horizon of T communication stages. In the proposed formulation, the integer variables are all binary. To facilitate the development of our method, we present the proposed connectivity scheduling problem as a combination of two problems: i) a discrete path planning problem for the available mobile APs, wherein we decide on the successive locations of every AP throughout the movement horizon T , and ii) a routing control problem, where at each stage we determine the source nodes that are active, inactive, or required to operate as relays. The aforementioned two parts of the problem are coupled by the fact that the routing decisions of

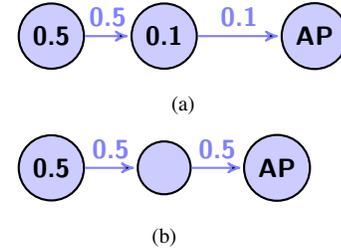


Fig. 2: A simple example that illustrates the benefits of mobile APs. The numbers inside the nodes indicate their corresponding r_j^{min} values, while the numbers next to the edges indicate their corresponding R_{jn} values. a) If the AP is not allowed to move, then the problem remains infeasible even if we introduce communication stages, since the link between the middle node and the AP is not strong enough to accommodate for the r_j^{min} of the left node by relaying, b) In this case, the middle node is active at the first stage, while the left node remains inactive. At the second stage, the AP moves to establish a stronger link with the middle node, thus enabling the top node to route at a rate of 0.5 using the middle node as relay.

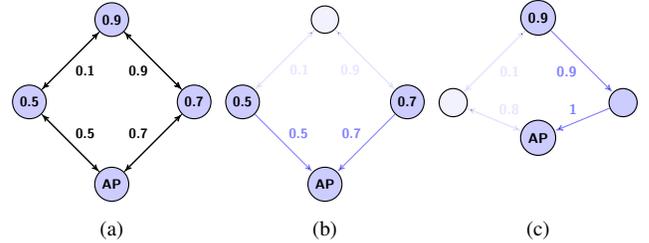


Fig. 3: A network with 3 source nodes and a single AP. The numbers inside the nodes indicate their corresponding r_j^{min} values, while the numbers next to the edges indicate their corresponding R_{jn} values. a) The 1-stage routing problem is infeasible since not all the r_j^{min} can be satisfied given the current network structure, b) At the first stage, only the middle nodes route directly to the AP, while the top node remains inactive, c) At the second stage, the AP moves closer to the rightmost middle node to establish a stronger link, thus allowing the top node to route its data using the rightmost middle node as a relay (the left middle node is inactive now). Note that, even if we were to half the r_j^{min} rates of the 1-stage static problem of Fig. (a) to account for the added routing time introduced by the second stage, then the problem would still be infeasible for this particular network structure. The key here is to exploit the mobility of the AP.

the source nodes depend on the locations of the mobile APs. The final formulation is a single MIP problem and is presented in section III-C.

Remark 1: In the proposed multi-stage communication problem the data rate requirements r_j^{min} and the time horizon T are coupled by the finite duration Δ of each communication stage. Specifically, assume that a source node generates data that it can completely communicate in one communication stage of length Δ by transmitting at a rate r_j^{min} . If this node generates data during part, or the entirety, of the multi-stage duration $T\Delta$, then transmitting at a rate r_j^{min} might not be feasible anymore. In this case, characterizing the tradeoff between r_j^{min} , T , and Δ is critical in obtaining a feasible solution, although other problem-specific considerations can also be important. For simplicity, here we assume that nodes do not generate data when they are not being serviced. While this assumption can sometimes be restrictive, it still allows us to periodically service networks, when otherwise this would not be possible.

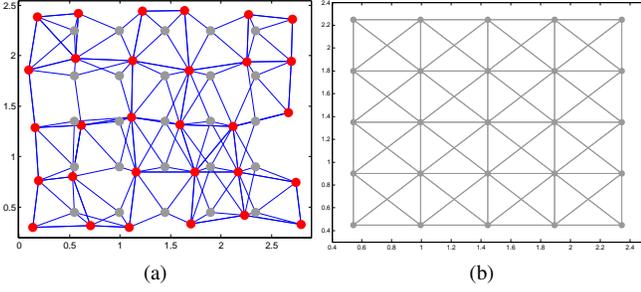


Fig. 4: a) An example problem setup indicating the locations of the source nodes (red), the candidate locations for the APs (gray), as well as the available communication links (blue). b) An example of the mobility graph \mathcal{G} defining the candidate locations for the APs (gray nodes), as well as the permissible movements between these locations (gray lines).

A. Path Planning of the Mobile APs

Consider the *mobility graph* $\mathcal{G} = (\mathcal{M}, \mathcal{E})$ whose set of nodes $\mathcal{M} = \{1, \dots, M\}$ defines the available locations on which the APs can be placed at each time stage of the planning horizon T . For instance, Fig. 4a illustrates an example problem setup indicating the locations of the source nodes, the candidate locations for the APs, as well as the available communication links, i.e., the links that can successfully transmit positive data rates $R_{jn} > 0$, between all nodes. On the other hand, the edges $(m, l) \in \mathcal{E}$ of \mathcal{G} express whether an AP located at node $m \in \mathcal{M}$ at time stage t can move to another node $l \in \mathcal{M}$ at time stage $t+1$. In other words the edge (m, l) is part of \mathcal{G} if and only if an AP can transition from location m to location l in a single time step. The graph \mathcal{G} is defined a priori by the user and will be, in general, a directed graph. An example case for \mathcal{G} is depicted in Fig. 4b. Note that in the proposed formulation, the edges of the graph \mathcal{G} can vary between the various time stages, however, for simplicity here we consider that they remain the same for all $t \in \mathcal{T}$, where \mathcal{T} denotes the set of all time stages, i.e., $\mathcal{T} = \{1, \dots, T\}$.

For each time stage $t \in \mathcal{T}$, we introduce the vector $\mathbf{z}^t \in \{0, 1\}^M$ of binary variables. The value of each entry z_m^t defines whether there is at least one active AP at location $m \in \mathcal{M}$ at time $t \in \mathcal{T}$; we assume that a value of 1 indicates the presence of an AP, while a value of 0 indicates the opposite. Let also $\mathbf{z} \in \{0, 1\}^{MT}$ denote the vector stacking all the \mathbf{z}^t variables. For a given t , the non-zero entries of \mathbf{z}^t will be exactly K if we consider collision avoidance between the APs, i.e., if we require that no more than a single AP can occupy a node $m \in \mathcal{M}$ at each time stage $t \in \mathcal{T}$. On the other hand, if we allow two or more APs to coincide at the same location at the same time, then \mathbf{z}^t will have at most K non-zero entries. Note that given the communication model described in section II, the latter case essentially means that only one of the coinciding APs needs to be active at that particular location and time stage. This is because the communication links to the source nodes will be the same for all the coinciding APs and there is no limit on the amount of data packets that each AP can receive. In what follows we will focus on scenarios where collision avoidance is a necessary requirement, as is typically the case in practice.

Consider further the variables $p_{ml}^t \in \{0, 1\}$ which indicate whether a motion step from location $m \in \mathcal{M}$ to location $l \in \mathcal{M}$ takes place after the communication stage t is completed; we assume that if the variable p_{ml}^t is 1, then a motion step from m to l occurs, while no motion happens for the value 0. Note that we allow the case $m = l$, and that the p_{ml}^t are defined for $t \in \mathcal{T} \setminus \{T\}$, since there is no further motion at the end of the horizon. Let $\mathbf{p}_m^t \in \{0, 1\}^M$ denote the vector stacking all outgoing path variables for location $m \in \mathcal{M}$ at time stage t . Also, let $\mathbf{p}^t \in \{0, 1\}^{M^2}$ indicate the vector stacking all the \mathbf{p}_m^t for a given $t \in \mathcal{T} \setminus \{T\}$, and $\mathbf{p} \in \{0, 1\}^{M^2(T-1)}$ be the vector stacking all the \mathbf{p}^t .

Note that the \mathbf{p}_m^t variables do not contain any information on whether location $m \in \mathcal{M}$ actually contains an AP at time t ; this information is encoded in the variable z_m^t . Hence, it is necessary to couple the variables \mathbf{p} and \mathbf{z} through the appropriate constraints such that the proposed path planning approach makes sense. First, we need to ensure that a motion step occurs from location $m \in \mathcal{M}$ at a time stage $t \in \mathcal{T}$ if and only if there exists an AP at location m at t . For this, we define the following set of constraints

$$z_m^t = \sum_{l \in \mathcal{M}} \mathbf{A}_{ml} p_{ml}^t, \quad \forall m \in \mathcal{M}, t \in \mathcal{T} \setminus \{T\}, \quad (3)$$

that require a motion step to occur from m at t if the location m contained an AP at time t . Here, we have defined the adjacency matrix \mathbf{A} of the mobility graph \mathcal{G} , i.e., the square $M \times M$ matrix whose entry \mathbf{A}_{ml} is 1 if the edge (m, l) belongs to \mathcal{G} , and 0 otherwise. Introducing \mathbf{A}_{ml} in (3) is necessary in order to ensure that the right hand side of (3) only contains admissible motion steps. Note that at most a single admissible motion step can occur from m at t , since at most one AP is located at this node due to the collision avoidance requirement. Hence, only one of the terms $\mathbf{A}_{ml} p_{ml}^t$ must be equal to 1, if z_m^t is also 1, otherwise all p_{ml}^t must be equal to 0. The fact that constraint (3) is an equality constraint ensures that this requirement is satisfied.

Next, we need to ensure that z_m^t takes the value 1 if an admissible motion step from some node l towards m occurs at the time $t-1$, i.e., if $\sum_{l \in \mathcal{M}} \mathbf{A}_{lm} p_{lm}^{t-1} \neq 0$. For this, we impose the following set of constraints

$$\sum_{l \in \mathcal{M}} \mathbf{A}_{lm} p_{lm}^{t-1} = z_m^t, \quad \forall m \in \mathcal{M}, t \in \mathcal{T} \setminus \{1\}, \quad (4)$$

where again we include the terms \mathbf{A}_{lm} in the equation in order to enforce the selection of motion steps that are admissible for a given mobility graph \mathcal{G} . Note that the above set of constraints (4) not only accounts for the correct succession of motion steps for the APs, but it also imposes the collision avoidance requirement. Since the z_m^t and p_{lm}^{t-1} are all binary variables, no more than a single motion step can lead to node $m \in \mathcal{M}$ from any other $l \in \mathcal{M}$. Hence, this precludes the scenario where two APs coincide at the same location at the same time stage.

B. Multi-stage Routing Control

In this section we discuss how to determine the routing variables associated with each communication stage of the

TABLE I: Table of Notation

Sets and Graphs	
\mathcal{J}	\triangleq Set of source nodes
\mathcal{K}	\triangleq Set of AP nodes
\mathcal{N}	\triangleq Union of \mathcal{J} and \mathcal{K}
\mathcal{M}	\triangleq Set of candidate locations for the APs
\mathcal{A}	\triangleq Union of \mathcal{J} and \mathcal{M}
\mathcal{T}	\triangleq Set of time stages
\mathcal{G}	\triangleq Mobility graph for the APs
\mathcal{C}	\triangleq Communication graph for the distributed algorithm
Variables and parameters	
T_{ji}	\triangleq Routing decision from node j to i
R_{ji}	\triangleq Channel reliability from node j to i
r_j	\triangleq Average rate of data generation at node j
r_j^{min}	\triangleq Minimum requirement for rate r_j
f_j, g_{ji}	\triangleq Convex objective functions
z_m^t	\triangleq Variable indicating the presence of an AP at location m and time t
p_{ml}^t	\triangleq Path variable indicating a movement step for an AP from location m to l at time t
d_j^t	\triangleq Variable indicating the satisfaction of r_j^{min} at time t

problem. The goal is to determine when and which source nodes are active, inactive, or must operate as relays. This also entails determining the routing decisions for all the time stages within the planning horizon. The communication model for each time stage is the one presented in Section II. Of course, the optimal routing decisions are directly related to the locations of the APs, which establishes the connection with the path planning problem that was discussed in Section III-A.

In the proposed multi-stage routing problem, nodes that are not being serviced at a given stage t are assumed to not generate data at that time stage. We model this situation by introducing the binary variables $d_j^t \in \{0, 1\}$ that express whether the minimum data rate requirement r_j^{min} for source node $j \in \mathcal{J}$ is “on” or “off” at time stage t . Consequently, we introduce the constraints

$$r_j^t \geq d_j^t r_j^{min}, \quad \forall j \in \mathcal{J}, t \in \mathcal{T}, \quad (5)$$

and
$$\sum_{t \in \mathcal{T}} d_j^t = 1, \quad \forall j \in \mathcal{J}.$$

If $d_j^t = 1$, then the end-to-end rate r_j^t has to be greater or equal to r_j^{min} , such that the minimum data rate requirement is satisfied at time stage t . The constraint $\sum_{t \in \mathcal{T}} d_j^t = 1$ ensures that the r_j^{min} is “on” exactly once over the multi-stage horizon. Using this approach, at each time stage $t \in \mathcal{T}$, every source node $j \in \mathcal{J}$ can be either active, where we have $r_j^t > 0$, act as a relay, where we have $r_j^t = 0$ and at least one $T_{jn}^t > 0$, or be completely inactive, where all r_j^t and T_{jn}^t are zero.

A fundamental difference between the single-stage routing problem discussed in Section II and the proposed multi-stage formulation of this problem is that in the single-stage case the structure of the communication network, that is encoded in the rate functions R_{jn} , is fixed. However, when mobile APs are considered the network structure is time-varying and depends on the AP locations. This means that the rate functions R_{jn} are now variables, which are multiplied with the routing decisions T_{jn} in the communication model, cf.

(2). To address the nonlinearities introduced by considering the products of variables $R_{jn}T_{jn}$, we exploit the fact that the APs can only be located on the nodes of the graph \mathcal{G} . More specifically, we define the set \mathcal{A} as the union of \mathcal{J} and \mathcal{M} , i.e. $\mathcal{A} = \{1, \dots, J, J+1, \dots, J+M\}$, and then define the routing decision variables T_{ja}^t over all $a \in \mathcal{A}$. Then, it is necessary to impose constraints which ensure that if a node $m \in \mathcal{M}$ contains an AP at time t , then the T_{jm}^t towards that location m are allowed to be nonzero, while no routing towards m should be allowed if $z_m^t = 0$. To achieve this, we couple the routing decision variables T_{jm}^t from every $j \in \mathcal{J}$ to some $m \in \mathcal{M}$ with the variables $z_m^t \in \{0, 1\}$, cf. Section III-A, using the following set of constraints

$$\sum_{j \in \mathcal{J}} T_{jm}^t \leq Q_m z_m^t, \quad \forall m \in \mathcal{M}, t \in \mathcal{T}. \quad (6)$$

Here, the parameter Q_m denotes the number of $j \in \mathcal{J}$ that can successfully transmit positive data rates R_{jm} to a receiver located at a particular node $m \in \mathcal{M}$. The inclusion of this parameter is necessary to account for the fact that for any location m there are multiple T_{jm}^t which can take any value in the $[0, 1]$ interval.

C. The MIP Problem

Based on the discussions in Sections III-A and III-B we obtain the following MIP formulation of the connectivity scheduling problem

$$\begin{aligned} \min_{\mathbf{r}, \mathbf{T}, \mathbf{z}, \mathbf{p}, \mathbf{d}} \quad & \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} f_j(r_j^t) + \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} \sum_{a \in \mathcal{A}} g_{ja}(T_{ja}^t) \\ \text{s.t.} \quad & r_j^t = \sum_{a \in \mathcal{A}} T_{ja}^t R_{ja} - \sum_{i \in \mathcal{J}} T_{ij}^t R_{ij}, \quad \forall t \in \mathcal{T}, j \in \mathcal{J}, \end{aligned} \quad (7a)$$

$$\sum_{a \in \mathcal{A}} T_{ja}^t \leq 1, \quad \forall t \in \mathcal{T}, j \in \mathcal{J}, \quad (7b)$$

$$r_j^t \geq d_j^t r_j^{min}, \quad \forall t \in \mathcal{T}, j \in \mathcal{J}, \quad (7c)$$

$$\sum_{t \in \mathcal{T}} d_j^t = 1, \quad \forall j \in \mathcal{J}, \quad (7d)$$

$$\sum_{j \in \mathcal{J}} T_{jm}^t \leq Q_m z_m^t, \quad \forall m \in \mathcal{M}, t \in \mathcal{T}, \quad (7e)$$

$$\sum_{l \in \mathcal{M}} \mathbf{A}_{lm} p_{lm}^{t-1} = z_m^t, \quad \forall t \in \mathcal{T} \setminus \{1\}, m \in \mathcal{M}, \quad (7f)$$

$$\sum_{l \in \mathcal{M}} \mathbf{A}_{ml} p_{ml}^t = z_m^t, \quad \forall t \in \mathcal{T} \setminus \{T\}, m \in \mathcal{M}, \quad (7g)$$

$$T_{ja}^t \in [0, 1], \quad \forall t \in \mathcal{T}, j \in \mathcal{J}, a \in \mathcal{A}, \quad (7h)$$

$$z_m^t, d_j^t \in \{0, 1\}, \quad \forall t \in \mathcal{T}, m \in \mathcal{M}, j \in \mathcal{J}, \quad (7i)$$

$$p_{ml}^t \in \{0, 1\}, \quad \forall t \in \mathcal{T} \setminus \{T\}, m, l \in \mathcal{M}, \quad (7j)$$

$$z_m^1 = I_m, \quad \forall m \in \mathcal{M}, \quad (7k)$$

where the $\mathbf{r}, \mathbf{T}, \mathbf{z}, \mathbf{p}, \mathbf{d}$ denote the vectors stacking all the corresponding variables. The very last set of constraints $z_m^1 = I_m, \forall m \in \mathcal{M}$, are the initialization constraints that determine the starting locations of the APs; we set $I_m = 1$ for all the K locations $m \in \mathcal{M}$ that contain an AP at $t = 1$, and $I_m = 0$ otherwise. Table I summarizes the notation used in this paper.

IV. DISTRIBUTED COMMUNICATION SCHEDULING

The single MIP formulation (7) for the connectivity scheduling problem presented in Section III can be computationally challenging as the problem size grows. Moreover, in practical applications there are additional issues arising from the fact that all the problem parameters need to be communicated to a central computing unit, which will solve the optimization problem and then send the optimal decisions back to the nodes. Moreover, central gathering of information is vulnerable to delays and changes to the problem parameters, while in certain cases it also incurs privacy and security issues.

To address these issues, in this section we propose a distributed algorithm to solve the connectivity scheduling problem. Here, by distributed we mean a scheme where the members of the set $\mathcal{A} = \{1, \dots, J + M\}$ are all independent processors which cooperate to solve the connectivity scheduling problem (7) in an iterative fashion, by exchanging information over the network defined by the rate functions R_{ij} . Towards this goal, in Section IV-A, we first reformulate the MIP problem (7) into a convex problem by relaxing the integrality constraints into continuous ones. Then, in Section IV-B, we propose a distributed solution to this convex relaxation, based on the Accelerated Distributed Augmented Lagrangians (ADAL) algorithm that was recently developed by the authors in [26]. ADAL is a distributed optimization method that relies on *augmented Lagrangians* (AL), a regularization technique that is obtained by adding a quadratic penalty term to the ordinary Lagrangian of a problem [29]. In this paper, we modify ADAL by incorporating into the algorithm sparse optimization techniques, in order to recover the desired solution for the original MIP problem (7), which has only 0 or 1 values for the (now relaxed) binary variables. Specifically, we propose a scheme that utilizes a re-weighted ℓ_1 regularization approach embedded into the iterative procedure of ADAL; cf. [23] for more details on the origins of the re-weighted ℓ_1 regularization approach for non-distributed solutions.

A. Continuous Relaxation using Sparse Optimization

To obtain a continuous and convex relaxation of the original MIP problem (7) we first need to relax the integrality constraints of (7) by allowing the binary variables z_m^t , d_j^t , p_{ml}^t to lie anywhere in the $[0, 1]$ interval. Since the solution of the relaxed problem will in general lead to non-integral solutions, we then need to employ sparse optimization techniques that promote solutions with only a few non-zero entries for the vectors \mathbf{z}^t , \mathbf{d}^t , \mathbf{p}_m^t , $\forall t \in \mathcal{T}$, $m \in \mathcal{M}$. A standard approach is to add sparsity-promoting regularization terms to the objective function, with the most common being the ℓ_1 norm.

In our problem, we have three groups of binary variables that have been relaxed. First, there are the vectors $\mathbf{z}^t \in [0, 1]^M$ for every $t \in \mathcal{T}$ which must contain exactly K entries equal to 1 and the rest should be 0, to account for the presence of the APs at K nodes of the graph \mathcal{G} at each time stage. Second, there are the vectors \mathbf{d}_j for every $j \in \mathcal{J}$, which must have exactly one entry equal to 1 that determines the time stage when the r_j^{\min} of node j is activated. Finally, there are the vectors \mathbf{p}_m^t for every $m \in \mathcal{M}$ and $t \in \mathcal{T} \setminus \{T\}$ that must have

all zero entries if the location m does not contain an AP at time t . Otherwise, the \mathbf{p}_m^t must contain a single entry equal to 1 indicating the location $l \in \mathcal{M}$ to which the AP will move next. Based on this, the regularization terms that need to be added to obtain a continuous relaxation of the original MIP problem (7) are as follows

$$\sum_{t \in \mathcal{T}} \|\mathbf{z}^t\|_1 + \sum_{j \in \mathcal{J}} \|\mathbf{d}_j\|_1 + \sum_{t \in \mathcal{T} \setminus \{T\}} \sum_{m \in \mathcal{M}} \|\mathbf{p}_m^t\|_1. \quad (8)$$

Note that these terms are included to enforce entries that are either 0 or 1. The actual number of entries that should be 1 and 0 is determined by the remaining problem formulation and the interactions between variables. For example, the \mathbf{d}_j variables are subject to the constraints $\sum_{t \in \mathcal{T}} d_j^t = 1$, $\forall j \in \mathcal{J}$, while the \mathbf{z}^t variables are subject to the initialization constraints $z_m^1 = I_m$, $\forall m \in \mathcal{M}$, and are also coupled with the routing \mathbf{T} and path \mathbf{p} variables through multiple additional constraints.

An undesirable characteristic of ℓ_1 regularization is the dependence on magnitude; larger coefficients are penalized more heavily in the ℓ_1 norm than smaller coefficients. The most popular way to improve on this issue is given by a re-weighting iterative procedure proposed recently in [23]. The idea is to use appropriate weights on the regularization terms, and update these weights at each iteration. For our problem, the weighted version of the regularization terms in (8) takes the form

$$\sum_{t \in \mathcal{T}} (\mathbf{w}^t)^T \mathbf{z}^t + \sum_{j \in \mathcal{J}} \mathbf{u}_j^T \mathbf{d}_j + \sum_{t \in \mathcal{T} \setminus \{T\}} \sum_{m \in \mathcal{M}} (\mathbf{v}_m^t)^T \mathbf{p}_m^t, \quad (9)$$

where $(\cdot)^T$ denotes the transpose, and we have defined the weighting vectors $\mathbf{w}^t \in \mathbb{R}^M$, $\mathbf{u}_j \in \mathbb{R}^T$, and $\mathbf{v}_m^t \in \mathbb{R}^M$, and have also taken into account that all variables are non-negative such that the ℓ_1 norm of each vector is simply the sum of its entries.

At each iteration k of the re-weighted ℓ_1 scheme, we solve the following convex optimization problem

$$\begin{aligned} \min_{\mathbf{r}, \mathbf{T}, \mathbf{z}, \mathbf{p}, \mathbf{d}} \sum_{t \in \mathcal{T}} \left[(\mathbf{w}^{k,t})^T \mathbf{z}^t + \sum_{j \in \mathcal{J}} f_j(r_j^t) + \sum_{j \in \mathcal{J}} \sum_{a \in \mathcal{A}} g_{ja}(T_{ja}^t) \right] \\ + \sum_{j \in \mathcal{J}} (\mathbf{u}_j^k)^T \mathbf{d}_j + \sum_{t \in \mathcal{T} \setminus \{T\}} \sum_{m \in \mathcal{M}} (\mathbf{v}_m^{k,t})^T \mathbf{p}_m^t \\ \text{s.t. Constraints (7a)-(7h), and (7k),} \\ z_m^t, d_j^t \in [0, 1], \quad \forall t \in \mathcal{T}, m \in \mathcal{M}, j \in \mathcal{J}, \\ p_{ml}^t \in [0, 1], \quad \forall t \in \mathcal{T} \setminus \{T\}, m, l \in \mathcal{M}, \end{aligned} \quad (10)$$

where the terms $\mathbf{w}^{k,t}$, \mathbf{u}_j^k , $\mathbf{v}_m^{k,t}$ denote the values of the corresponding weighting vectors at iteration k . After solving (10) at each iteration k , we obtain the minimizers \mathbf{r}^k , \mathbf{T}^k , \mathbf{z}^k , \mathbf{p}^k , \mathbf{d}^k , the values of which are used to update the weighting terms for the next iteration $k + 1$ according to

$$\mathbf{w}^{k+1,t} = \frac{1}{\mathbf{z}^{k,t} + \epsilon_w}, \quad \forall t \in \mathcal{T}, \quad (11a)$$

$$\mathbf{u}_j^{k+1} = \frac{1}{\mathbf{d}_j^k + \epsilon_u}, \quad \forall j \in \mathcal{J}, \quad (11b)$$

$$\mathbf{v}_m^{k+1,t} = \frac{1}{\mathbf{p}_m^{k,t} + \epsilon_v}, \quad \forall m \in \mathcal{M}, t \in \mathcal{T} \setminus \{T\}. \quad (11c)$$

Algorithm 1 Re-weighted ℓ_1 method

Set $k = 1$. Define stability parameters $\epsilon_w, \epsilon_u, \epsilon_v$, and initial weight parameters $\mathbf{w}^1, \mathbf{u}^1, \mathbf{v}^1$.

1. For fixed weight vectors $\mathbf{w}^k, \mathbf{u}^k, \mathbf{v}^k$, calculate the variables $\mathbf{r}^k, \mathbf{T}^k, \mathbf{z}^k, \mathbf{p}^k, \mathbf{d}^k$ as the solutions to problem (10).
 2. Set the weights $\mathbf{w}^{k+1}, \mathbf{u}^{k+1}, \mathbf{v}^{k+1}$ according to eq. (11).
 3. Terminate if k reaches a pre-specified number of maximum iterations, or if the variables $\mathbf{r}^k, \mathbf{T}^k, \mathbf{z}^k, \mathbf{p}^k, \mathbf{d}^k$ did not change significantly from the previous iteration. Otherwise, increase k by one and return to Step 1.
-

The terms $\epsilon_w \in \mathbb{R}_+^M$, $\epsilon_u \in \mathbb{R}_+^T$, and $\epsilon_v \in \mathbb{R}_+^M$ in (11) are stability parameters. They ensure that very small values of $\mathbf{z}^{k,t}$, \mathbf{d}_j^k , and $\mathbf{p}_{m^{k+1},t}^k$ do not cause ill-conditioning in the optimization problem (10). As suggested in [23], the best values for the stability terms would be slightly smaller than the expected nonzero magnitudes of their corresponding variables. Nonetheless, the re-weighted procedure is, in general, reasonably robust to the choice of these parameters. The method is terminated either when a maximum pre-defined number of iterations is reached or when convergence is reached, i.e., when the values of the variables $\mathbf{r}^k, \mathbf{T}^k, \mathbf{z}^k, \mathbf{p}^k, \mathbf{d}^k$ stop varying significantly between successive iterations [23]. The exact value that determines termination is user-defined based on the desirable accuracy levels, with a value of 10^{-3} for the max-norm on the variables being preferred in most practical applications. The method is summarized in Alg. 1.

B. A Distributed Connectivity Scheduling Method

To develop the proposed distributed connectivity scheduling method, we first need to decompose the optimization problem (10) into appropriate subproblems that are solved iteratively and in parallel by a collection of computing nodes, henceforth referred to as “agents”, that exchange information via a dedicated communication network. In principle, there are many ways to decompose this problem. Here, we assume a general decomposition scheme, wherein all source nodes $j \in \mathcal{J}$ and all candidate AP locations $m \in \mathcal{M}$ are agents that coordinate to jointly solve the connectivity scheduling problem via the communication network defined by the rate functions R_{ja} . Note that, in the proposed formulation, the candidate AP locations are just points in two-dimensional space that the available mobile APs can move onto. Hence, in practical applications, they will typically not possess any processing or communication capabilities. This is easily resolved in practice by assigning the duties of each candidate AP location to any of the neighboring source nodes. Essentially, under this approach the source nodes coordinate in a distributed fashion to decide the optimal path for the mobile APs, as well as the optimal routing decisions. In what follows, we employ ADAL to characterize the coordination process that the agents use to solve the connectivity scheduling problem at hand.

The main building block of ADAL is the collections of the local convex subproblems that are solved independently by the agents. Given the aforementioned decomposition structure, these subproblems are formulated as follows. For clarity, we

will refer to the J agents corresponding to source nodes as “source agents”, and to the M agents corresponding to candidate AP locations as the “AP agents”. We also recall that the set $\mathcal{A} = \{1, \dots, J + M\}$ is comprised of all the agents. Every source agent $j \in \mathcal{J}$ is responsible for its routing variables towards other source nodes only, i.e., the variables r_j^t, d_j^t, T_{ji}^t , for all $t \in \mathcal{T}$ and $i \in \mathcal{J}$. On the other hand, every AP agent $m \in \mathcal{M}$ is responsible for the variables z_m^t, p_{ml}^t for all $l \in \mathcal{M}, t \in \mathcal{T}$, and also the routing decisions T_{jm}^t from all source nodes $j \in \mathcal{J}$ towards itself for all $t \in \mathcal{T}$. Under this decomposition structure, most of the constraints of problem (10) are local to the corresponding agents. The only coupling constraints between the agents, i.e., constraints involving variables controlled by more than one agent, are the routing flow constraints

$$r_j^t = \sum_{i \in \mathcal{J}} T_{ji}^t R_{ji} + \sum_{m \in \mathcal{M}} T_{jm}^t R_{jm} - \sum_{i \in \mathcal{J}} T_{ij}^t R_{ij}, \quad (12)$$

for all $j \in \mathcal{J}$, $t \in \mathcal{T}$, the routing sum constraints

$$\sum_{i \in \mathcal{J}} T_{ji}^t + \sum_{m \in \mathcal{M}} T_{jm}^t + q_j^t = 1, \quad (13)$$

for all $j \in \mathcal{J}$, $t \in \mathcal{T}$, and finally the path constraints

$$\sum_{l \in \mathcal{M}} \mathbf{A}_{lm} p_{lm}^{t-1} = z_m^t, \quad (14)$$

for all $m \in \mathcal{M}$, $t \in \mathcal{T} \setminus \{1\}$. Here, we introduced the slack variables $q_j^t > 0$ in (13) for all $t \in \mathcal{T}$, $j \in \mathcal{J}$, since ADAL requires the coupling constraints to be equalities, instead of inequalities as expressed in (10).

Denoting by \mathbf{x}_a^t the vector stacking all decision variables of an agent $a \in \mathcal{A}$ for time stage $t \in \mathcal{T}$, including the slack variables, then we can define appropriate matrices \mathbf{R}_a^t and \mathbf{E}_a^t for all $a \in \mathcal{A}$, $t \in \mathcal{T}$ such that (12) is compactly written as

$$\sum_{a \in \mathcal{A}} \mathbf{R}_a^t \mathbf{x}_a^t = \mathbf{0}, \quad \forall t \in \mathcal{T}, \quad (15)$$

while the constraints (13) take the form

$$\sum_{a \in \mathcal{A}} \mathbf{E}_a^t \mathbf{x}_a^t = \mathbf{1}, \quad \forall t \in \mathcal{T}. \quad (16)$$

Similarly, we can define appropriate matrices \mathbf{P}_m^t for all $m \in \mathcal{M}$ and $t \in \mathcal{T} \setminus \{1\}$ such that the path constraints (14) become

$$\sum_{m \in \mathcal{M}} \mathbf{P}_m^t \mathbf{x}_m^t = \mathbf{0}, \quad \forall t \in \mathcal{T} \setminus \{1\}. \quad (17)$$

Note that only the AP agents are coupled in (17).

In ADAL, the coupling constraints are accounted for by the AL framework; cf. [26] for a detailed overview and explanation of ADAL, and [30, 31] for an in-depth presentation of the AL framework. This entails introducing, for all $t \in \mathcal{T}$, the Lagrange (dual) variables $\boldsymbol{\lambda}^t \in \mathbb{R}^J$ for the J routing flow constraints (15), and also the $\boldsymbol{\mu}^t \in \mathbb{R}^J$ for the J routing sum constraints (16). Then, for every source agent $j \in \mathcal{J}$ we define

its local augmented Lagrangian Λ_j^ρ at iteration k of ADAL as

$$\begin{aligned} \Lambda_j^\rho(\mathbf{x}_j, \mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) &= (\mathbf{u}_j^k)^T \mathbf{d}_j + \sum_{t \in \mathcal{T}} \left[f_j(r_j^t) + \sum_{i \in \mathcal{J}} g_{ji}(T_{ji}^t) \right. \\ &+ (\boldsymbol{\lambda}^{k,t})^T \mathbf{R}_i^t \mathbf{x}_j^t + (\boldsymbol{\mu}^{k,t})^T \mathbf{E}_j^t \mathbf{x}_j^t + \frac{\rho}{2} \|\mathbf{R}_j^t \mathbf{x}_j^t + \sum_{a \in \mathcal{A}} \mathbf{R}_a^t \mathbf{x}_a^{k,t}\|^2 \\ &\left. + \frac{\rho}{2} \|\mathbf{E}_j^t \mathbf{x}_j^t + \sum_{a \in \mathcal{A}} \mathbf{E}_a^t \mathbf{x}_a^{k,t} - \mathbf{1}\|^2 \right], \end{aligned} \quad (18)$$

where \mathbf{x}_a^k denotes the vector of decision variables communicated to agent j from agent a at the beginning of iteration k . With respect to agent j , these are considered fixed parameters. Also, \mathbf{x}_j is the vector stacking all the decision variables of agent j , and the $\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k$ are the vectors stacking the corresponding variables for all time stages. Recall that we use the superscript k, t to denote the value of a variable pertaining to time stage t at iteration k . The penalty coefficient $\rho > 0$ in (18) is a properly defined parameter; cf. [29, 30] for an in-depth discussion on the role of ρ within the AL framework. In general, ρ can be a vector, or we can define different ρ for each penalty term to account for the varying magnitudes among the different constraints. For simplicity and without loss of generality, here we assume a single common scalar ρ . In order to form the local AL (18), agent j needs access only to a subset of the entries of $\mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k$ that are locally available. More specifically, for the dual variables $\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k$, it needs only those entries a for which $[\mathbf{R}_j^t]_a \neq \mathbf{0}$ and $[\mathbf{E}_j^t]_a \neq \mathbf{0}$, respectively, where $[\cdot]_a$ denotes the a -th row of a matrix. Define the graph $\mathcal{C} = (\mathcal{A}, \mathcal{D})$, where an edge (j, a) belongs to \mathcal{D} if and only if $R_{ja} > 0$, for all $j, a \in \mathcal{A}$; cf. Fig. 4a for an example of \mathcal{C} . Then, agent j needs access to the dual variables λ_a^k that correspond to the routing flow constraints of its 1-hop source neighbors in \mathcal{C} , and to the dual variable μ_j^k corresponding to its own routing sum constraint. For the primal variables \mathbf{x}_a^k that appear in the penalty terms, agent j needs access to all the routing decisions that are involved in the routing constraints of its 1-hop source neighbors in \mathcal{C} , and also the routing decisions T_{jm}^t for those AP agents $m \in \mathcal{M}$ with which it shares a positive link reliability metric R_{jm} .

Next, we need to define the local ALs for the AP agents. This entails introducing, for all time stages $t \in \mathcal{T} \setminus \{1\}$, the dual variables $\boldsymbol{\nu}^t \in \mathbb{R}^M$ that correspond to the M path constraints (17). Then, the local AL $\tilde{\Lambda}_m^\rho$ of each AP agent $m \in \mathcal{M}$ at iteration k is defined as

$$\begin{aligned} \tilde{\Lambda}_m^\rho(\mathbf{x}_m, \mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\nu}^k) &= \sum_{t \in \mathcal{T}} (w_m^{k,t})^T z_m^t + \sum_{t \in \mathcal{T} \setminus \{T\}} (\mathbf{v}_m^{k,t})^T \mathbf{P}_m^t \\ &+ \sum_{t \in \mathcal{T}} \left[\sum_{j \in \mathcal{J}} g_{jm}(T_{jm}^t) + (\boldsymbol{\lambda}^{k,t})^T \mathbf{R}_m^t \mathbf{x}_m^t + (\boldsymbol{\mu}^{k,t})^T \mathbf{E}_m^t \mathbf{x}_m^t + \right. \\ &\left. \frac{\rho}{2} \|\mathbf{R}_m^t \mathbf{x}_m^t + \sum_{a \in \mathcal{A}} \mathbf{R}_a^t \mathbf{x}_a^{k,t}\|^2 + \frac{\rho}{2} \|\mathbf{E}_m^t \mathbf{x}_m^t + \sum_{a \in \mathcal{A}} \mathbf{E}_a^t \mathbf{x}_a^{k,t} - \mathbf{1}\|^2 \right] \\ &+ \sum_{t \in \mathcal{T} \setminus \{1\}} \left[(\boldsymbol{\nu}^{k,t})^T \mathbf{P}_m^t \mathbf{x}_m^t + \frac{\rho}{2} \|\mathbf{P}_m^t \mathbf{x}_m^t + \sum_{l \in \mathcal{M}} \mathbf{P}_l^t \mathbf{x}_l^{k,t}\|^2 \right]. \end{aligned} \quad (19)$$

Recall that in the proposed decomposition structure, each AP agent $m \in \mathcal{M}$ is responsible for determining the routing decisions T_{jm}^t for those source agents $j \in \mathcal{J}$ that can directly route to m , i.e., those $j \in \mathcal{J}$ for which $R_{jm} > 0$. Let \mathcal{F} denote the set containing all such source nodes. This means that the AP agent needs access to the entries of the dual variables $\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k$ that correspond to the constraints of all source nodes that belong to \mathcal{F} . Similarly, it needs access to the entries of the dual variables $\boldsymbol{\nu}^k$ that correspond to the respective constraints of its 1-hop neighbors in the mobility graph \mathcal{G} . For the first two penalty terms in (19), the AP agent needs access to the variables included in the routing constraints of the source nodes that belong to \mathcal{F} , while for the last penalty term it needs the path variables from its 1- and 2-hop neighbors in the mobility graph \mathcal{G} .

We are now ready to define the local subproblems that are solved independently and in parallel at each iteration by the agents. For the source agents $j \in \mathcal{J}$, the local subproblems are

$$\begin{aligned} \min_{\mathbf{x}_j} \quad & \Lambda_j^\rho(\mathbf{x}_j, \mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \\ \text{s.t.} \quad & r_j^t \geq d_j^t r_j^{\min}, \quad \forall t \in \mathcal{T}, \\ & \sum_{t \in \mathcal{T}} d_j^t = 1, \\ & T_{ji}^t \in [0, 1], \quad \forall t \in \mathcal{T}, i \in \mathcal{J}, \\ & d_j^t \in [0, 1], \quad \forall t \in \mathcal{T}. \end{aligned} \quad (20)$$

On the other hand, every AP agent $m \in \mathcal{M}$ solves the local subproblem

$$\begin{aligned} \min_{\mathbf{x}_m} \quad & \tilde{\Lambda}_m^\rho(\mathbf{x}_m, \mathbf{x}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \\ \text{s.t.} \quad & \sum_{j \in \mathcal{J}} T_{jm}^t \leq Q_m z_m^t, \quad \forall t \in \mathcal{T}, \\ & \sum_{l \in \mathcal{M}} \mathbf{A}_{ml} p_{ml}^t = z_m^t, \quad \forall t \in \mathcal{T} \setminus \{T\}, \\ & T_{jm}^t \in [0, 1], \quad \forall t \in \mathcal{T}, j \in \mathcal{J}, \\ & z_m^t \in [0, 1], \quad \forall t \in \mathcal{T}, \\ & p_{ml}^t \in [0, 1], \quad \forall t \in \mathcal{T} \setminus \{T\}, l \in \mathcal{M}, \\ & z_m^1 = I_m. \end{aligned} \quad (21)$$

After finding the minimizers $\hat{\mathbf{x}}_a^k$ of (20) and (21) at iteration k , all agents update the primal variables, that need to be subsequently communicated to their neighbors, according to

$$\mathbf{x}_a^{k+1} = \mathbf{x}_a^k + \tau(\hat{\mathbf{x}}_a^k - \mathbf{x}_a^k). \quad (22)$$

Here, τ is an appropriately defined stepsize parameter. According to the theoretical analysis in [26], for the problem under consideration here, τ must satisfy $\tau \leq 1/q$, where q denotes the maximum node degree of the graph \mathcal{C} . Note also that the update (22) does not involve the \mathbf{d} variables, since these variables do not appear in any penalty term of the local ALs and, hence, do not need to be communicated.

The weight parameters are updated locally at each agent after the subproblems (20) and (21) have been solved at iteration k , in a manner similar to the centralized method (11).

Algorithm 2 ADAL with re-weighted ℓ_1 regularization

Set $k = 1$. Define stability parameters $\epsilon_w, \epsilon_u, \epsilon_v$, initial weight parameters $\mathbf{w}^1, \mathbf{u}^1, \mathbf{v}^1$, initial primal variables \mathbf{x}_a^1 for all $a \in \mathcal{A}$, and initial dual variables λ^1, μ^1, ν^1 .

1. For fixed weight vectors $\mathbf{w}^k, \mathbf{u}^k, \mathbf{v}^k$, primal variables \mathbf{x}_a^k for all $a \in \mathcal{A}$, and dual variables λ^k, μ^k, ν^k , every source agent $j \in \mathcal{J}$ calculates the minimizer $\hat{\mathbf{x}}_j^k$ according to (20), while every AP agent $m \in \mathcal{M}$ calculates the minimizer $\hat{\mathbf{x}}_m^k$ according to (21).
2. Every agent $a \in \mathcal{A}$ updates its primal variables according to (22), and communicates the results to its appropriate neighbors. It also updates its local weight parameters according to (23).
3. After receiving the updated primal variables from their neighbors, every source agent $j \in \mathcal{J}$ updates its dual variables $\lambda_j^{k+1,t}, \mu_j^{k+1,t}$ for all $t \in \mathcal{T}$ according to (24)-(25), while every AP agent updates its dual variables $\nu_m^{k+1,t}$ for all $t \in \mathcal{T} \setminus \{1\}$ according to (26).
3. Terminate if the max-norm of all constraint violations falls below a pre-specified threshold. Otherwise, increase k by one and return to Step 1.

Specifically, they take the form

$$\mathbf{w}^{k+1,t} = \frac{1}{\hat{\mathbf{z}}^{k,t} + \epsilon_w}, \quad \forall t \in \mathcal{T}, \quad (23a)$$

$$\mathbf{u}_j^{k+1} = \frac{1}{\hat{\mathbf{d}}_j^k + \epsilon_u}, \quad \forall j \in \mathcal{J}, \quad (23b)$$

$$\mathbf{v}_m^{k+1,t} = \frac{1}{\hat{\mathbf{p}}_m^{k,t} + \epsilon_v}, \quad \forall t \in \mathcal{T} \setminus \{1\}, m \in \mathcal{M}. \quad (23c)$$

Note that the weight parameters are only locally controlled and do not need to be communicated.

After the updated primal variables, cf. (22), have been communicated, the dual update steps take place according to

$$\begin{aligned} \lambda_j^{k+1,t} &= \lambda_j^{k,t} + \\ &\tau \rho \left(r_j^{k+1,t} + \sum_{i \in \mathcal{J}} T_{ij}^{k+1,t} R_{ij} - \sum_{a \in \mathcal{A}} T_{ja}^{k+1,t} R_{ja} \right), \end{aligned} \quad (24)$$

and

$$\mu_j^{k+1,t} = \mu_j^{k,t} + \tau \rho \left(\sum_{a \in \mathcal{A}} T_{ja}^{k+1,t} + q_j^{k+1,t} - 1 \right). \quad (25)$$

The dual update steps are also distributed, since they only require access to the variables associated with the corresponding constraints. A straightforward approach here is to let each source agent $j \in \mathcal{J}$ be responsible for the dual variables λ_j^t, μ_j^t for all $t \in \mathcal{T}$, which correspond to its own routing flow and routing sum constraints. Similarly, each AP agent $m \in \mathcal{M}$ is responsible to update its dual variables ν_m^t for all $t \in \mathcal{T} \setminus \{1\}$ according to

$$\nu_m^{k+1,t} = \nu_m^{k,t} + \tau \rho \left(\sum_{l \in \mathcal{M}} \mathbf{A}_{ml} p_{ml}^{k+1,t-1} - z_m^{k+1,t} \right). \quad (26)$$

A common termination criterion for AL methods, distributed or not, is to monitor the amount of constraint violations for the constraints that have been incorporated into

the AL. For our problem, this includes the TJ routing flow constraints (12), the TJ routing sum (13) constraints, and the $(T-1)M$ path constraints (14). A threshold of 10^{-3} for the max-norm of all constraint violations is considered sufficient in most practical applications [31]. Moreover, for distributed schemes such as the one proposed here, it is desirable to terminate all the agents' computations and updates at the same time. This is usually achieved via a "flooding" mechanism, where if an agent satisfies the local termination criterion (in our case the local constraint violations being below a user-defined threshold) continuously for some reasonable number of iterations, it transmits a "termination" flag to all its neighbors. The updated flags are re-transmitted by all agents at each iteration, thereby "flooding" the network. The agents terminate when their list of flags only contains "termination" ones.

The proposed modified ADAL method with re-weighted ℓ_1 regularization is summarized in Alg. 2. Note that we have presented a scheme wherein the two iterative procedures, i.e., ADAL and the re-weighting approach, are merged into a single iteration indexed by k . The convergence of this scheme is not straightforward, since the original theoretical analysis of ADAL [26] assumes a fixed objective function from iteration to iteration, while the re-weighting procedure here causes the objective function to change. Nevertheless, numerical experiments presented in Section V show that the method converges fast to a desired integer solution.

Remark 2 (Decomposition Scheme): Note that alternative decomposition schemes are also possible. For instance, we could have let each source node $j \in \mathcal{J}$ control all the routing decisions T_{jm}^t for every $m \in \mathcal{M}$. In that scenario, the routing sum constraints (7b) would be local to the source agent i , and the constraints (7e) would be the ones coupling the decisions between source and AP agents, instead.

V. NUMERICAL EXPERIMENTS

In this section we present numerical results in order to illustrate the validity and effectiveness of the proposed distributed method for the connectivity scheduling problem. The main objectives here are two: i) we verify that the proposed combination of ADAL with the re-weighted ℓ_1 heuristic converges for various problem setups, and ii) we illustrate that the proposed distributed scheme converges to a solution that has the desired 0-1 structure for the relaxed binary variables.

The simulations presented below are carried out in MATLAB using the IBM CPLEX solver. As our first study case, we consider the problem (P1) depicted in Fig. 5, which consists of a network with $J = 72$ source nodes, $K = 2$ mobile APs, and $M = 25$ candidate AP locations. The permissible motion steps between candidate AP locations are the same as in Fig. 4b, i.e., left, right, up, down, and diagonally. We formulate a connectivity scheduling problem (7) where the goal is to satisfy the minimum data rate requirements r_j^{\min} for all source nodes $j \in \mathcal{J}$, while also using as short routing paths as possible. The objective function does not contain any $f_j(r_j)$ functions, since we aim to simply meet the minimum requirements, and are not interested in maximizing the rates. Moreover, we set $g_{ji}(T_{ji}) = T_{ji}$ to penalize long routing paths

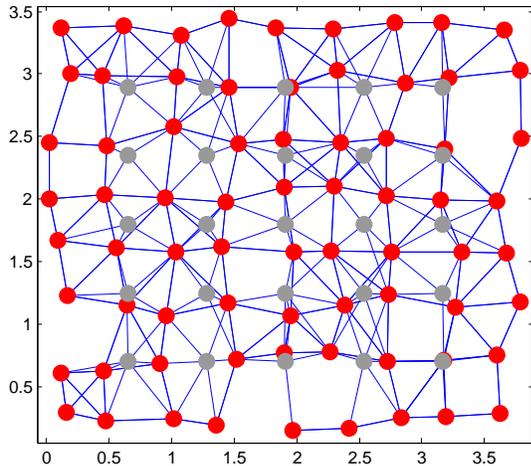


Fig. 5: The network configuration for problem (P1). The red dots correspond to the 72 source nodes, the grey dots are the 25 candidate AP locations, while the blue lines indicate links that can successfully transmit positive data rates, i.e., $R_{ji} > 0$.

towards the APs, i.e., we express preference towards solutions with only a few nonzero T_{ji} .

The parameters r_j^{min} are randomly generated by sampling from a uniform distribution in the interval $[0, 1]$. For simplicity, the rate functions R_{ji} can only take the values 0 and 1, depending on the distance between the nodes j and i . The links with $R_{ji} = 1$ are shown in Fig. 5. We deliberately consider a problem where the minimum rate requirements r_j^{min} cannot be satisfied in a single communication stage with $K = 2$ APs positioned in the top right and bottom left positions; see Fig. 6b for the network configuration of this particular problem. In other words, we intentionally study a problem for which the formulation described in Section II leads to an infeasible optimization problem, such that the proposed multi-stage connectivity scheduling approach is necessary.

Fig. 6 and 7 illustrate the solutions of (P1) obtained by the centralized MIP formulation and the distributed convex relaxation scheme for a motion horizon of $T = 5$ time stages. Note that the choice of $M = 25$ makes sense given the horizon of 5 time steps; larger values for M would involve redundant nodes that the mobile APs would never be able to visit, due to the 5-step horizon. For each time stage t , we plot the active source nodes, for which $r_j^t \geq r_j^{min}$, the relay nodes, for which $r_j^t = 0$ and at least one $T_{ji}^t > 0$, and also the inactive nodes, for which all T_{ji}^t and r_j^t are zero. Moreover, we plot the current locations of the two mobile APs, i.e., the two nodes $m \in \mathcal{M}$ for which $z_m = 1$, and all the routing decisions T_{ji}^t for each stage t . Observing Fig. 6 and 7, we see that the proposed distributed method actually converges to a desired solution, where all the source nodes are serviced within the time horizon. Moreover, we observe that the centralized solution differs from the distributed one. This is to be expected since in the distributed case we solve a relaxed version of the original non-convex MIP problem.

For the simulation depicted in Fig. 6 and 7, the stability parameters $\epsilon_w, \epsilon_u, \epsilon_v$ are all set to 0.005, while the penalty coefficient for the local ALs is set to $\rho = 40$. The stepsize

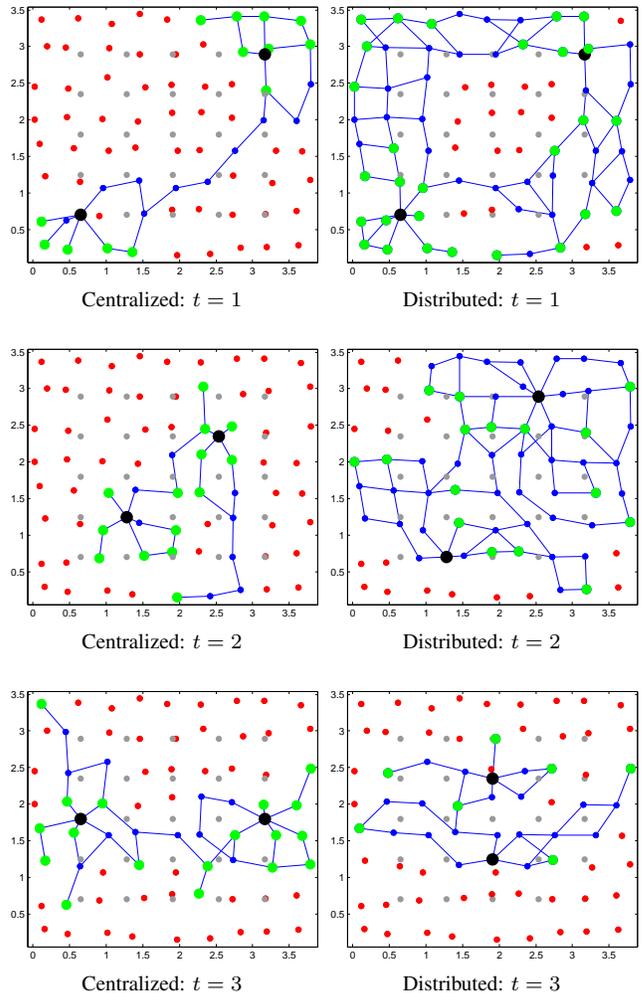


Fig. 6: Stage-by-stage illustration of the solutions obtained by the centralized MIP formulation, and the distributed convex relaxation scheme for problem (P1) with $K = 2$ mobile APs and a motion horizon of $T = 5$ steps. Only the first 3 time steps are depicted here; the 2 remaining steps are shown in Fig. 7. The red dots indicate inactive sources, the large green dots are the activated sources, while the small blue dots correspond to source nodes that operate only as relays. The black dots indicate the current location of the APs, the grey dots correspond to empty candidate AP locations, and the blue lines indicate the routing decisions T_{ij}^t for that time stage.

is set to $\tau = 1.8/q$, where the maximum degree q of the graph \mathcal{C} for this problem is $q = 10$. Note that the theoretically permissible values for the stepsize are $\tau \leq 1/q$, however, in most practical implementations this value can be multiplied by a factor of up to 2 to accelerate the convergence without compromising it; see also [26] for more examples. For these choices of parameters, all the relaxed binary variables $\mathbf{d}, \mathbf{p}, \mathbf{z}$ converged to values either 0 or 1 when the algorithm was terminated, as desired. In general, the proposed distributed method is fairly stable to the choice of ρ , i.e., the choice of ρ does not appear to significantly affect the convergence, as long as it is sufficiently large. For the stability parameters, values in the interval $[0.003, 0.01]$ appear to work best; smaller values can cause instability, while larger values tend to distort the 0 – 1 structure of the relaxed binary variables.

In order to gauge the convergence behavior of the proposed

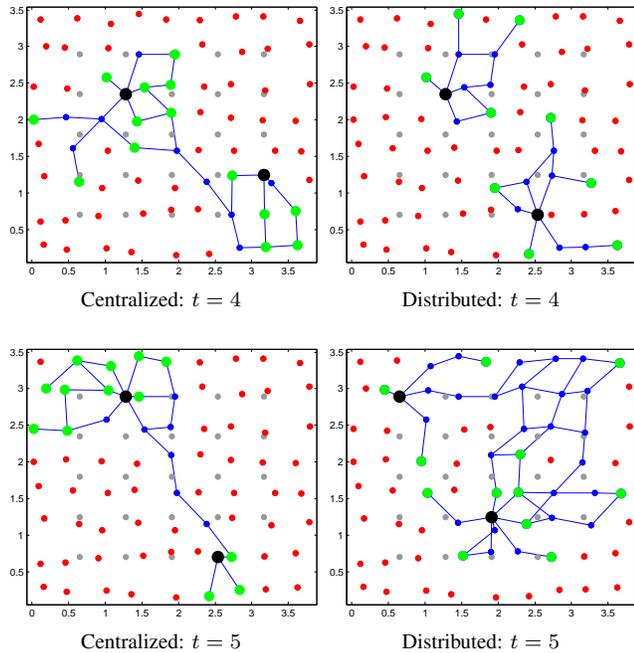


Fig. 7: Time steps 4 and 5 of the solutions obtained by the centralized MIP formulation, and the distributed convex relaxation scheme for problem (P1) with $K = 2$ mobile APs and a motion horizon of $T = 5$ steps.

distributed scheme, in Fig. 8 we plot the values of the most violated constraints for each of these three sets of constraints, i.e., we plot the max-norm of the constraint violations vector for each constraint set; recall also the discussion about the termination criteria towards the end of Section IV-B. We observe that the proposed distributed method can reach a sufficient accuracy level of 10^{-3} within around 1000 iterations for this particular problem. As a reference point, we mention that each iteration of ADAL for this particular problem, which involves solving the $J + M = 97$ local subproblems, requires approximately 3 seconds in MATLAB using the IBM CPLEX solver.

The proposed framework also allows us to address more realistic scenarios with more APs, obstacles, and spatially varying communication channels. The presence of obstacles is easily incorporated into the model by properly choosing the mobility graph \mathcal{G} for the APs. The channel variation is accounted for by appropriate pre-computation of the rate functions R_{ji} . In order to further examine the effectiveness of the proposed distributed algorithm in such applications, we next consider a problem with more mobile APs operating in the presence of wall obstacles. In particular, we consider a problem with $J = 65$ source nodes, $K = 3$ mobile APs, $M = 25$ candidate AP locations, $T = 4$ time stages motion horizon, and 2 walls that constrain the motion of the APs while also degrading the quality of communication channels that connect nodes on opposite sides of a wall. We refer to this problem setup as (P2); see Fig. 9. Similar to the first simulation, the link reliability parameters R_{ji} take the values 0 and 1, depending on the distance between the nodes j and i . However, now if a channel (the line between

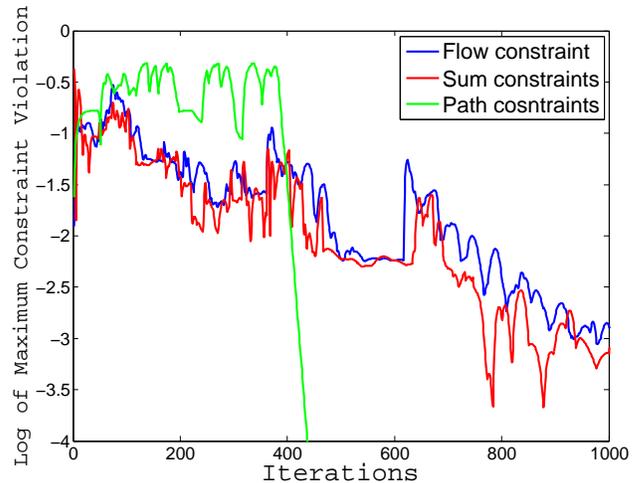


Fig. 8: Convergence behavior of the constraint violations for the routing flow (12), the routing sum (13), and the path (14) constraints of the problem setup depicted in Fig. 5-7. Plotted is the logarithm of the maximum constraint violation in absolute value, i.e., the maximum norm, for each set of constraints at every iteration.

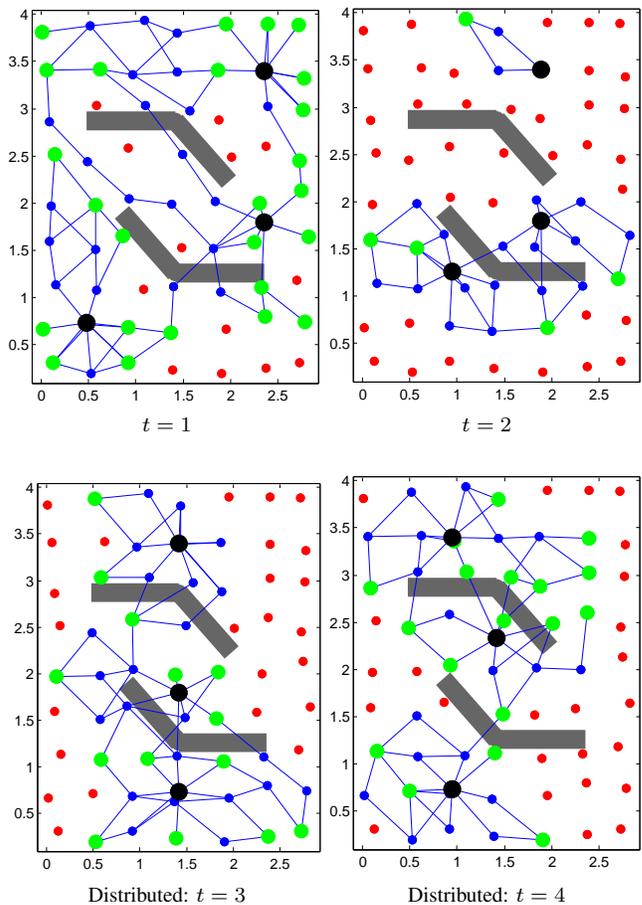


Fig. 9: Stage-by-stage illustration of the solution obtained by the proposed distributed scheme applied to problem (P2), which involves $K = 3$ mobile APs, two obstacles, and a motion horizon of $T = 4$ steps. The grey structures are walls that constrain the motion of the APs. The communication channels connecting nodes on opposite sides of a wall have reduced quality.

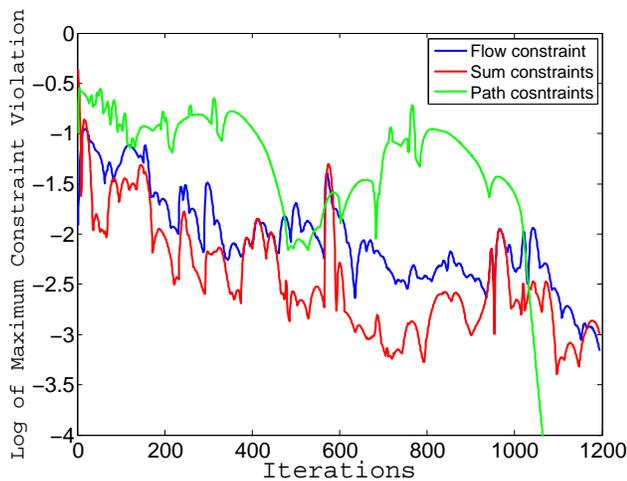


Fig. 10: Convergence behavior of the constraint violations for the routing flow (12), the routing sum (13), and the path (14) constraints for problem (P2) that includes obstacles. Depicted is the logarithm of the maximum constraint violation in absolute value, i.e., the maximum norm, for each set of constraints at every iteration.

two nodes) crosses the walls, its corresponding value of R_{ji} is reduced to 0.5. Moreover, for this scenario we also add the objective function $\sum_{j \in \mathcal{J}} f_j(r_j) = -\sum_{j \in \mathcal{J}} r_j$, which seeks for solutions that not only satisfy the minimum rate requirements, but also try to maximize the individual rates of each active source at the corresponding time stage. Fig. 9 and 10 illustrate the solution obtained by the proposed distributed scheme. We observe that the method is still able to reach a plausible solution.

VI. CONCLUSIONS

In this paper we introduced a novel framework that exploits mobility to enable reliable communications in wireless networks that do not possess the necessary resources to support the generated rates of information. Specifically, we defined the *connectivity scheduling* problem, a scheme that seeks to jointly optimize the motion of robotic APs and the communication decisions for all source nodes in a wireless network over a discrete-time horizon, so that the end-to-end integrity of communications is guaranteed over time. We initially formulated the problem using a Mixed Integer Programming approach, where all the integer variables are binary. Due to the inherent disadvantages of centralized formulations for network optimization problems, we also proposed a distributed algorithm to solve the problem at hand. Towards this goal, we first presented a convex continuous reformulation of the problem that utilizes the re-weighted ℓ_1 regularization scheme to recover the desired $0-1$ structure for the, now relaxed, binary variables. Then, we proposed the distributed solution to the problem by appropriately combining the re-weighted ℓ_1 approach with the Accelerated Distributed Augmented Lagrangians (ADAL) algorithm, a distributed optimization method that was recently developed by the authors. Numerical experiments were presented to validate the efficiency and effectiveness of the proposed distributed approach.

REFERENCES

- [1] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, January 2006.
- [2] M. C. DeGennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. 45th IEEE Conference on Decision and Control*, San Diego, CA, December 2006, pp. 3628–3633.
- [3] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 812–816, August 2007.
- [4] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," in *Decision and Control, 2007 46th IEEE Conference on*, Dec 2007, pp. 3591–3596.
- [5] K. Savla, G. Notarstefano, and F. Bullo, "Maintaining limited-range connectivity among second-order agents," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 187–205, 2009.
- [6] M. Ji and M. Egerstedt, "Coordination control of multi-agent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, August 2007.
- [7] M. Zavlanos, M. Egerstedt, and G. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, Sept 2011.
- [8] M. J. Neely, "Optimal peer-to-peer scheduling for mobile wireless networks with redundantly distributed data," *IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 2086–2099, 2014.
- [9] N. Enzai, S. Rais, and R. Darus, "An overview of scheduling algorithms in mobile ad-hoc networks," in *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on*, Dec 2010, pp. 120–124.
- [10] F. Hoffmann, D. Medina, and A. Wolisz, "Joint routing and scheduling in mobile aeronautical ad hoc networks," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 6, pp. 2700–2712, July 2013.
- [11] L. Ying and S. Shakkottai, "Scheduling in mobile ad hoc networks with topology and channel-state uncertainty," in *INFOCOM 2009, IEEE*, April 2009, pp. 2347–2355.
- [12] L. Chen, S. Low, M. Chiang, and J. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–13.
- [13] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, Apr. 2006.
- [14] M. Zavlanos, A. Ribeiro, and G. Pappas, "Network integrity in mobile robotic networks," *Automatic Control, IEEE Transactions on*, vol. 58, no. 1, pp. 3–18, Jan 2013.
- [15] M. M. Zavlanos, A. Ribeiro, and G. J. Pappas, "Mobility and routing control in networks of robots," in *Proc. 2010 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, December 2010, pp. 7545–7550.
- [16] A. Ghaffarkhah and Y. Mostofi, "Communication-aware motion planning in mobile networks," *Automatic Control, IEEE Transactions on*, vol. 56, no. 10, pp. 2478–2485, Oct 2011.
- [17] Y. Yan and Y. Mostofi, "Robotic router formation in realistic communication environments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 810–827, Aug 2012.
- [18] J. Le Ny, A. Ribeiro, and G. Pappas, "Adaptive communication-constrained deployment of unmanned vehicle systems," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 5, pp. 923–934, June 2012.
- [19] D.-T. Ho, E. I. Grotli, P. B. Sujit, T. A. Johansen, and J. B. Sousa, "Optimization of wireless sensor network and UAV data acquisition," *J. Intell. Robotics Syst.*, vol. 78, no. 1, pp. 159–179, Apr. 2015.
- [20] D. T. Ho, E. I. Grütli, and T. A. Johansen, "Heuristic algorithm and cooperative relay for energy efficient data collection with a UAV and WSN," in *Computing, Management and Telecommunications (ComMan-Tel), 2013 International Conference on*, Jan 2013, pp. 346–351.
- [21] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. S. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, "Underwater data collection using robotic sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 899–911, June 2012.
- [22] D. Bertsimas and R. Weismantel, *Optimization over integers*. Dynamic Ideas, 2005.
- [23] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

- [24] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Found. Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, Jan. 2012.
- [25] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.
- [26] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos, "An augmented Lagrangian method for distributed optimization," *Mathematical Programming*, vol. 152, pp. 405–434, August 2015.
- [27] N. Chatzipanagiotis and M. Zavlanos, "On the convergence rate of a distributed augmented lagrangian optimization algorithm," in *American Control Conference (ACC), 2015*, July 2015, pp. 541–546.
- [28] —, "A distributed algorithm for convex constrained optimization under noise," *IEEE Transactions on Automatic Control*, 2015, DOI: 10.1109/TAC.2015.2504932.
- [29] A. Ruszczyński, *Nonlinear Optimization*. Princeton, NJ, USA: Princeton University Press, 2006.
- [30] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
- [31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [32] D. Bickson, Y. Tock, A. Zymnis, S. Boyd, and D. Dolev, "Distributed large scale network utility maximization," in *Proceedings of the 2009 IEEE international conference on Symposium on Information theory*, 2009.
- [33] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.
- [34] M. Zargham, A. Ribeiro, A. Jadbabaie, and A. Ozdaglar, "Accelerated dual descent for network optimization," in *Proceedings of the American Control Conference*, San Francisco, CA, 2011.
- [35] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 2, pp. 221–229, April 2013.
- [36] —, "Gossip-based random projection algorithm for distributed optimization: Error bound," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 6874–6879.
- [37] Y. Nesterov, "Subgradient methods for huge-scale optimization problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 275–297, 2014.
- [38] D. Jakovetic, J. Freitas Xavier, and J. Moura, "Convergence rates of distributed nesterov-like gradient methods on random networks," *Signal Processing, IEEE Transactions on*, vol. 62, no. 4, pp. 868–882, Feb 2014.
- [39] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed adm via dual averaging," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 904–909.
- [40] A. Koppel, F. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 8292–8296.
- [41] S. S. Kia, J. Cortis, and S. Martinez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254 – 264, 2015.
- [42] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.
- [43] J. Mulvey and A. Ruszczyński, "A diagonal quadratic approximation method for large scale linear programs," *Operations Research Letters*, vol. 12, pp. 205–215, 1992.
- [44] A. Ribeiro, Z.-Q. Luo, N. D. Sidiropoulos, and G. B. Giannakis, "Modelling and optimization of stochastic routing for wireless multihop networks," in *Proc. 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Anchorage, Alaska, May 2007, pp. 1748–1756.
- [45] S. Shakkottai and R. Srikant, "Network optimization and control," *Found. Trends Netw.*, vol. 2, no. 3, pp. 271–379, Jan. 2007.
- [46] M. Chiang, "Nonconvex optimization for communication networks," in *Advances in Applied Mathematics and Global Optimization*, ser. Advances in Mechanics and Mathematics. Springer US, 2009, vol. 17, pp. 137–196.



Nikolaos Chatzipanagiotis received the Diploma in Mechanical Engineering, and the M.Sc. degree in Microsystems and Nanodevices from the National Technical University of Athens, Athens, Greece, in 2006 and 2008, respectively. He also received a Ph.D. degree in Mechanical Engineering from Duke University, Durham, NC, in 2015.

His research interests include optimization theory and algorithms with applications on networked control systems, wired and wireless communications, and multi-agent mobile robotic networks.



Michael M. Zavlanos received the Diploma in mechanical engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2002, and the M.S.E. and Ph.D. degrees in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, in 2005 and 2008, respectively.

From 2008 to 2009 he was a post-doctoral researcher in the department of electrical and systems engineering at the University of Pennsylvania, Philadelphia. He then joined the Stevens Institute of Technology, Hoboken, NJ, as an assistant professor of mechanical engineering, where he remained until 2012. Currently, he is an assistant professor of mechanical engineering and materials science at Duke University, Durham, NC. He also holds a secondary appointment in the department of electrical and computer engineering. His research interests include a wide range of topics in the emerging discipline of networked systems, with applications in robotic, sensor, communication, and biomolecular networks. He is particularly interested in hybrid solution techniques, on the interface of control theory, distributed optimization, estimation, and networking.

Dr. Zavlanos is a recipient of the 2014 Office of Naval Research Young Investigator Program (YIP) Award and the 2011 National Science Foundation Faculty Early Career Development (CAREER) Award. He was also a finalist for the best student paper award at CDC 2006.