Co-Optimization of Motion, Communication, and Sensing in Real Wireless Channel Environments via Monte Carlo Tree Search

Hong Cai, *Member, IEEE*, and Yasamin Mostofi, *Fellow, IEEE*

***Abstract*—We consider the problem where a robot navigates from a start position to a destination and needs to sense some sites along the way. The robot collects data when sensing each site and needs to transmit all collected data to a remote station by the end of its trip, as it moves along its path, under time/energy constraints, and while operating in real wireless fading environments. Our goal is to minimize the robot's total motion and communication energy costs by co-optimizing its path, data transmission along the path, and sensing decisions, under given constraints and while considering the stochastic wireless channel. In this paper, we show how to solve this co-optimization problem efficiently and with performance guarantees. More specifically, we formulate a specially-designed Markov Decision Process (MDP) and utilize Monte Carlo Tree Search (MCTS) to efficiently and optimally solve it. While the co-dependence of communication, sensing, and motion decisions makes this joint optimization challenging, we show that by considering the transmission optimization in the terminal reward and motion actions in the state transitions, we can iteratively optimize the sensing/motion and the communication parts in different stages of MCTS, in a way that allows us to equivalently solve the original co-optimization problem efficiently. We mathematically prove the convergence of our approach, characterize its convergence speed, and derive key properties of the optimum solution. We extensively evaluate our approach in realistic wireless environments where the channel experiences path loss, shadowing, and multi-path fading and is unknown to the robot.**

***Index Terms*—Communication and sensing, co-optimization of motion, optimization, robotics, sensor networks.**

I. Introduction

IN A ROBOTIC network, robots with limited local sensing, communication, and actuation capabilities interact with their environment and each other to perform given tasks [1]–[3]. Such networks can tremendously impact various areas, such as mobile service provisioning, search and rescue, surveillance, and extending cellular network coverage, among other possibilities. In order to properly design such networked robotic systems, it is important to jointly consider and optimize the robot's sensing, motion, and communications.

Traditionally, robotics and communications were studied separately in these systems. But in more recent years, researchers have started to look into communication-aware robotics, where realistic wireless communication models are explicitly taken into account for various robotic tasks, such as data relaying [4]–[7], cooperative transmission [8]–[10], robot-assisted wireless coverage [11]–[13], and data gathering [14]–[16]. These works, however, are mainly focused on the co-optimization of motion and communication, without considering the sensing aspect. Some other works, on the other hand, focus on the synergy between sensing and motion, without taking communications into account [17]–[22].

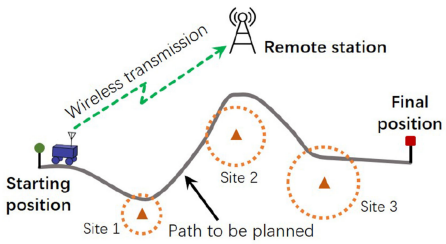
More related to this article are studies on the co-optimization of motion, communication, and sensing for data transmission from a field-sensing robot to a remote station in realistic channel environments and under resource constraints. Due to the complex spatial dynamics of a wireless channel and the challenges imposed by sensing and path planning, most existing works consider an ideal and known channel environment, and mainly focus on the sensing and motion aspects [23]–[25]. A small number of recent studies (including ours) attempt to consider realistic communication issues in this context but either proposed a heuristic approach with no performance guarantees for the generated trajectory and transmission policy [26], or mainly considered a predefined path to focus on a subset of sensing and communication issues [27].

In this article, we consider a general robotic task scenario that involves planning the robot's entire path, its data transmission along the path in a realistic and previously unknown channel environment, as well as its sensing of the field. More specifically, as illustrated in Fig. 1, the robot navigates from a start position to a given final destination, and needs to sense a number of sites in the field. For each site, the robot must move within a certain distance in order to sense it. When visiting each site, the robot collects new sensing data. The robot is required to transmit all its collected data (and possibly some initial data) to the remote station by the end of the trip, and while traversing the field. Its transmission energy cost is subject to a spatially varying and priorly unknown wireless communication channel that experiences fading. This captures several real-world robotic

Manuscript received 10 February 2021; revised 13 July 2021 and 14 December 2021; accepted 6 February 2022. Date of publication 15 March 2022; date of current version 19 September 2022. This work was supported by the National Science Foundation RI under Grant #2008449. Recommended by Associate Editor Lucia Pallottino. *(Corresponding author: Hong Cai.)*

The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: hcai@ece.ucsb.edu; ymostofi@ece.ucsb.edu).

Digital Object Identifier 10.1109/TCNS.2022.3158746



Co-Optimization of Motion, Comm., and Sensing in Wireless Fading Environments

Minimize:
- Total motion and communication energy cost

Subject to:
- Path and sensing constraints
- Data transmission constraints
- Operation time constraints

Variables:
- Communication rate/power
- Motion waypoints/path
- Sensing decisions

Fig. 1. (Left) Example of the robotic task scenario considered in this article. (Right) High-level description of the co-optimization problem. We are interested in solving this challenging problem efficiently and with performance guarantees.

applications. For instance, a field robot may be tasked with inspecting sites of interest in an environmental monitoring or a search and rescue mission, and needs to transmit the sensing data back to a remote center during the operation. In another example, a robot can be used to collect data from static sensors (with limited communication ranges) that are spread out over a field and it needs to communicate back the collected data as it traverses the area.

In general, this is a considerably challenging problem due to the coupling of sensing, communication, and motion decisions, the stochastic nature of real wireless channels, as well as the limited time/energy budget for the operation. In this article, we are interested in solving this problem efficiently and with performance guarantees. More specifically, we need to find the co-optimum motion, communication, and sensing decisions such that the total motion and communication energy cost is minimized and other system constraints are satisfied, while operating in realistic fading channel environments that are previously unknown to the robot. In order to tackle this considerably complex and combinatorial co-optimization problem, we propose a novel approach utilizing the Markov decision process (MDP) and Monte Carlo tree search (MCTS), and show how we can solve this problem efficiently and with theoretical convergence guarantees. MCTS, initially introduced for game-playing AI [28], is a solution approach for sequential decision-making problems, and has been utilized in other fields since its introduction. In recent years, it has also been utilized for some robotic applications, for instance, for viewpoint planning [20], legged locomotion [29], and pose estimation [30]. However, MCTS has not been used in the context of communication-aware robotics. But a direct formulation of MDP and applying MCTS will not work for our co-optimization problem of interest due to the coupled sensing-motion-communication action space and the associated complex constraints. In this article, we thus show how to properly translate our problem to a specially designed MDP and exploit MCTS to solve it efficiently and with performance guarantees. The following are the main contributions of this article.

1) We consider a complex robotic co-optimization problem, which involves sensing, data transmission, and path planning, in a realistic, priorly unknown wireless fading channel environment. Fig. 1 (right) shows a high-level description of the problem, which we are interested in solving efficiently and with performance guarantees. We propose a novel approach to solve this co-optimization problem by formulating a specially designed MDP and utilizing MCTS to efficiently and optimally solve it. As we shall see, by considering the transmission optimization in the MDP terminal reward evaluation and the motion actions in the state transitions, we can reduce the complexity of this problem and iteratively optimize the sensing/-motion and communication parts in different stages of MCTS, which enables us to efficiently solve the original challenging co-optimization problem. In order to address the stochastic and priorly unknown nature of the channel, we use a probabilistic channel prediction framework and show how it can be incorporated in our proposed MCTS-based approach.
2) We mathematically characterize an upper bound on the probability that our proposed approach does not result in the optimum decision. We show that our algorithm converges to the optimum as the number of iterations increases and provides a bound on the convergence speed. We further mathematically characterize the properties of the optimum solution.
3) Using a realistic 2-D wireless channel environment, we thoroughly validate the performance of our proposed approach. We further compare our approach with a benchmark method that separately optimizes motion and communication, and show that our approach significantly outperforms it. For instance, based on an evaluation of 50 problem instances, our solution uses 55% less total energy on average. Finally, we compare it with the most related state-of-the-art work in this area.

The rest of this article is organized as follows. Section II provides the preliminaries. In Section III, we formulate the co-optimization problem and in Section IV, we discuss how to transform the co-optimization into an MDP and utilize MCTS to solve it. In Section V, we prove the convergence and show the theoretical properties of our proposed approach. In Section VI, we validate our approach in a realistic 2-D wireless channel environment. Finally, Section VII concludes this article.

II. Preliminaries

In this section, we summarize the communication and motion energy models to be used in this article. We further summarize how wireless channels can be predicted in real channel environments. We also provide an overview of MDP [31] and MCTS [32].

A. Channel Prediction and Communication Energy Model

Consider the case where the robot adopts the commonly used M-ary quadrature amplitude modulation (MQAM) for transmission [33].[1] As shown in [33], the required transmission power can be well approximated by $\Gamma_{\mathrm{T}} = (2^r - 1)\ln(5p_{\mathrm{BER}})/(-1.5\gamma)$, where r is the spectral efficiency (i.e., transmission rate divided by bandwidth, in b/s/Hz), p_{BER} is the required bit error rate (BER), and γ is the received channel-to-noise ratio (CNR). During operation, the robot needs to assess its transmission power along any given unvisited path for planning purposes. This requires it to predict the channel at unvisited locations over the space, based on a few channel samples. Due to the real-world wireless propagation effects of path loss, shadowing, and multipath fading, the CNR is best modeled as a spatial stochastic process. Then, given a small number of prior channel samples in the same environment and based on [34], the CNR (in dB) at an unvisited location q, $\Upsilon_{\mathrm{dB}}(q)$, can be best modeled by a Gaussian random variable, with its expectation and variance given by

$$\begin{aligned}\mathbb{E}[\Upsilon_{\mathrm{dB}}(q)] &= H_q\hat{\theta} + \Psi^T(q)\Phi^{-1}(Y - H_Q\hat{\theta})\\ \Sigma(q) &= \hat{\alpha}^2 + \hat{\sigma}^2 - \Psi^T(q)\Phi^{-1}\Psi(q)\end{aligned} \tag{1}$$

where $Y = [y_1, \ldots, y_m]^T$ are the m priorly collected CNR measurements (in dB); $Q = [q_1, \ldots, q_m]$ are the measurement locations; $\hat{\theta}$, $\hat{\alpha}$, $\hat{\beta}$, and $\hat{\sigma}$ are the estimated channel parameters; $H_q = [1 \quad -10\log_{10}(\|q - q_b\|)]$, $H_Q = [H_{q_1}^T, \ldots, H_{q_m}^T]^T$; $\Psi(q) = [\hat{\alpha}^2\exp(-\|q - q_1\|/\hat{\beta}), \ldots, \hat{\alpha}^2\exp(-\|q - q_m\|/\hat{\beta})]^T$; and $\Phi = \Omega + \hat{\sigma}^2 I_m$ with $[\Omega]_{i,j} = \hat{\alpha}^2\exp(-\|q_i - q_j\|/\hat{\beta})$ $\forall i, j \in \{1, \ldots, m\}$ and I_m denoting the identity matrix.

This formulation allows the robot to predict the channel quality at any unvisited locations, based on a small number of prior channel measurements in the same environment.[2] See [34] for more details and the performance of this channel predictor in different real environments.

Based on this channel prediction, the expected required transmission power at location q is given by $\mathbb{E}[\Gamma_T(q)] = (2^r - 1)\mathbb{E}[1/\Upsilon(q)]/Z$, where $\Upsilon(q)$ is the predicted CNR at location q, $\mathbb{E}[1/\Upsilon(q)]$ can be evaluated given the log-normal distribution of $\Upsilon(q)$ (or Gaussian in dB), and $Z = -1.5/\ln(5p_{\mathrm{BER}})$. Given a transmission time duration, the robot can then predict the communication energy cost for transmitting from any unvisited location to the remote station.

[1]MQAM represents a broad family of standard digital modulation methods, including the one used in 802.11 Wi-Fi standards.

[2]It should be noted that the robot needs to predict the uplink channel for the purpose of its sensing-path-communication co-optimization, as we shall see. Thus, the small number of prior channel measurements can be collected by the remote station and transmitted back to the robot which will be in charge of channel prediction over the space. Alternatively, if time division duplex is used, the robot can directly use a small number of downlink channel samples in order to predict the channel elsewhere.

B. Motion Energy Model

Based on experimental studies, a mobile robot's motion power can be modeled by a linear function of its speed for a number of platforms [35]: $\Gamma_{\mathrm{M}} = \kappa_1 u + \kappa_2$ when $0 < u \leq u_{\max}$, and $\Gamma_{\mathrm{M}} = 0$ when $u = 0$, where u and $u_{\max}$ are the robot's speed and maximum speed, respectively, and κ_1 and κ_2 are positive constants determined by the robot's motor, mechanical transmission system, and external load. Suppose that the robot travels at a constant speed u_{const}. The motion energy cost for a travel distance of l is then given by $\mathscr{E}_m = (\kappa_1 + \kappa_2/u_{\mathrm{const}})l$.

C. Markov Decision Process

An MDP is a mathematical framework for modeling a discrete-time sequential decision-making process. Suppose that a decision-making agent is in some state. The goal of MDP is to find the optimum action to take in this state. An MDP is defined by four components as follows.

1) State: There is a finite set of states S, where each $s \in S$ represents a state that the decision-making agent can be in. When the agent reaches a terminal state, it finishes the decision-making process.
2) Action: There is a finite set of actions A, where each $a \in A$ is a feasible action that the agent can take. The action set can be state dependent, i.e., for each state s, the set of feasible actions A_s is different.
3) Transition: $\zeta(s, a, s')$ is the probability that by taking action a in state s, the agent moves to state s'. The MDP is deterministic if there is no randomness in the state transitions.
4) Reward: $w(s, a, s')$ is the immediate reward that the agent receives after transitioning from s to s', due to action a. There can be a terminal reward $w_T(s_T)$ when the agent reaches a terminal state s_T.

In this article, we are interested in finite-horizon, deterministic, undiscounted MDPs, where the core problem is to find a decision policy $\pi: S \to A$ that maximizes the cumulative reward of a state s, which is given by

$$X_s = \sum_{t=t_s}^{T-1} w(s_t, \pi(s_t), s_{t+1}) + w_T(s_T) \tag{2}$$

where t_s is the time instance when agent is in state s and T is the time step that the agent reaches a terminal state.

Given an initial state s_1, the optimization for deriving the optimum decision policy can be written as follows:

$$\max_{\pi(s)} \sum_{t=1}^{T-1} w(s_t, \pi(s_t), s_{t+1}) + w_T(s_T) \tag{3}$$

where we maximize the cumulative reward of s_1, over the space of all possible decision policies.

D. Monte Carlo Tree Search

MCTS is a popular algorithm that can be used to efficiently solve finite-horizon, large-scale MDPs. In this overview, we

assume that the MDP is finite-horizon, deterministic, and undiscounted, which is the most relevant form for the problem-of-interest in this article.

As discussed in Section II-C, the goal of the MDP is to find a policy $\pi(s)$ that maximizes the cumulative reward. Denote the optimum cumulative reward of s as $X_s^\star$. The optimum policy can then be recursively written as

$$\pi^\star(s) = \underset{a\in A_s}{\text{argmax}}\, w(s,\, a,\, s') + X_{s'}^\star. \tag{4}$$

In theory, traditional methods like value/policy iteration can be used to find the optimum policy by solving for the optimum actions for all states. However, they quickly become intractable for MDPs with a large state space. On the other hand, MCTS is an online algorithm that computes the optimum action for the current state, rather than for all states. Specifically, it constructs a tree to trace the future states that could be reached from the current state and biases the computation (i.e., Monte Carlo simulations) toward states that are more likely to produce large cumulative rewards. Meanwhile, it also methodically explores states with fewer Monte Carlo samples. In this way, MCTS not only continuously refines the cumulative reward estimates of the promising states, but reduces the chance of missing a good state due to insufficient Monte Carlo sampling. By adopting a proper sampling rule [e.g., using the upper confidence bound (UCB) of (5)] that balances such exploitation and exploration, the expectation of the cumulative reward estimate of a state converges to the optimum efficiently and the probability of failing to choose the optimum action also converges to zero rapidly. These properties make it computationally favorable to apply MCTS to large MDPs for online decision-making.

Next, we briefly explain how MCTS works.[3] It starts with a tree containing only the root node τ_1 that represents the initial state s_1. It then iteratively grows the tree, where each node uniquely represents one of the MDP states and each edge represents a state transition.[4] For each state s included in the tree, MCTS maintains a visit count, n_s, and an estimate of the optimum cumulative reward after n_s visits, $\bar{X}_{s,\,n_s}$.[5] In each MCTS iteration, there are four main stages as follows:

1) Selection: In each iteration, MCTS performs a tree traversal from the root node τ_1. During the traversal, at each node (i.e., a state of the MDP), MCTS selects an action according to some selection policy, which leads to the next node. It continues this traversal until the selected action leads to a state s_{next} that does not have a corresponding node in the current tree. Denote the last traversed node in the current tree as τ_{last}.
2) Expansion: We add a new node τ_{next} representing s_{next} to the tree, as a child node of τ_{last}.
3) Simulation: Starting from s_{next}, random feasible actions are taken in the MDP until this process reaches a terminal state.
4) Back up: Upon reaching a terminal state, we can calculate the cumulative reward for any state s involved in the tree traversal of the current iteration: $X_{s,\,n_s+1} = \sum_{t=t_s}^{T-1} w(s_t,\, a_t,\, s_{t+1}) + w_T(s_T)$, where t_s is the time step when this traversal is in state s, T is the time when it reaches the terminal state, and n_s records the number of previous iterations where s was part of the traversal. Then, for each s in the current traversal, we update the estimate of the optimum cumulative reward as follows: $\bar{X}_{s,\,n_s+1} = (n_s \bar{X}_{s,\,n_s} + X_{s,\,n_s+1})/(n_s+1)$, where $\bar{X}_{s,\,n_s}$ is the average value of the cumulative rewards of s seen in the past n_s iterations where s was part of the traversal and $\bar{X}_{s,\,n_s+1}$ is the updated average cumulative reward. We also increment the visit count n_s by 1.

These four steps are repeated until the number of MCTS iterations reach a predefined computation budget. The agent then takes an action according to this decision policy: $a^\pi = \text{argmax}_{a\in A_{s_1}}\, w(s_1,\, a,\, s') + \bar{X}_{s',\,n_{s'}}$, and transitions to the next state s_2. In order to determine the action to take in s_2, the agent performs a new batch of MCTS iterations, with s_2 being the new initial state. This online planning process ends when the agent arrives at a terminal state.

In order to ensure efficient convergence, it is necessary to choose a proper selection policy while traversing down the tree in the first MCTS step. A common choice is based on the UCB [28]

$$a^\dagger = \underset{a\in A_s}{\text{argmax}}\; w(s,\, a,\, s') + \bar{X}_{s',\,n_{s'}} + \sqrt{2\ln(n_s)/n_{s'}} \tag{5}$$

where action $a^\dagger$ leads the traversal to the next tree level.[6]

It can be seen from (5) that the first two terms bias MCTS to select an action that is estimated to produce a large cumulative reward, while the third term allows MCTS to explore an action that leads to a less-visited state. As we shall see, using such a tree traversal strategy will result in an efficient convergence for our co-optimization problem.

III. Co-optimization Problem Formulation

Consider the scenario where a robot travels from a start position p_s to a given destination p_f in a wireless channel environment, as shown in Fig. 1. During the trip, the robot needs to sense V sites in the field. In order to sense a site j, the robot must be within a sensing range Δ_j of it, in order to collect D_j bits of sensing data from this site. It also carries D_0 bits of initial data when starting the operation. The robot is required to transmit all of the collected and initial data to the remote station by the end of the trip, while minimizing its total motion and communication energy usage.

[3]When describing the MCTS process, we use "node" to refer to a tree node in the search tree and use "state" for a state of the corresponding MDP. We refer the readers to [32] for a comprehensive survey of MCTS.

[4]We assume a one-to-one mapping from tree nodes to states. This is true in our problem, since we will use the robot's path history as the state. It should, however, be noted that not all states of the MDP are represented in the tree.

[5]$\bar{X}_{s,\,n_s}$ is set to 0 initially when there is no visit to state s yet (i.e., $n_s = 0$).

[6]Note that the third term in (5) can be infinity when $n_{s'} = 0$. When there is only one such s', the policy chooses the action that leads to this s'. When there are multiple such next states, then it is common to simply choose one of the corresponding actions randomly. When $n_s = 1$ and $n_{s'} = 0$, the third term is also taken as infinity.

In the optimization formulation, we consider a discretized 2-D workspace consisting of regular grids. The robot's path is defined by a sequence of waypoints, each being the center of its corresponding grid. Two consecutive waypoints belong to two neighboring grid locations in the environment and determine a step in the path. Let $P = [p_1, .., p_K]$ denote a path from the start position $p_1 = p_s$ to the final position $p_K = p_f$. Let $\tilde{p}_1, \ldots, \tilde{p}_V$ denote the locations of the V sites to be sensed.[7] The robot's path is subject to a total operation time budget: $\mathscr{T}(P) \leq T$, where $\mathscr{T}(P)$ is the total time needed to traverse the path P and T is the time budget. Let $R = [r_1, \ldots, r_K]$ denote the respective spectral efficiencies to be used at the corresponding waypoints $[p_1, \ldots, p_K]$.[8] The communication bandwidth B is constant during the operation and we denote $\tilde{D}_j = D_j/B \, \forall j \in \{0, \ldots, V\}$. The transmission time duration for each waypoint is t_c. The robot uses a constant speed u_{const} when moving. The motion energy cost of traversing the path P is $\mathscr{E}_m(P)$ and the communication energy cost is denoted by $\mathscr{E}_c(P, R; \Upsilon)$, where Υ is the predicted channel over the space and $\Upsilon(q)$ is the predicted channel at location q, based on the channel prediction framework of Section II-A. The co-optimization problem can then be formulated as follows:

$$
\begin{aligned}
\min_{P,R} \quad & \mathscr{E}_m(P) + \mathscr{E}_c(P, R; \Upsilon) \\
\text{s.t.} \quad & (1) \ \min\nolimits_{i\in\{1,\ldots,K\}} \|p_i - \tilde{p}_j\| \leq \Delta_j \quad \forall j \in \{1, \ldots, V\} \\
& (2) \ \mathscr{T}(P) \leq T \\
& (3) \ p_1 = p_s, \ p_K = p_f \\
& (4) \ \sum_{i=1}^{K} r_i t_c = \sum_{j=0}^{V} \tilde{D}_j \\
& (5) \ \sum_{i=1}^{k_j} r_i t_c \leq \sum_{z=0}^{j-1} \tilde{D}_z \quad \forall j \in \{1, \ldots, V\} \\
& (6) \ 0 \leq r_i \leq r_{\max} \quad \forall i \in \{1, \ldots, K\}
\end{aligned}
\tag{6}
$$

where the objective function is the total motion and communication energy cost, with the path and the spectral efficiencies to be optimized. Constraints (1)–(3) are related to the motion. Constraint (1) ensures that the robot moves within the sensing range for each site at some point in the path. Constraint (2) ensures that the total travel time does not exceed the time budget. Constraint (3) ensures that the path is from the start position to the destination. The remaining constraints are related to the transmission. Constraint (4) enforces the robot to transmit all the data by the end of the trip. Constraint (5) ensures a valid transmission plan, i.e., the data are transmitted only after they have been collected. In this constraint, $k_j = \min\{i \in \{1, \ldots, K\} \mid \|p_i - \tilde{p}_j\| \leq \Delta_j\} \, \forall j \in \{1, \ldots, V\}$, which is the index of the first waypoint within the sensing range for site j. In other words, D_j are available after the robot has moved to p_{k_j}. Constraint (6) provides the feasible range for the spectral efficiencies. It should be noted that the robot minimizes the expectation of its communication energy cost in the objective since the predicted channel over the space is a multivariate random variable.

Due to the combinatorial nature of path planning and the coupling between motion and communication, problem (6) is a nonlinear, nonconvex, and combinatorial optimization problem, which is challenging to solve. Let us consider a simplified version first, where the robot's path $P = [p_1, \ldots, p_K]$ is already given and we only need to optimize the spectral efficiencies. Based on the channel prediction model of Section II-A, this subproblem can be written as follows:

$$
\begin{aligned}
\min_{R} \quad & \sum_{i=1}^{K} \frac{(2^{r_i} - 1)}{Z} \mathbb{E}\left[\frac{1}{\Upsilon(p_i)}\right] \cdot t_c \\
\text{s.t.} \quad & (1) \ \sum_{i=1}^{K} r_i t_c = \sum_{j=0}^{V} \tilde{D}_j \\
& (2) \ \sum_{i=1}^{k_j} r_i t_c \leq \sum_{z=0}^{j-1} \tilde{D}_z \quad \forall j \in \{1, \ldots, V\} \\
& (3) \ 0 \leq r_i \leq r_{\max} \quad \forall i \in \{1, \ldots, K\}
\end{aligned}
\tag{7}
$$

where the objective function is the communication energy cost and the communication-related constraints are the same as in problem (6). Since the entire path P is given, $\mathbb{E}[1/\Upsilon(p_i)]$ can be evaluated based on the predicted channel at location p_i. It can be seen that this is a convex optimization problem which can be solved very efficiently, as the objective function is convex and the constraints are linear [36].

This is an important observation that motivates our proposed approach to solve the co-optimization problem. If we can design an algorithm where the sensing/motion part and the transmission part are computed iteratively in different steps, then we can potentially make the problem tractable, as the decision space is considerably simpler when we only consider sensing/motion and the convex transmission subproblem can be solved efficiently. In the next section, we show how we can indeed achieve this by properly formulating an MDP for the co-optimization problem (6) and utilizing MCTS to solve it.

IV. Solving the Co-optimization Problem via Monte Carlo Tree Search

The co-optimization problem (6) can be seen as a sequential decision-making problem, where at each step, the robot decides on a motion action and a transmission action such that its total energy cost is minimized. (MDP, thus, provides a suitable framework for modeling this problem. In order to formulate a proper MDP (i.e., satisfying the Markov property), a state should not only include the robot's current waypoint, but also

[7]The order of visit for the sites is dictated by their order on the shortest path from the start position to the destination through these sites. This is the optimum order as long as no two sites are too close to each other.

[8]It should be noted that we do not consider quantization of the rates in our formulation. Our work can be easily extended to include this, by using the corresponding literature in the area of communications.

its past waypoints. This creates an exponentially large state space, making it infeasible to use traditional methods such as value/policy iteration. By utilizing MCT, on the other hand, we can efficiently handle the large state space. However, the coupled sensing-motion-transmission action space still presents a challenge since we cannot determine the optimum spectral efficiencies when the path is not fully determined, given the complex communication constraints of problem (6).

In order to resolve this issue, we propose a specially designed MDP where we do not need to consider the transmission actions in the state transitions. More specifically, we only consider the motion actions in the state transitions, while the spectral efficiencies are optimized when evaluating the terminal reward in a terminal state. This allows us to iteratively optimize over the sensing-motion and the communication parts in two different stages of our proposed MCTS, and get around the challenge of determining the optimal spectral efficiencies without a complete path. Note that while the sensing-motion and the communication parts are handled in two different stages, they are still jointly optimized within the same MDP framework. As we shall see in this section, we can theoretically establish the equivalence between this MDP and the original co-optimization problem (6). We next describe in details how we formulate this MDP.

Algorithm 1: MCTS-based Solution to Problem (6).

Inputs: Initial and final positions: p_s and p_f, operation time budget: T, maximum spectral efficiency: $r_{\max}$, initial data load: $\tilde{D}_0$, and location, sensing range, and sensing data for each site: $\tilde{p}_j$, Δ_j, $\tilde{D}_j$, $\forall j \in \{1, ..., V\}$.

The robot's initial state is given by $s = [p_s]$.

while *The robot has not reached* p_f **do**

  Initialize the iteration count: $i = 1$.

  while $i \leq N_I$ **do**

    Given the current state s, perform the steps of selection, expansion, simulation, terminal reward evaluation, and back up, as described in Sec. IV-B.

  end

  Update the robot's state based on Eq. 12.

end

Extract the complete path from p_s to p_f, $P^\star$, from the state.

Given $P^\star$, solve for the optimum spectral efficiencies $R^\star$ (via problem (7)).

Return $P^\star$ and $R^\star$.

A. Co-optimization as a Markov Decision Process

The optimization problem (6) can be written as a deterministic, finite-horizon, and undiscounted MDP as follows.

State: Let $P' = [p_1, \ldots, p_k]$ denote a partial path, where $p_1 = p_s$ and p_k may not have reached p_f. P' is the first part of a complete path from p_s to p_f. For instance, a partial path of $[p_1, p_2, p_3]$ indicates that the robot starts at $p_1 = p_s$ and will move to p_2 and p_3 sequentially, with the remaining path not yet determined. A partial path P' then represents a state in this MDP. A state is a terminal state if the corresponding partial path has reached the destination, i.e., $p_k = p_f$.

Action: Given a state in this MDP with $P' = [p_1, \ldots, p_k]$, an action is represented by a next location that the robot can move to. The feasible set of actions is the set of those neighboring locations of p_k, after moving to which the robot can still reach the destination while satisfying the motion constraints on the entire path, i.e., constraints (1)–(3) of problem (6).

Transition: Given a feasible action, the next state is obtained by appending this location to the current partial path. For instance, given a current state of $[p_1]$ and an action of p_2, the next state is given by $[p_1, p_2]$. There is no randomness in moving from one location to another.

Reward: We take the reward to be zero for any state transition. The terminal reward is then taken as: $w_T(P) = -(\mathscr{E}_m(P) + \min_{R\in\Omega_R} \mathscr{E}_c(P, R; \Upsilon))$, where P is the complete path from p_s to p_f associated with the terminal state and Ω_R is the feasible set of spectral efficiencies as defined by constraints (1)–(3) in problem (7). The first term (in parentheses) is the motion energy cost of traversing P and the second term is the communication energy cost given by solving problem (7), which is taken as infinity if problem (7) is infeasible given P. $-w_T(P)$ is then the minimum total energy cost given that the robot would use the full path P.

It can be seen that we have moved the transmission optimization into the terminal reward computation and only the motion part is considered in the MDP actions. Since the MDP's reward function captures the objective function of problem (6) and its actions conform to the constraints of problem (6), solving this MDP provides the optimum solution to problem (6). We formally show this in the next Proposition.

Proposition 1: By solving the MDP of Section IV-A with the initial state $s_1 = [p_s]$, we obtain the optimum solution to the original co-optimization problem (6).

Proof: As the MDP is undiscounted and there is no reward for any intermediate step, the cumulative reward of s_1 is equal to the terminal reward: $w_T(P) = -(\mathscr{E}_m(P) + \min_{R\in\Omega_R} \mathscr{E}_c(P, R; \Upsilon))$. By maximizing the cumulative reward of s_1, we have the following optimization problem:

$$\max_{p_2, \ldots, p_K} \; -(\mathscr{E}_m(P) + \min\nolimits_{R\in\Omega_R} \mathscr{E}_c(P, R; \Upsilon)) \tag{8}$$

where we optimize over the actions $p_2, \ldots, p_K$, which also need to be feasible, i.e., the complete path $P = [p_1, \ldots, p_K]$ needs to satisfy constraints (1)–(3) of problem (6).

The solution to the MDP maximizes the negative value of the total energy cost [i.e., minimizes the objective function in problem (6)], and the resulting motion actions and spectral efficiencies need to satisfy their respective constraints in the original problem (6). As such, the path $P^\star$ and the spectral efficiencies $R^\star$ obtained by solving the MDP of Section IV-A are also the optimum solution to problem (6). ■

B. Solution via Monte Carlo Tree Search

The MDP formulation of Section IV-A facilitates applying MCTS. First, in order to solve problem (6), we do not need to derive the optimum decisions for all the MDP states, as many of them are suboptimal and thus irrelevant to the original optimization problem. Instead, MCTS provides a methodical

framework to bias the computation toward promising states. Second, MCTS is well suited for the structure of the MDP. In each iteration, it only needs to perform the transmission optimization at the end of the simulation stage, when the simulation reaches a terminal state and the path becomes complete. This avoids the difficulty of determining the optimal spectral efficiencies without a complete path. Lastly, the convex optimization-based terminal reward evaluation [using problem (7)] allows for fast Monte Carlo simulations. Next, we describe in details how to utilize MCTS to solve this MDP.

1) Selection Policy: As discussed in Section II-D, we keep track of the average cumulative reward $\bar{X}_{s,n_s}$ and a visit count n_s for each state s during the MCTS iterations. These quantities are used in the UCB-based selection policy for selecting the next state to move to during the tree traversal stage

$$s^\dagger = \underset{s' \in C_s}{\operatorname{argmax}} \ \bar{X}_{s',n_{s'}} + \sqrt{2\ln(n_s)/n_{s'}} \tag{9}$$

where $s^\dagger$ is the next state to move to and C_s is the set of next states reachable from s via a feasible action. Note that we select from the next states instead of the actions, since there is no intermediate reward and the transition is deterministic.

2) Expansion and Simulation: When adding a new node to the tree, the expansion part follows from the standard MCTS. In the simulation stage, a random feasible action (i.e., a random next waypoint) is taken until the path reaches the destination. More specifically, given a state with partial path $P' = [p_1, \ldots, p_k]$, the set of feasible next locations is the set of the neighboring locations of p_k, after moving to which the robot can visit the unsensed sites and reach the final position within the remaining time budget [i.e., satisfying constraints (1)–(3) of problem (6)]. Denote the next site to sense as $\tilde{p}_{j'}$. We can obtain a random feasible next location by sampling from the set: $\Omega_{\text{next}} = \{p_{k+1} \mid \|\tilde{p}_{j'} - p_{k+1}\|_{\mathscr{T}} + \sum_{j=j'+1}^{V} \|\tilde{p}_j - \tilde{p}_{j-1}\|_{\mathscr{T}} + \|p_f - \tilde{p}_V\|_{\mathscr{T}} \leq T - \mathscr{T}(P') - \|p_{k+1} - p_k\|_{\mathscr{T}} \quad \text{and} \quad p_{k+1} \in \Omega_{p_k}\}$, where $\|p_i - p_j\|_{\mathscr{T}}$ is the minimum time needed to move from p_i to p_j (on the grids) and Ω_{p_k} is the set of neighboring locations of p_k. It can be seen that any location from Ω_{next} allows the robot to visit the exact location of each unsensed site and then reach the destination within the remaining time, i.e., any location from this set is a feasible next location.[9]

3) Reward Evaluation and Back Up: The simulation ends when it encounters a terminal state, i.e., the partial path reaches the destination. In order to facilitate the convergence of MCTS, we need to transform the terminal reward $w_T(P)$ into $[0, 1]$ [see (8) for the details of $w_T(P)$] [28]. To do this, we utilize a simple baseline strategy where the robot moves along straight lines, sequentially from the start position to each exact site location and finally to the destination. We denote the negative total energy cost of this straight-path baseline as w_{sp}. We then take $\tilde{w}_T(P) = \max(0, 1 - w_T(P)/w_{sp})$ as the transformed reward, which is in $[0, 1]$. $\tilde{w}_T(P)$ indicates the percentage total energy cost reduction over this baseline and is zero if the total cost resulted from P is larger than that of the baseline.

The transformed reward $\tilde{w}_T(P)$ is then backed up. For each state s that is part of the current tree traversal, its average cumulative reward is updated as follows:

$$\bar{X}_{s,n_s+1} = (\tilde{w}_T(P) + n_s \bar{X}_{s,n_s})/(n_s + 1) \tag{10}$$

and the visit count n_s is incremented by 1. In addition to this standard MCTS back-up procedure, for each s, we also record the maximum cumulative reward seen so far, as follows:

$$\widehat{X}_{s,n_s+1} = \max\{\widehat{X}_{s,n_s}, \tilde{w}_T(P)\} \tag{11}$$

where $\widehat{X}_{s,n_s} = 0$ initially when $n_s = 0$.

4) Solution Extraction: Given the initial state s, we perform N_I MCTS iterations (N_I dictated by the computation budget). As there is no randomness in the rewards of this MDP, $\widehat{X}_{s,n_s}$ records the best solution (i.e., the minimum total energy cost) seen so far given that the robot would traverse the partial path P'_s. Therefore, instead of using $\bar{X}_{s,n_s}$ as in standard MCTS, we use $\widehat{X}_{s,n_s}$ to decide the next location that the robot should move to

$$s^\pi = \underset{s' \in C_s}{\operatorname{argmax}} \ \widehat{X}_{s',n_{s'}} \tag{12}$$

where the last waypoint in the partial path of s^π is the next location for the robot to move to and C_s is the set of next states reachable from s via a feasible action.

We then set s^π as the new initial state and perform another N_I iterations, after which we select the best next state. This process ends when we reach a terminal state, where we obtain the best complete path, $P^\star$, and we can solve for the optimum spectral efficiencies, $R^\star$. $P^\star$ and $R^\star$ are then the solution to problem (6) given by our proposed approach. Our proposed algorithm is summarized in Algorithm 1.

Remark 1: (Applicability to other modulation schemes): While we assume a general MQAM model in our derivations, our proposed approach is applicable to other modulation techniques as long as the transmission power can be written as a function of the transmission rate and the location-dependent channel quality [e.g., in the objective function of problem (7)], which is used for terminal reward evaluation in our MCTS.

V. Theoretical Analysis

In this part, we mathematically prove the convergence and characterize the convergence speed of our proposed approach. We further characterize properties of the optimum solution.

A. Convergence and Optimality

Since we have shown the equivalence between our MDP formulation and the original co-optimization problem (6) (in Proposition 1), our convergence and optimality analysis in this part directly establishes the performance guarantees of our MCTS-based solution for problem (6).

[9]Note that this set does not contain all the possible feasible next locations, as there may exist a feasible next location that requires the robot to not visit the exact locations of the unsensed sites in order to reach the destination within the remaining time. However, finding such a feasible next location requires solving an optimization problem similar to the shortest-path traveling salesman problem with neighborhoods [37], which is computationally expensive for the simulation stage. On the other hand, sampling a point from Ω_{next} is quick, and as we shall see in Section VI, allows us to efficiently obtain near-optimal solutions to the complex co-optimization problem (6).

The first result shows the convergence of the expected average cumulative reward of a state.[10] It should be noted that $X_s^\star$ is the optimum cumulative reward of state s in the following theorem, as defined in Section II-D.

Theorem 1: When MCTS with the UCB selection policy is applied to the MDP of Section IV-A, for any state s, the bias of $\bar{X}_{s,n_s}$ is $\mathrm{O}(\ln(n_s)/n_s)$, i.e., $\|\mathbb{E}[\bar{X}_{s,n_s}] - X_s^\star\| = \mathrm{O}(\ln(n_s)/n_s)$. Moreover, $X_s^\star = \max_{s'\in C_s} X_{s'}^\star$, where C_s is the set of next states reachable from state s via a feasible action, which corresponds to the set of child nodes of the node of state s in the tree.

Proof: Note that the immediate reward is zero for any state transition. The results then follow directly from [28, Ths. 2 and 6]. ■

Theorem 1 shows that the bias of the average cumulative reward of a state s is $\mathrm{O}(\ln(n_s)/n_s)$. Moreover, the expected average cumulative reward converges to the optimum cumulative reward of the optimum next state. By the optimum next state of a state s, we mean the next state $s^\star$ which has the maximum optimum cumulative reward, i.e., $s^\star = \mathrm{argmax}_{s'\in C_s} X_{s'}^\star$.

Next, we explicitly derive what the optimum cumulative reward is for each state by using induction.

Theorem 2: When MCTS with the UCB selection policy is applied to the MDP of Section IV-A, for state s, $X_s^\star = \max_{s'\in L_s} \tilde{w}_T(P_{s'})$, where L_s is the set of terminal states with s being their common ancestor in the tree and $P_{s'}$ is the complete path associated with the terminal state s'.

Proof: This can be shown by induction. First, the statement is true for a (sub)tree with only one node representing a terminal state. Then, consider a nonterminal state s in the tree. Assume that for each immediate next state $s' \in C_s$, $X_{s'}^\star = \max_{s''\in L_{s'}} \tilde{w}_T(P_{s''})$. Based on Theorem 1, we then have $X_s^\star = \max_{s'\in C_s} X_{s'}^\star = \max_{s'\in C_s} \max_{s''\in L_{s'}} \tilde{w}_T(P_{s''}) = \max_{s''\in L_s} \tilde{w}_T(P_{s''})$, which completes the proof. ■

Theorem 2 says that $X_s^\star$ is equal to the maximum terminal reward over all the terminal states reachable from s, which is indeed the optimum cumulative reward of s in this MDP.

As discussed in Section IV-B, unlike the standard MCTS, we use $\widehat{X}_{s,n_s}$ to decide the next waypoint for the robot to move to, as shown in (12). In the next results, we prove the convergence of this policy and characterize its key properties. First, we show that the bias of the maximum cumulative reward converges to zero and is no greater than that of the average cumulative reward.

Theorem 3: When MCTS with the UCB selection policy is applied to the MDP of Section IV-A, for any state s, $\|\mathbb{E}[\widehat{X}_{s,n_s}] - X_s^\star\| \le \|\mathbb{E}[\bar{X}_{s,n_s}] - X_s^\star\| = \mathrm{O}(\ln(n_s)/n_s)$.

Proof: It is always true that $\widehat{X}_{s,n_s} \ge \bar{X}_{s,n_s}$. Thus, $\mathbb{E}[\widehat{X}_{s,n_s}] \ge \mathbb{E}[\bar{X}_{s,n_s}]$. In addition, $\bar{X}_{s,n_s} \le \widehat{X}_{s,n_s} \le X_s^\star$. Therefore, we have $\|\mathbb{E}[\widehat{X}_{s,n_s}] - X_s^\star\| \le \|\mathbb{E}[\bar{X}_{s,n_s}] - X_s^\star\| = \mathrm{O}(\ln(n_s)/n_s)$. ■

Next, we study the probability of failing to move to the optimum next state, when using the decision policy of (12). Specifically, we derive an upper bound on this probability, which shows that it converges to zero and characterizes its convergence speed. We first present two results from [28], which will be used in our proof.

Theorem 4. [28, Th. 3]: There exists some positive constant ρ such that for state s and any of its immediate next states $s' \in C_s$, $n_{s'} \ge \rho \ln(n_s)$.

Theorem 5. [28, Th. 4]: For state s, the following bounds hold for any given $\delta > 0$, provided that n_s is sufficiently large: $\mathbb{P}(\bar{X}_{s,n_s} \le \mathbb{E}[\bar{X}_{s,n_s}] - 9\sqrt{2n_s\ln(2/\delta)}/n_s) \le \delta$ and $\mathbb{P}(\bar{X}_{s,n_s} \ge \mathbb{E}[\bar{X}_{s,n_s}] + 9\sqrt{2n_s\ln(2/\delta)}/n_s) \le \delta$.

Theorem 4 shows that the number of visits for a state is lower bounded by a function of the number of visits of its parent node. Theorem 5 shows how the average cumulative reward of a state concentrates to its expectation probabilistically, as the number of visits increases.

By utilizing these bounds, we next show how to derive an upper bound on the failure probability.

Theorem 6: For any state s, by using the maximum cumulative reward-based decision policy of (12), the probability of failing to reach the optimum next state, $\mathbb{P}(s^\pi \ne s^\star)$, satisfies the following inequality, provided that n_s is sufficiently large:

$$\mathbb{P}(s^\pi \ne s^\star) \le \tilde{\rho}/\ln(n_s) \tag{13}$$

where $s^\star$ is the optimum next state (i.e., $s^\star = \mathrm{argmax}_{s'\in C_s} X_{s'}^\star$), s^π is the next state given by the decision policy of (12), and $\tilde{\rho}$ is a positive constant.

Proof: When $\widehat{X}_{s,'n_{s'}} > \widehat{X}_{s^\star,n_{s^\star}}$ for any $s' \in C_s\backslash\{s^\star\}$, the next state given by (12) will be different from the optimum next state, i.e., $s^\pi \ne s^\star$. As such, we have the following upper bound for $\mathbb{P}(s^\pi \ne s^\star)$, since the events $\widehat{X}_{s,'n_{s'}} \ge \widehat{X}_{s^\star,n_{s^\star}}$ $\forall s' \in C_s\backslash\{s^\star\}$ are not necessarily mutually exclusive

$$\mathbb{P}(s^\pi \ne s^\star) \le \sum_{s'\in C_s\backslash\{s^\star\}} \mathbb{P}(\widehat{X}_{s,'n_{s'}} \ge \widehat{X}_{s^\star,n_{s^\star}}). \tag{14}$$

Assume that $X_{s^\star}^\star > X_{s'}^\star$ $\forall s' \in C_s\backslash\{s^\star\}$, i.e., there exists only one optimum next state. Since $X_{s^\star}^\star - \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] = \mathrm{O}(\ln(n_{s^\star})/n_{s^\star})$, as shown in Theorem 1, there exists a sufficiently large N_1 such that when $n_{s^\star} \ge N_1$, $\mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - X_{s'}^\star \ge h_{s'}$, for all $s' \in C_s\backslash\{s^\star\}$, where $h_{s'} = (X_{s^\star}^\star - X_{s'}^\star)/2$.

Given $n_{s^\star} \ge N_1$, if $\widehat{X}_{s,'n_{s'}} \le X_{s'}^\star$ and $\widehat{X}_{s^\star,n_{s^\star}} > \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - h_{s'}$, then $\widehat{X}_{s,'n_{s'}} < \widehat{X}_{s^\star,n_{s^\star}}$, which leads to

$$\begin{aligned}\mathbb{P}(\widehat{X}_{s,'n_{s'}} \ge \widehat{X}_{s^\star,n_{s^\star}}) \le{}& \mathbb{P}(\widehat{X}_{s,'n_{s'}} > X_{s'}^\star)\\ &+ \mathbb{P}(\widehat{X}_{s^\star,n_{s^\star}} \le \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - h_{s'})\end{aligned}$$

where it can be easily seen that $\mathbb{P}(\widehat{X}_{s,'n_{s'}} > X_{s'}^\star) = 0$. Furthermore, since $\bar{X}_{s^\star,n_{s^\star}} \le \widehat{X}_{s^\star,n_{s^\star}}$, we have

$$\begin{aligned}&\mathbb{P}(\widehat{X}_{s^\star,n_{s^\star}} \le \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - h_{s'})\\ &\le \mathbb{P}(\bar{X}_{s^\star,n_{s^\star}} \le \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - h_{s'}).\end{aligned}$$

By setting $\delta = 1/n_{s^\star}$ for state $s^\star$ by using the result of Theorem 5, we have

$$\mathbb{P}(\bar{X}_{s^\star,n_{s^\star}} \le \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - 9\sqrt{2\ln(2n_{s^\star})/n_{s^\star}}) \le 1/n_{s^\star}$$

[10]The expectation is taken over the randomness in the simulation stage of MCTS, as well as over the random selection when multiple next states have an infinitely large second term in (9) due to $n_{s'} = 0$.

where $9\sqrt{2\ln(2n_{s^\star})/n_{s^\star}}$ converges to zero as $n_{s^\star}$ goes to infinity. As such, there exists a sufficiently large N_2 such that when $n_{s^\star} \geq N_2$, we have $9\sqrt{2\ln(2n_{s^\star})/n_{s^\star}} \leq h_{s'} \quad \forall s' \in C_s \backslash \{s^\star\}$. Therefore, given $n_{s^\star} \geq N_2$, we have the following for each $s' \in C_s \backslash \{s^\star\}$:

$$\begin{aligned}&\mathbb{P}(\bar{X}_{s^\star,n_{s^\star}} \leq \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - h_{s'}) \\ &\leq \mathbb{P}(\bar{X}_{s^\star,n_{s^\star}} \leq \mathbb{E}[\bar{X}_{s^\star,n_{s^\star}}] - 9\sqrt{2\ln(2n_{s^\star})/n_{s^\star}}) \leq \frac{1}{n_{s^\star}} \\ &\leq \frac{1}{\rho \ln(n_s)}\end{aligned}$$

where the last inequality is based on Theorem 4.

Based on (14), we then have $\mathbb{P}(s^\pi \neq s^\star) \leq \tilde{\rho}/\ln(n_s)$, where $\tilde{\rho}$ is a constant that depends on ρ and the size of C_s. Moreover, in order to have $n_{s^\star} \geq \max\{N_1, N_2\}$, we need $n_s \geq \max\{e^{N_1/\rho}, e^{N_2/\rho}\}$, based on Theorem 4. Therefore, given n_s sufficiently large, the inequality of (13) holds. ■

As we can see in the above theorem, for a state s, the failure probability is upper bounded by a function of n_s, which is $O(1/\ln(n_s))$. In other words, the failure probability converges to zero at least as fast as $\tilde{\rho}/\ln(n_s)$, as n_s increases.

B. Properties of the Optimum Solution

In this part, we mathematically characterize properties of the optimum spectral efficiencies. First, we define two waypoints p_{i_1} and p_{i_2} to be in the same *segment* of the path if they satisfy one of the following three conditions:

$$\begin{aligned}&1)\ k_{j-1} < i_1,\ i_2 \leq k_j;\ \text{for some } j \in \{2, \ldots, V\} \\ &2)\ i_1,\ i_2 \leq k_1; \qquad 3)\ i_1,\ i_2 > k_V\end{aligned} \tag{15}$$

where $k_j = \min\{i \in \{1, \ldots, K\} \mid \|p_i - s_j\| \leq \Delta_j\} \quad \forall j \in \{1, \ldots, V\}$. In other words, if two waypoints belong to the same segment of the path, then the robot does not collect new data when traveling between these two points.

By analyzing the Karush–Kuhn–Tucker (KKT) conditions [36], we can next characterize the optimum transmission spectral efficiencies for problem (6). Note that in the following theorem, $\Upsilon(q)$ is the random variable describing the predicted channel at location q, as defined in Section II-A.

Theorem 7: Consider problem (6). Given a complete path P, the optimum transmission spectral efficiencies satisfy the following properties:

1) $r_i^\star \geq r_j^\star$, if $\mathbb{E}[1/\Upsilon(p_i)] \leq \mathbb{E}[1/\Upsilon(p_j)]$ and p_i, p_j belong to the same segment of the path;

2) $r_i^\star = 0$, if $\mathbb{E}[1/\Upsilon(p_i)]$ is above a certain threshold;

3) $r_i^\star = r_{\max}$, if $\mathbb{E}[1/\Upsilon(p_i)]$ is below a certain threshold.

Proof: Given a complete path P, problem (6) reduces to problem (7). Assume $D_0 > 0$ and the robot does not need to use the maximum spectral efficiency all the time. There exists some strictly feasible solution to problem (7), indicating that Slater's condition holds [36], and since problem (7) is convex, the KKT conditions are sufficient and necessary for optimality [36]. The Lagrangian is then as follows, where $\upsilon \succeq 0$, $\nu \succeq 0$, $\eta \succeq 0$, and λ are the dual variables:

$$\begin{aligned}\mathcal{L}(R, \upsilon, \nu, \eta, \lambda) = &\sum_{i=1}^{K} \frac{(2^{r_i}-1)}{Z} \mathbb{E}\left[\frac{1}{\Upsilon(p_i)}\right] + \upsilon_i(r_i - r_{\max}) \\ &- \nu_i r_i + \sum_{j=1}^{V} \eta_j \left(\sum_{i=1}^{k_j} r_i - \frac{1}{t_c}\sum_{z=0}^{j-1} \tilde{D}_z\right) \\ &- \lambda \left(\sum_{i=0}^{K} r_i - \frac{1}{t_c}\sum_{j=0}^{V} \tilde{D}_j\right).\end{aligned} \tag{16}$$

For the KKT conditions, in addition to primal and dual feasibility, we have the gradient condition and complementary slackness as follows, with the optimum variables marked by $\star$. $\forall i \in \{1, \ldots, K\}$, we have

$$\begin{aligned}&\nabla_{r_i^\star}\mathcal{L} = \frac{2^{r_i^\star}\ln(2)}{Z}\mathbb{E}\left[\frac{1}{\Upsilon(p_i)}\right] + \upsilon_i^\star - \nu_i^\star + \sum_{j \in \mathcal{V}_i} \eta_j^\star - \lambda^\star = 0 \\ &\upsilon_i^\star(r_i^\star - r_{\max}) = 0, \qquad \nu_i^\star r_i^\star = 0\end{aligned} \tag{17}$$

where $\mathcal{V}_i = \{j \mid j \in \{1, \ldots, V\} \text{ and } k_j \geq i\}$. We also have the following additional complementary slackness conditions due to constraint (2) of problem (7):

$$\eta_j^\star \left(\sum_{i=1}^{k_j} r_i^\star - \frac{1}{t_c}\sum_{z=0}^{j-1} \tilde{D}_z\right) = 0 \qquad \forall j \in \{1, \ldots, V\}. \tag{18}$$

Denote $\tilde{\eta}_i = \sum_{j \in \mathcal{V}_i} \eta_j^\star$ from the KKT conditions. Then, the optimum spectral efficiencies can be derived as follows:

$$r_i^\star = \begin{cases} 0, & \text{if } \mathbb{E}\left[1/\Upsilon(p_i)\right] \geq \frac{(\lambda^\star - \tilde{\eta}_i)Z}{\ln(2)} \\ r_{\max}, & \text{if } \mathbb{E}\left[1/\Upsilon(p_i)\right] \leq \frac{(\lambda^\star - \tilde{\eta}_i)Z}{2^{r_{\max}}\ln(2)} \\ \log_2\left(\frac{(\lambda^\star - \tilde{\eta}_i)Z}{\ln(2)\mathbb{E}[1/\Upsilon(p_i)]}\right), & \text{otherwise.}\end{cases} \tag{19}$$

As $\tilde{\eta}_i$ is the same for any two waypoints in the same segment of the path, it can be confirmed that the three properties stated in this theorem hold based on (19). ■

Theorem 7 shows that within the same segment of the path, the optimum spectral efficiency should be higher (lower) where the channel quality is better (worse). The channel quality is measured by $\mathbb{E}[1/\Upsilon(p_i)]$, which is lower (higher) for a better (worse) channel. Moreover, when the channel quality is better than a certain threshold, the robot should take the maximum spectral efficiency and when it is worse than a certain threshold, there should be no transmission.

Consider a data intensive case, where the remaining onboard data to be transmitted is non-zero all throughout the path, up to the last segment, with no restriction on the last segment. Then, property (1) of Theorem 7 holds for any two waypoints in the path and the optimum spectral efficiencies can be solved using bisection, as we show next.

Corollary 1: When the remaining data are consistently nonzero before the robot senses the last site, the optimum

spectral efficiencies also satisfy: $r_i^\star \geq r_j^\star$, if $\mathbb{E}[1/\Upsilon(p_i)] \leq \mathbb{E}[1/\Upsilon(p_j)]$, for any two waypoints in the path. Moreover, the optimum spectral efficiencies can be solved using bisection.

Proof: When the remaining data are always nonzero before the robot visits the last site, we have $\eta_j^\star = 0\ \forall j \in \{1, \ldots, V\}$ based on the complementary slackness conditions of (18). The optimum spectral efficiencies are then as follows:

$$r_i^\star = \begin{cases} 0, & \text{if } \mathbb{E}\left[1/\Upsilon(p_i)\right] \geq \frac{\lambda^\star Z}{\ln(2)} \\ r_{\max}, & \text{if } \mathbb{E}\left[1/\Upsilon(p_i)\right] \leq \frac{\lambda^\star Z}{2^{r_{\max}}\ln(2)} \\ \log_2\left(\frac{\lambda^\star Z}{\ln(2)\mathbb{E}[1/\Upsilon(p_i)]}\right), & \text{otherwise.} \end{cases} \tag{20}$$

It can then be easily confirmed that the property stated in this corollary is true. Next, we show how to obtain the optimum spectral efficiencies via bisection. Based on (20), we have $r_i^\star = \max\left(0, \min\left(r_{\max}, \log_2\left(\frac{\lambda^\star Z}{\ln(2)\mathbb{E}[1/\Upsilon(p_i)]}\right)\right)\right)\ \forall i \in \{1, \ldots, K\}$. Due to constraint (4) of problem (6), $\sum_{i=0}^{K} r_i^\star - \frac{1}{t_c}\sum_{j=0}^{V}\tilde{D}_j = 0$ needs to hold. Therefore, $\lambda^\star$ is the solution to the following:

$$\sum_{i=0}^{K}\max\left(0, \min\left(r_{\max}, \log_2\left(\frac{\lambda^\star Z}{\ln(2)\mathbb{E}[1/\Upsilon(p_i)]}\right)\right)\right) - \frac{1}{t_c}\sum_{j=0}^{V}\tilde{D}_j = 0$$

where $\lambda^\star$ can be solved via bisection. Given $\lambda^\star$, we can then calculate the optimum spectral efficiencies from (20). ■

Next, we consider a special case where the robot does not need to sense any sites. This captures several real-world robotic motion and communication scenarios. The corresponding co-optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{P,\,R}\ \mathscr{E}_m(P) + \mathscr{E}_c(P,\,R;\,\Upsilon) \\ &\text{s.t. } (1)\ \mathscr{T}(\mathrm{P}) \leq \mathrm{T},\ (2)\ \mathrm{p}_1 = \mathrm{p_s},\ \mathrm{p_K} = \mathrm{p_f} \\ &\quad (3)\ \sum_{\mathrm{i}=1}^{\mathrm{K}} \mathrm{r_i t_c} = \tilde{\mathrm{D}}_0,\ (4)\ 0 \leq \mathrm{r_i} \leq \mathrm{r_{max}} \quad \forall \mathrm{i} \in \{1, \ldots, \mathrm{K}\} \end{aligned} \tag{21}$$

where the robot only needs to navigate from the start position to the destination and transmit the D_0 initial data.

The next corollary characterizes the optimum transmission spectral efficiencies in this special case.

Corollary 2: Consider the special case shown in problem (21). Given a complete path P, the optimum transmission spectral efficiencies satisfy all the properties in Theorem 7 and Corollary 1, and can be solved using bisection.

Proof: In this special case, there are no sensing-related constraints. As such, given a complete path P, we no longer have the terms with $\eta_j^\star$ in the KKT conditions (see the proof of Theorem 7). The optimal spectral efficiencies are then given as follows:

$$r_i^\star = \begin{cases} 0, & \text{if } \mathbb{E}\left[1/\Upsilon(p_i)\right] \geq \frac{\lambda^\star Z}{\ln(2)} \\ r_{\max}, & \text{if } \mathbb{E}\left[1/\Upsilon(p_i)\right] \leq \frac{\lambda^\star Z}{2^{r_{\max}}\ln(2)} \\ \log_2\left(\frac{\lambda^\star Z}{\ln(2)\mathbb{E}[1/\Upsilon(p_i)]}\right), & \text{otherwise.} \end{cases} \tag{22}$$

Based on (22), it can be confirmed that the properties stated in this corollary are true. Furthermore, since we have $\sum_{i=1}^{K} r_i^\star = \tilde{D}_0/t_c$, $\lambda^\star$ can be solved via bisection. ■

VI. Simulation Experiments

In this section, we solve the co-optimization problem (6) in realistic 2-D wireless channel environments using our proposed approach. We first consider a scenario that involves sensing, communication, and motion, and present the solution obtained by using our proposed co-optimization approach. We further extensively compare it with a benchmark method that separately optimizes sensing-motion and communication. More specifically, the benchmark first computes the shortest path that satisfies the sensing-motion constraints of problem (6) and subsequently, given the path, optimizes the transmission along this path by solving problem (7).[11] While this benchmark separately optimizes sensing-motion and transmission, it provides the maximum amount of co-optimization based on the available existing methods. Finally, we consider the special case of problem (21) with no sensing and further compare our approach with the best related state-of-the-art method.

A. Cooptimizing Sensing, Communication, and Motion

We validate our proposed approach using a realistic simulated 2-D wireless channel environment, where the channel parameters (obtained from real wireless measurements [34]) are: $\hat{\theta} = [-41.34,\ 3.86]$, $\hat{\alpha} = 3.20$, $\hat{\beta} = 3.09$ m, and $\hat{\sigma} = 1.64$. The robot predicts the channel with 1% prior channel measurements from random locations in this environment, based on the prediction framework of Section II-A. The required BER is 10^{-6}. The communication bandwidth is 20 MHz. The receiver noise power is -100 dBm. The maximum spectral efficiency is 6 b/s/Hz. The motion parameters are: $\kappa_1 = 7.4$ and $\kappa_2 = 0.29$, based on real power measurements of a robot [35]. The robot uses a constant speed of 1 m/s. The workspace is 50 m $\times$ 50 m with a grid size of 2 m $\times$ 2 m. We use the eight-neighbor setting: the robot can move to one of the eight neighboring grids that is within the workspace from its current grid in one step. We set the number of MCTS iterations for each step to be $N_I = 50$.

In this experiment, the robot's starting position is $[48,\ 2]$ and the final position is $[2,\ 48]$. The three sites are located at $[46,\ 16]$, $[44,\ 30]$, and $[24,\ 48]$, respectively, and the sensing ranges for

[11] Obtaining the shortest path that satisfies the motion constraints in problem (6) requires solving a traveling salesman problem with neighborhoods, and with the start and final positions directly connected in the tour. Given such a tour, we then remove the edge between the start and final positions in order to obtain the path from the start position to the final one. We utilize a self-organizing map-based algorithm to compute such a path [37].

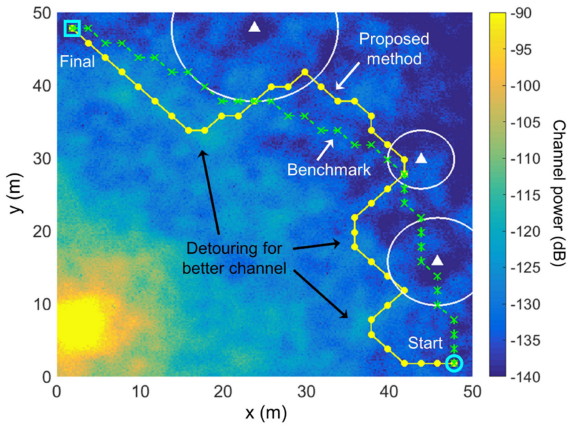


Fig. 2. Paths from solving problem (6) with three sites to sense, by using our proposed approach (yellow) and the benchmark (green). The yellow dots and solid curve represent the waypoints and the path, respectively, as given by our proposed approach. The green crosses and dashed curve represent the waypoints and the path given by the benchmark method. The circle and the square indicate the start and final positions, respectively. The white triangles indicate the site locations and the white circles indicate the sensing ranges for the sites. The colormap indicates the true channel power over this environment, where brighter (darker) colors indicate higher (lower) channel qualities. See the color pdf for optimal viewing.

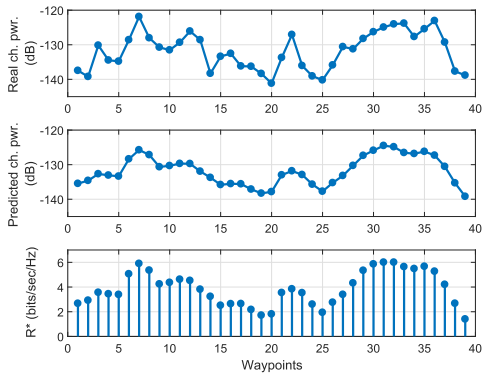


Fig. 3. Optimal transmission spectral efficiencies from solving problem (6) with three sites to sense, by using our proposed approach. The first and second rows show the actual and the predicted channel powers at the waypoints, respectively. The third row shows the optimum spectral efficiencies for the waypoints.

them are 6, 4, and 10 m, respectively. The robot has an initial data load of 60 b/Hz and needs to transmit 30 b/Hz additional data after sensing each site. The robot has a total time budget of 110 s. Fig. 2 shows the resulted path (yellow solid curve) by using our proposed approach. It can be seen that the robot detours toward areas with better channel qualities as needed (indicated by the black arrows in Fig. 2). Even on the path from site 2 to site 3, where the channel quality is generally poor, the robot is still able to detour a bit to exploit a slightly better channel (see the color pdf). Fig. 3 then shows the spectral efficiencies along the path using our proposed approach. It can be seen that the robot adopts a higher (lower) spectral efficiency when the channel quality is better (worse). As the remaining data to be transmitted are never zero before the last site, this confirms the theoretical result of

TABLE I
AVERAGE ENERGY COSTS BY USING OUR PROPOSED APPROACH AND THE BENCHMARK METHOD OVER 50 RANDOM PROBLEM INSTANCES

Methods	$\mathscr{E}_c$	$\mathscr{E}_m$	$\mathscr{E}_c + \mathscr{E}_m$
Benchmark method	27,008 J	576 J	27,585 J
Proposed approach	9,142 J	818 J	9,960 J
Overall energy saving			55%

The second, third, and fourth columns show the communication, motion, and total costs (in Joules), respectively. The last row shows the total energy saving by using our method, as compared to the benchmark.

Corollary 1. On the other hand, since the benchmark separately optimizes the path and the transmission, its path (green dashed curve in Fig. 2) is unaware of the channel. This makes the robot traverse in areas with poor channel qualities, resulting in a large total energy cost of 10910 J. In contrast, the total energy cost by using our proposed approach is 4757 J, which is 56% lower. It takes our approach 47.59 s to solve the co-optimization in this case (on a 3.40 GHz i7 PC). Although the benchmark takes only 5.30 s to compute the solution, the resulting total cost is significantly higher.

Next, we more extensively compare the performance of our proposed approach with the benchmark, over 50 problem instances. In these problem instances, the number of sites ranges from 1 to 4. In each problem instance, we use a different realization of the channel, and the site locations, sensing ranges, and the amount of data to be transmitted are randomized. Table I shows the energy costs (in Joules) by using our proposed approach and the benchmark, respectively, averaged over the 50 problem instances. It can be seen that overall, our proposed approach significantly reduces the total energy cost by 55%, as compared to the benchmark. This is because our approach is able to properly cooptimize sensing/motion and communication. For instance, our approach significantly reduces the communication cost by having the robot detour to areas with a better channel quality as needed, at the expense of a slightly higher motion cost.

We next experiment with different settings of our algorithm to further study its computational aspects. First, we reduce the number of MCTS iterations at each step from 50 to 10. This considerably reduces the computation time by 78% while still providing a much smaller total cost (e.g., 5751 J in the experiment of Fig. 4) when compared to the benchmark. In addition, it takes less than 0.5 s to compute each step, making our algorithm promising for real-time use. Next, we study the effect of a higher spatial resolution. We experiment with a smaller grid size of 1 m $\times$ 1 m. This results in slightly smaller total costs (e.g., 4299 J in the experiment of Fig. 4) while increasing the runtime by 3.9$\times$ and the memory usage (i.e., tree size) by 2.6$\times$, when compared to using 2 m $\times$ 2 m grids. This indicates that a very fine-grained spatial resolution can be computationally expensive and may not lead to significant performance gains.

B. Special Case of Co-optimization Without Sensing

We now consider the special case of problem (21), where there is no sensing. The 2-D environment, the motion and

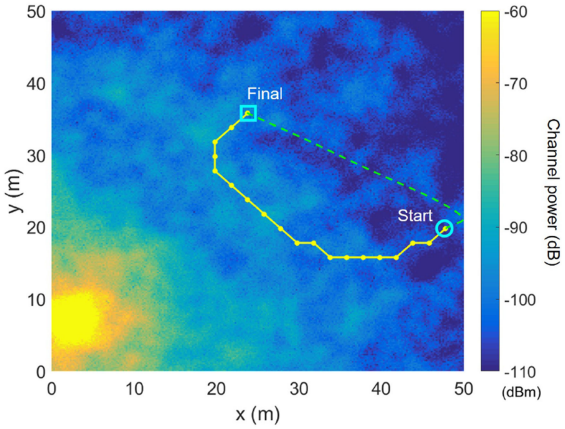


Fig. 4. Resulting path when using our proposed approach to solve problem (6) for the case of no sensing. The yellow curve shows the path and the yellow dots indicate the waypoints. The green dashed curve further shows the path obtained by using the method of [26]. The circle and the square indicate the starting and final positions, respectively. The colormap indicates the true channel power over this environment, where brighter (darker) colors indicate higher (lower) channel qualities. See the color pdf for optimal viewing.

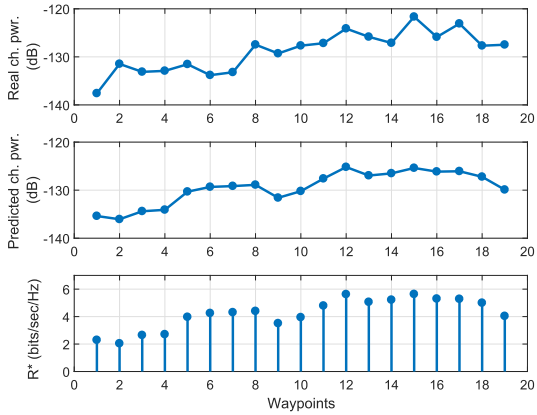


Fig. 5. Optimum transmission spectral efficiencies from solving problem (6) when there is no sensing, by using our proposed approach. The first and second rows show the actual and the predicted channel powers at the waypoints, respectively. The third row shows the optimum spectral efficiencies for the waypoints.

communication parameters, and the number of MCTS iterations for each step are the same as in Section VI-A. In this experiment, the robot starts from the initial position $[24,\ 36]$ and plans a path to the destination $[48,\ 20]$, with an operation time budget of 55 s. The robot needs to transmit a total of 80 b/Hz initial data to the remote station by the end of the trip.

Fig. 4 shows the resulting path (yellow) by using our proposed approach. It can be seen that the robot detours into a region with a better channel quality, before finally reaching the destination. Fig. 5 shows the optimum spectral efficiencies along this path given by our proposed approach, which are higher (lower) for waypoints with a better (worse) channel quality. This confirms the theoretical results of Corollary 2. In this experiment, the total energy cost by using our proposed approach is 1642 J, while the benchmark costs 10 721 J, which is over $6\times$ as much as that of our proposed approach. It takes our approach 16.73 s to compute the solution in this case.

While for the general case with sensing, there is no existing approach that can cooptimize motion, communication, and sensing, for the special case of no sensing, the work of [26] cooptimizes motion and communication using a different approach. More specifically, [26] formulates the trajectory-transmission co-optimization as an optimal control problem and employs a numerical algorithm to solve it.[12] As compared to our proposed approach, [26] does not guarantee convergence to the optimum solution, has a higher total energy cost, and requires more computation time. For instance, in this experiment, the total energy cost by using [26] is 3140 J, which is almost $2\times$ as much as that of our approach. Their algorithm takes 43.92 s to compute the solution, which is over $2.5\times$ slower than ours. The resulting path by using [26] is also shown in Fig. 4 (green). While this generated path is able to exploit the good channel regions while traversing to the final point (as indicated by the brighter blue colors near the path), this method is not able to steer the robot toward the best regions where the robot can minimize the total motion-communication cost.

Overall, these results demonstrate that our proposed approach is capable of cooptimizing sensing, communication, and motion for the complex optimization problem (6). Furthermore, our approach considerably outperforms the benchmark, which separately optimizes motion and communication.

VII. Conclusion

In this article, we studied the co-optimization of a robot's sensing, communication, and motion in a realistic wireless channel environment. In order to solve this complex optimization problem, we proposed a novel approach where we transformed the co-optimization problem into a specially designed MDP and utilized MCTS to solve it. More specifically, we showed that by iteratively optimizing the sensing/motion and the communication parts in different stages of MCTS, we can equivalently solve the original challenging co-optimization problem very efficiently. We then mathematically proved the convergence of our proposed approach, and characterized its convergence speed as well as other key properties of the optimum solution. Finally, we demonstrated the efficacy of our proposed approach in realistic 2-D wireless channel environments via extensive simulations.

As part of future works, it would be interesting to explore the real-time deployment of our proposed algorithm and further extend it to multirobot scenarios.

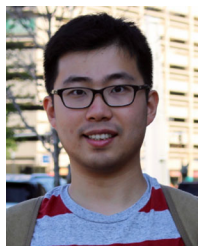
References

[1] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[2] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[12]Note that there is no straightforward way to extend [26] to the general scenario with sensing.

[3] S. Timotheou and G. Loukas, "Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios," in *Proc. ACM Symp. Appl. Comput.*, 2009, pp. 1171–1175.
[4] 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.
[5] A. Ghaffarkhah and Y. Mostofi, "Dynamic networked coverage of time-varying environments in the presence of fading communication channels," *ACM Trans. Sensor Netw.*, vol. 10, no. 3, pp. 1–38, 2014.
[6] Y. Wu, B. Zhang, X. Yi, and Y. Tang, "Communication-motion planning for wireless relay-assisted multi-robot system," *IEEE Wireless Commun. Lett*., vol. 5, no. 6, pp. 568–571, Dec. 2016.
[7] A. Zhou *et al.*, "Robotic millimeter-wave wireless networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1534–1549, Aug. 2020.
[8] A. Muralidharan and Y. Mostofi, "Energy optimal distributed beamforming using unmanned vehicles," *IEEE Control Netw. Syst*., vol. 5, no. 4, pp. 1529–1540, Dec. 2018.
[9] D. S. Kalogerias and A. P. Petropulu, "Spatially controlled relay beamforming," *IEEE Trans. Signal Process*., vol. 66, no. 24, pp. 6418–6433, Dec. 2018.
[10] G. Sun *et al.*, "Improving performance of distributed collaborative beamforming in mobile wireless sensor networks: A multi-objective optimization method," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6787–6801, Aug. 2020.
[11] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1647–1650, Aug. 2016.
[12] H. Wu, X. Tao, N. Zhang, and X. Shen, "Cooperative UAV cluster-assisted terrestrial cellular networks for ubiquitous coverage," *IEEE J. Sel. Areas Commun*., vol. 36, no. 9, pp. 2045–2058, Sep. 2018.
[13] L. Zhu *et al.*, "3-D beamforming for flexible coverage in millimeter-wave UAV communications," *IEEE Wireless Commun. Lett*., vol. 8, no. 3, pp. 837–840, Jun. 2019.
[14] G. A. Hollinger *et al.*, "Underwater data collection using robotic sensor networks," *IEEE J. Sel. Areas Commun*., vol. 30, no. 5, pp. 899–911, Jun. 2012.
[15] J. U. Khan and H.-S. Cho, "A distributed data-gathering protocol using AUV in underwater sensor networks," *Sensors*, vol. 15, no. 8, pp. 19331–19350, 2015.
[16] S. Wang, M. Xia, and Y.-C. Wu, "Backscatter data collection with unmanned ground vehicle: Mobility management and power allocation," *IEEE Trans. Wireless Commun*., vol. 18, no. 4, pp. 2314–2328, Apr. 2019.
[17] A. Meliou, A. Krause, C. Guestrin, and J. M. Hellerstein, "Nonmyopic informative path planning in spatio-temporal models," in *Proc. AAAI Conf. Artif. Intell.*, 2007, pp. 602–607.
[18] K. H. Low, J. M. Dolan, and P. Khosla, "Active Markov information-theoretic path planning for robotic environmental sensing," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2011, pp. 753–760.
[19] N. Atanasov, B. Sankaran, J. Le Ny, G. J. Pappas, and K. Daniilidis, "Nonmyopic view planning for active object classification and pose estimation," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1078–1090, Oct. 2014.
[20] T. Patten, W. Martens, and R. Fitch, "Monte Carlo planning for active object classification," *Auton. Robots*, vol. 42, no. 2, pp. 391–421, 2018.
[21] A. Arora, P. M. Furlong, R. Fitch, S. Sukkarieh, and T. Fong, "Multi-modal active perception for information gathering in science missions," *Auton. Robots*, vol. 43, no. 7, pp. 1827–1853, 2019.
[22] M. Popović *et al.*, "An informative path planning framework for UAV-based terrain monitoring," *Auton. Robots*, vol. 44, no. 6, pp. 889–911, 2020.
[23] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1271–1287, 2014.
[24] G. Hitz, E. Galceran, M. Garneau, F. Pomerleau, and R. Siegwart, "Adaptive continuous-space informative path planning for online environmental monitoring," *J. Field Robot.*, vol. 34, no. 8, pp. 1427–1449, 2017.
[25] L. Bottarelli, M. Bicego, J. Blum, and A. Farinelli, "Orienteering-based informative path planning for environmental monitoring," *Eng. Appl. Artif. Intell.*, vol. 77, pp. 46–58, 2019.
[26] U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, "Motion-communication co-optimization with cooperative load transfer in mobile robotics: An optimal control perspective," *IEEE Trans. Control Netw. Syst*., vol. 6, no. 2, pp. 621–632, Jun. 2019.
[27] H. Cai and Y. Mostofi, "Human-robot collaborative site inspection under resource constraints," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 200–215, Feb. 2019.
[28] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in *Proc. Eur. Conf. Mach. Learn.*, 2006, pp. 282–293.
[29] P. Clary, P. Morais, A. Fern, and J. Hurst, "Monte-Carlo planning for agile legged locomotion," in *Proc. Int. Conf. Automated Plan. Scheduling*, 2018, vol. 28, pp. 446–450.
[30] C. Mitash, A. Boularias, and K. E. Bekris, "Improving 6D pose estimation of objects in clutter via physics-aware Monte Carlo tree search," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3331–3338.
[31] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, USA: Wiley, 1994.
[32] C. Browne *et al.*, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.
[33] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press 2005.
[34] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Trans. Wireless Commun*., vol. 11, no. 3, pp. 964–978, Mar. 2012.
[35] Y. Mei, Y. Lu, Y. Hu, and C. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Trans. Robot*., vol. 22, no. 3, pp. 507–522, Jun. 2006.
[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[37] J. Faigl, P. Váňa, and J. Deckerová, "Fast heuristics for the 3D multi-goal path planning based on the generalized traveling salesman problem with neighborhoods," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2439–2446, Jul. 2019.



Hong Cai (Member, IEEE) received the B.E. degree in electronic and and computer engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2013, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, CA, USA, in 2015 and 2020, respectively.

His research interests include robotic visual understanding and robot decision optimization.

Yasamin Mostofi (Fellow, IEEE) received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1997, and the M.S. and Ph.D. degrees in wireless communications from Stanford University, Stanford, CA, USA, in 1999 and 2004, respectively.

She is currently a Professor with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA, USA. Her research has appeared in several reputable news venues such as BBC, *Huffington Post*, Daily Mail, Engadget, TechCrunch, NSF Science360, ACM News, and *IEEE Spectrum*, among others. Her research interests include intersection of communications and robotics on mobile sensor networks., X-ray vision for robots, RF sensing, communication-aware robotics, occupancy estimation, see-through imaging, and human–robot collaboration.

Dr. Mostofi was the recipient of 2016 Antonio Ruberti Prize from IEEE Control Systems Society, the Presidential Early Career Award for Scientists and Engineers (PECASE), the National Science Foundation (NSF) CAREER Award, and IEEE 2012 Outstanding Engineer Award of Region 6, among other awards. She is currently on the Board of Governors for IEEE CSS, a Senior Editor for IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS (TCNS), and a Program Cochair for ACM MobiCom 2022.