

Communication-Aware Information Gathering with Dynamic Information Flow

Abdallah Kassir, Robert Fitch and Salah Sukkarieh
Australian Centre for Field Robotics (ACFR)
The University of Sydney, NSW Australia
{a.kassir, rfitch, salah}@acfr.usyd.edu.au

Abstract—We are interested in the problem of how to improve estimation in multi-robot information gathering systems by actively controlling the rate of communication between robots. Communication is essential in such systems for decentralised data fusion and decision making, but wireless networks impose capacity constraints that are frequently overlooked. In order to make efficient use of available capacity, it is necessary to consider a fundamental trade-off between communication cost, computation cost and information value. We introduce a new problem, *dynamic information flow*, that formalises this trade-off in terms of decentralised constrained optimisation. We propose algorithms that dynamically adjust the data rate of each communication link to maximise an information gain metric subject to constraints on communication and computation resources. The metric is balanced against the communication resources required to transmit data and the computation cost of processing sensor data to form observations. The optimisation process selectively routes raw sensor data or processed observation data to zero, one or many robots. Our algorithms therefore allow large systems with many different types of sensors and computational resources to maximise information gain performance while satisfying realistic communication constraints. We also present experimental results with multiple ground robots and multiple sensor types that demonstrate the benefit of dynamic information flow in comparison to simpler bandwidth-limiting methods.

I. INTRODUCTION

Decentralised information gathering is an important task where outdoor robots coordinate to build maps, search for targets, track targets and classify objects. Coordination in outdoor systems relies on wireless communication between robots, but existing communication networks are subject to physical limits that are often idealised, over-simplified or ignored (Ghaffarkhah and Mostofi, 2011). We are interested in developing principled algorithms for decentralised information gathering that are designed to respect these limits and to use available communication resources efficiently.

The decentralised information gathering task is where a team of robots actively cooperates to maximise information about a given phenomenon. This task forms the basis of many applications such as cooperative search (Bourgault et al., 2004; Gan and Sukkarieh, 2011; Hollinger and Singh, 2012), target tracking (Xu et al., 2013b; Chung et al., 2006; Hollinger et al., 2009) and environmental monitoring (Golovin et al., 2010; Smith et al., 2011; Delle Fave et al., 2012). Robot teams are useful for information gathering because they can exploit diverse sensing and motion capabilities, access multiple simultaneous viewpoints and cover large areas more rapidly than single-robot systems. Communication is fundamental to

the task because robots must share data for estimation and coordinated decision-making.

Communication is not an infinite resource. However, research in multi-robot systems often makes two invalid assumptions that fail to respect the physical limits of real communication networks. The first such assumption is that simultaneous communication between multiple pairs of robots is independent. In most existing wireless networks, bandwidth resources are shared *globally* and link capacity decreases rapidly as the number of robots increases (Gupta and Kumar, 2000). The second invalid assumption, sometimes called the *r-disc* model, is that constant bandwidth is available within a given radius about a robot and that zero bandwidth is available otherwise. Real communication links are far more variable (Malmirchegini and Mostofi, 2012). The implications of failing to consider communication limitations are significant and hence communication in realistic environments is currently the topic of considerable research interest (Sadler et al., 2013).

One possible approach to address the issue of communication limits is to simply increase total network bandwidth by using more powerful and sophisticated radio hardware. However, it is always possible to generate a problem instance that exceeds any given resource limit. Sensors such as 3D laser range-finders generate data at a high rate, typically 1.7 million points per second. High-resolution cameras can produce data at even higher rates. Controlling the communication of such data is essential to real-world application of decentralised information gathering systems.

We believe that a better approach is to develop algorithms that make efficient use of the communication resources at hand. We refer to this approach as *communication-aware* information gathering. The idea is to choose when and how a given pair of robots should communicate based on the information value of the communication and given resource limits. This idea can be viewed as roughly analogous to *network flow*, where flow represents the data communicated within the system.

The main challenge is that information may be represented at multiple levels of abstraction ranging from raw sensor data to highly compressed forms such as target state observations. Therefore, we must choose not only how to route data but also in what form. This decision must consider computation costs, since data may be processed at various possible locations within a system with varying resource capacity. A given robot may process its sensor data on-board, transmit this

data to a powerful off-board processing station or rely on the computation resource of another robot. Manual design of a communication policy in this context is difficult and can result in poor communication efficiency. For example, down-sampling the rate of sensor data transmission may obey bandwidth constraints but can lead to unnecessary degradation in the performance of state estimation algorithms. The design task increases in difficulty for large heterogeneous systems, and any fixed policy would require redesign after any change to the underlying hardware or task. Therefore, the information flow must be adjusted dynamically and autonomously.

In this paper, we formalise the notion of communication-aware information gathering by introducing a novel problem formulation which we call the *Dynamic Information Flow (DIF)* problem. Given a graph-based representation of a decentralised information gathering system, the objective is to maximise the information value of communication by minimising a cost-based metric subject to constraints. The graph representation models an information gathering team as a system where data flows along a typical pipeline comprising sensors, perception algorithms, estimation algorithms and control algorithms. These logical elements are connected by communication links with associated costs, and a system may contain many such elements. For example, a single laser sensor may be connected to many other elements implemented on multiple robot platforms.

The DIF problem structure is designed to model trade-offs between information value, communication cost and computation cost at the system level. The information value of sensor observations is not defined globally but instead is defined relative to the belief state of each estimator element. Link costs are abstract costs that model both communication and computation. For example, a given sensor observation may be of high value to an estimator, but obtaining this information may incur a high cost due to the computational demands of a perception algorithm or due to large communication bandwidth requirements. Formulating the problem in this way provides a mechanism to balance these diverse costs against information value in a principled manner.

We define the problem as a family of optimisation problems with two concrete variants, *min-cost-DIF* and *threshold-DIF*. A solution to the problem is in the form of a set of multicast flow rates that determine which pairs of robots communicate at any given time. In *min-cost-DIF*, the objective is to minimise the sum of link costs, assuming the relative scale of these costs is known. In *threshold-DIF*, the relative scale of costs is not assumed to be known and the objective is to find a solution that satisfies a given cost threshold.

The DIF problem formulation provides several significant benefits. Because link costs are abstract and dynamic, the problem admits any realistic communication link model and is not limited to the r-disc assumption. The *threshold-DIF* variant can model the global bandwidth constraint imposed by common shared-channel communication systems. Modelling system elements logically as a graph where flow rates are dynamically optimised avoids the need to manually pre-determine the information architecture of the system. This property is particularly useful for heterogeneous systems with

many types of robots that have a range of sensing and computational resources. Although previous work has explored the problem of controlling communication between estimation and control elements (Semsar-Kazerooni and Khorasani, 2009; Kassir et al., 2012), this paper represents the first comprehensive effort to address communication between sensing and estimation.

We present algorithms and analysis for both problem variants. Our solution to *min-cost-DIF* is based on an adaptation of multicast routing. We prove that *min-cost-DIF* can be transformed such that existing multicast routing algorithms may be applied, and we present one such algorithm. Our solution to *threshold-DIF* is based on an optimisation method known as the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). We derive a decentralised version of this algorithm which we call *Distributed ADMM (DADMM)* and show how it can be applied to solve *threshold-DIF*. We analyse convergence and running time for all algorithms and validate these results through simulations including up to 28 nodes.

We also present experimental results that illustrate the behaviour of our algorithms and compare information gain performance with simple bandwidth-limiting methods. The task we consider is to track a moving target using multiple types of sensors. For the case of *min-cost-DIF*, the experimental system consists of one mobile robot equipped with a camera and one auxiliary static ground station. We also present simulation results for two mobile ground robots. For *threshold-DIF*, the experimental system consists of two outdoor mobile robots, with and without an auxiliary static camera. One robot is equipped with a 2D laser sensor and the other is equipped with a 3D laser. To further evaluate the performance of our algorithms, we present results from Monte Carlo simulations that demonstrate statistical significance.

Our results show that the DIF algorithms efficiently use available communication bandwidth to increase information gain. We observe that sensor data is either processed on-board or transmitted and processed at the ground station appropriately. We also observe that information from multiple sensor sources is communicated selectively based on sensor utility, available bandwidth and route overlap.

The paper is organised as follows. Section II discusses related work in the general area of communication limits in decentralised information gathering systems. The DIF problem and its two variants are defined in Sec. III. We present algorithms, analysis and experimental evaluation for the case of *min-cost-DIF* in Sec. IV. We then address *threshold-DIF* in Secs. V and VI; Sec. V presents our decentralised form of ADMM and Sec. VI presents algorithms, analysis and experimental evaluation based on the results from Sec. V. Section VII discusses the method of estimating sensor observation utility that we use in our implementations and Sec. VIII concludes the paper.

II. RELATED WORK

Existing work has studied aspects of communication-aware information gathering from several different perspectives. One

idea is to simply ease resource limits by boosting the capacity of wireless networks. Multi-radio multi-channel networks (Wu et al., 2000; Xing et al., 2007) can significantly increase network capacity by using multiple communication channels in parallel. Our previous work has shown that a single channel may be reused in a neighbour-to-neighbour architecture while avoiding mutual interference (Kuo and Fitch, 2014). Whereas the aim of these approaches is to maximise the throughput available from source to destination, this paper takes a complementary view. Instead of transmitting as much data as possible, we attempt to transmit only the most valuable data possible. Thus, data with little information value does not consume communication resources and available bandwidth is used efficiently.

From the perspective of *optimal control*, the entire problem (including communication decisions) can be modelled as a Decentralised Partially Observable Markov Decision Process (Dec-POMDP) (Gmytrasiewicz and Durfee, 2001; Goldman and Zilberstein, 2003; Carlin and Zilberstein, 2008; Williamson et al., 2008). Partially Observable Markov Decision Processes (POMDPs) are a powerful and general approach but are computationally intractable for large problems due to the “curse of dimensionality”. We are interested in problems with many robots and sensors, and we focus on computationally efficient solutions to the more specialised DIF problem.

Related problems in efficient information sharing have been studied in the context of sensor networks. In (Kulik et al., 2002), the SPIN routing protocol is introduced as a routing mechanism for sensor networks. Sensor nodes send an advertising message that contains metadata about the sensor information available and potential recipients send requests as required. However, the semantics of the metadata are not specified and are considered application dependent. In (Gupta et al., 2006), a sensor scheduling strategy for multiple sensors is proposed with bounds on the estimation error covariance. The strategy determines the order of sensor selection. Our work presents a general framework which allows for the dynamic selection of subsets of sensors in addition to dynamically selecting the processing platform for the transmitted sensor data.

In the context of target tracking, Chen et al. (Chen et al., 2010) propose an algorithm for sensor networks that uses minimal communication by only transmitting relative changes. In our previous work (Xu et al., 2013a), robots learn to predict the observation utility of other robots and adjust inter-robot communication accordingly. These approaches offer promising results for the application of target tracking but do not address information flow for nodes with heterogeneous capabilities and arbitrary utility functions. Our approach seeks to solve the trade-off between communication, computation and sensing in a general, yet computationally efficient, manner.

Another related problem is *communication-aware motion planning*, where robots choose paths that are advantageous with respect to properties of the communication network, such as connectivity maintenance (Hsieh et al., 2008; Mostofi, 2009; Yan and Mostofi, 2013; Stachura and Frew, 2011; Lindhé and Johansson, 2013; Twigg et al., 2013). Connectivity is important to networked robots, but the problem we propose in

this paper has a distinctly different objective. We build on the work in connectivity maintenance and assume that the network is connected. Our focus is on using this connected network efficiently by choosing when data should be transmitted and by choosing the best multicast routing. Instead of planning a path for a robot through its workspace, communication-aware information gathering involves planning a path for information through a network.

A more closely related problem is *distributed linear quadratic control* (Speyer et al., 2008; Molin and Hirche, 2009; Semsar-Kazerooni and Khorasani, 2009; Kassir et al., 2012), where each controller in a team makes control decisions based on knowledge of the team state vector. The full state vector may not be relevant to a given controller, and so the problem is to identify which state elements should be communicated. Our work again is distinct in that we focus on controlling the flow of observations as opposed to controller state information. Observations may be in the form of raw sensor data or in a processed task-dependent form such as target track estimates. We consider how to best route these observations through the network by balancing the computational effort required to process observations with the cost of communication and the value of this information in its various forms at any given time.

A classic problem in network flow optimisation is the *minimum cost flow* problem (Ahuja et al., 1993). The minimum cost flow problem has known efficient decentralised solutions; however, the DIF problem is more closely related to the *multicast network routing* problem. This problem is equivalent to the Steiner tree problem on directed graphs which is NP-complete (Ramanathan, 1996). In a special case using *network coding*, multicast routing can be solved in polynomial time and in a decentralised manner (Cui et al., 2007; Xi and Yeh, 2010). Our algorithms exploit this special case. However, in our implementations we use an approximation that approaches the performance provided by network coding in relatively small networks.

III. PROBLEM FORMULATION

In this section, we define the dynamic information flow problem. We introduce the general problem formulation and then define two important variants; one assumes *a priori* knowledge of link costs and the other bounds the sum of all link costs by a given global maximum.

A. Dynamic information flow

Our goal is to maximise information gain by controlling the flow of information within a decentralised information gathering system subject to communication and processing constraints. Data is continuously produced by sensors and consumed by other elements of the system. We first define a graph structure that models these system elements and the data flow between elements. Given such a model, we then define the dynamic information flow problem in general form.

A decentralised information gathering system is a configuration of several elemental components. *Sensors* are elements

that generate sensor data measured by physical sensing devices, such as laser scanners and cameras. Data from such sensors are transformed into *observations* by applying algorithms such as object detection and classification. *Processors* are computational elements that perform these processing tasks. Processors may be cascaded if necessary. The observations generated by processors act as input into *estimator* elements that maintain belief states. For example, an estimator could be an Extended Kalman Filter (EKF) in the tracking case or an occupancy grid in the mapping case.

Data flows via a communication system from sensors to processors, from processors to other processors and from processors to estimators. The topology of the resulting network is a directed acyclic graph, where information value, communication limits and computation limits induce costs or capacity constraints on the links in the graph. The induced link costs may vary according to the properties of the underlying communication mechanism, which may not be the same for all links. Elements of the system generally are physically distributed among multiple robots or ground stations and therefore communicate using an inter-robot communication system such as a wireless network. It is also possible for multiple elements to reside within a single physical platform and communicate using an intra-robot communication system such as a wired network or in-memory communication.

These system elements can also be viewed in terms of the well-known network flow problem (Ahuja et al., 1993) as follows. The commodity that flows through the network in this case is information in the form of sensor data or processed observations. Sensors correspond to supply nodes, estimators correspond to demand nodes, and processors correspond to intermediate, or transshipment, nodes. Communication links between nodes correspond to arcs or links between nodes of the network.

An example diagram of a decentralised information gathering system is shown in Fig. 1(a) with the corresponding system topology represented by the directed acyclic graph shown in Fig. 1(b). These diagrams could correspond, for example, to the case of two robots tracking a target using different types of sensors and with access to an off-board processing station. For target detection, each robot either processes its raw sensor data on-board or transmits the data to be processed off-board. Moreover, the robots can choose to either share raw sensor data or processed point observations instead.

Formally, a decentralised information gathering team is represented by a directed acyclic graph (DAG) $G = \{V, E\}$ where V is the set of vertices (or equivalently, nodes) and E is the set of edges or links. In the graph G , for every $i, k \in V$, if $(i, k) \in E$ then we say that k is a child node of i and i is a parent node of k . The set $\mathcal{C}(i) = \{k \in V : (i, k) \in E\}$ is the set of children of node i . Similarly, the set $\mathcal{P}(k) = \{i \in V : (i, k) \in E\}$ is defined as the set of parents of node k . We define $\mathcal{N}(i) = \mathcal{P}(i) \cup \mathcal{C}(i)$ as the neighbourhood of node i . A node with no parents is called a head node. A node with no children is called a tail node. We denote the depth of the graph G as $\kappa(G)$ defined as the number of nodes in the longest path from a head node to a tail node. The set $\bar{\mathcal{C}}(i) = \{k \in V : \text{there exists a directed path from } i \text{ to } k\}$ is referred to as the

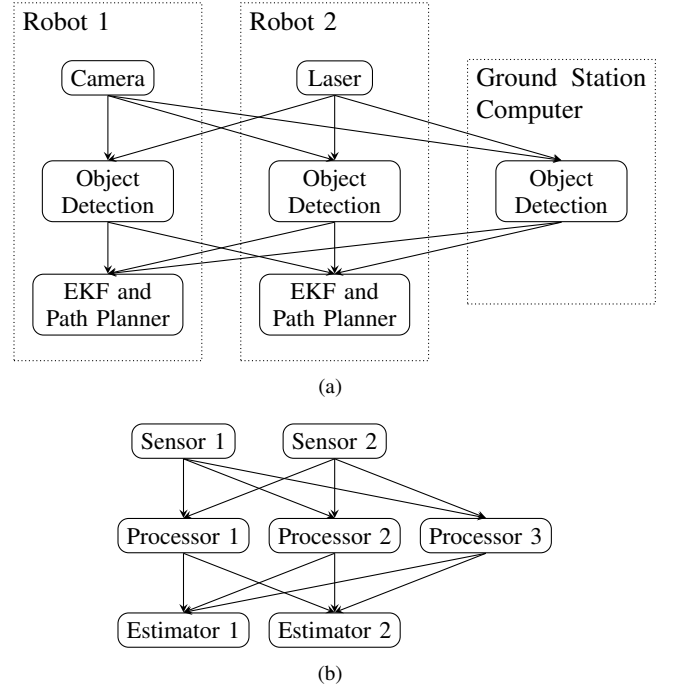


Fig. 1. (a) An example of a decentralised information gathering system with two robots and one off-board processor. (b) The corresponding network topology in our dynamic information flow formulation.

set of successors of node i . The set $\bar{\mathcal{P}}(k) = \{i \in V : \text{there exists a directed path from } i \text{ to } k\}$ is referred to as the set of ancestors of node i .

The set of nodes is composed of three subsets: the set of sensors V_s which act as sources, the set of processor nodes V_p which act as intermediate nodes and the set of estimator nodes V_e which act as destination nodes. Links connect nodes in V_s to nodes in V_p and nodes in V_p to nodes in V_e .

Sensor data is multicast from each sensor node $m \in V_s$ to all connected estimator nodes $j \in V_e$. Sensor m produces data at a fixed rate and this data is consumed by connected estimators at the same rate. To represent this production/consumption rate we introduce the variable $r_i^m(j)$ at node i for each sensor m and destination j . Variable $r_i^m(j)$ is called the inward flow and is set to sensor m 's data rate if $i = m$ or else the negative of sensor m 's data rate if $i = j$ or 0 otherwise. The average data rate of the flow passing through link (i, k) originating from source m and destined to j is defined as $x_{ik}^m(j)$. As an example, Fig. 2 shows a graph of an acyclic network with a single source. The inward flow variables are indicated for the source and destination nodes. A possible flow variable configuration is also shown for each link inside square brackets. The left entry is for j_1 and the right entry is for j_2 .

A set of flow variables $\{x_{ik}^m(j) : j \in V_e\}$ will lead to an average total flow of h_{ik} on link (i, k) . The relation between the total flow and the destination-specific flow variables will also depend on the underlying multicast implementation. With network coding, the total flow is simply the maximum flow over all destinations as defined in (1) (Ahlsweide et al., 2000). This relation will be assumed for the current problem formulation. The general validity of this assumption is further

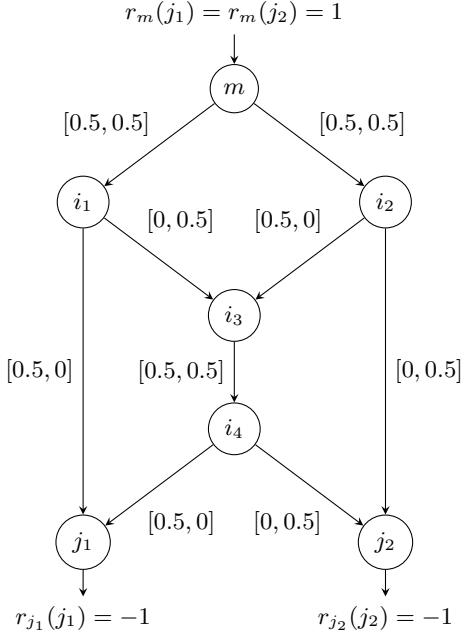


Fig. 2. An example routing configuration for the butterfly network. Numbers within brackets are the flow values for each link. The first value corresponds to destination j_1 and the second corresponds to destination j_2 .

discussed in Sec. IV-C.

$$h_{ik}^m = \max_{j \in V_e} x_{ik}^m(j) \quad (1)$$

Communication load, computation load and sensor observation utility induce a net link cost of c_{ik}^m per unit of data flow from source m passing through link (i, k) . The link cost is multiplied by the total flow h_{ik}^m to obtain the total link cost arising from source m . Summing over all sources, link (i, k) has a total cost of $\sum_m c_{ik}^m h_{ik}^m$.

Sensor observations induce a reward when reaching an estimator. In order to represent this reward, the information value of sensor m to estimator j is treated as a *negative cost* that is subtracted from the cost of each link incident to j .

Because a sensor observation may have little value for a given estimator, the system requires a mechanism by which a sensor can decide not to send any data to a certain destination. We model this option by adding a virtual, zero-cost link directly from each sensor to all connected estimators.

We now define the general dynamic information flow problem as follows. Given link costs $\{c_{ik}^m\}$ and inward flow rates $\{r_i^m(j)\}$, choose the set of flow variables $\{x_{ik}^m(j)\}$ such that the total cost summed over all links in the network is minimised subject to constraints. Link costs and constraints may vary over time.

B. Min-Cost-DIF

We define the first concrete form of the general problem, *min-cost-DIF*, according to the constrained optimisation (2-5). Information value, communication and computation resource demand are represented using link costs. This formulation is appropriate for situations where the relative costs between the items are known *a priori*.

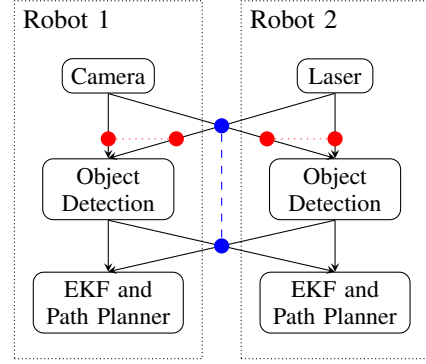


Fig. 3. An example of a system with inter-link constraints. Links tagged with the dashed line share a wireless communication medium, while those tagged with a dotted line share a common processing resource.

$$\text{minimise} \quad \sum_{(i,k) \in E, m \in V_s} c_{ik}^m h_{ik}^m \quad (2)$$

$$\text{subject to} \quad x_{ik}^m(j) \geq 0 \quad (3)$$

$$h_{ik}^m = \max_{j \in V_e} x_{ik}^m(j) \quad (4)$$

$$\sum_{l \in \mathcal{P}(i)} x_{li}^m(j) - \sum_{k \in \mathcal{C}(i)} x_{ik}^m(j) + r_i^m(j) = 0 \quad (5)$$

The first constraint (3) ensures that flow is always positive. The second constraint (4) represents the multicast condition and the third constraint (5) ensures that the sum of all inward and outward flow at a node is zero.

In min-cost-DIF, link costs may change over time due to changes in communication and processing costs as well as changes in sensor utility. For example, robots may move closer or further away from each other, resulting in a change in communication costs. Sensor viewpoint may also change, leading to a change in the value of on-board sensor observations.

C. Threshold-DIF

We introduce a second problem variant, *threshold-DIF*, to represent the case where the correct scale between communication costs, computation costs and information value is *not* known *a priori*. In this case, communication bandwidth and processing power are viewed as limited resources. The goal of threshold-DIF is thus to maximise information gain subject to communication bandwidth and processing power constraints.

We augment (2-5) to include two additional constraints (6-7) and define three additional input parameters, ν_{ik}^m , C_{ik} and K_s , to represent resource capacity limits. Constraint (6) bounds the weighted sum of flows originating from different sensors to respect a fixed capacity C_{ik} . The summation over all sensors in V_s is required since a link may carry messages originating from different sensors. Weights $\{\nu_{ik}^m\}$ are used to scale flow values h_{ik}^m on a per-link basis. For example, variations in required communication bandwidth due to link quality can be modelled by assigning appropriate values to $\{\nu_{ik}^m\}$. Similar to link costs, these variables may change over time.

$$\sum_{m \in V_s} \nu_{ik}^m h_{ik}^m \leq C_{ik} \quad (6)$$

TABLE I
LIST OF SYMBOLS.

| | |
|---------------|---|
| $r_i^m(j)$ | Inward flow rate at node i corresponding to source m and destined to j |
| $x_{ik}^m(j)$ | Flow rate through link (i, k) originating from source m and destined to j |
| h_{ik}^m | Total flow rate through link (i, k) originating from source m |
| c_{ik}^m | Cost per unit data flow through link (i, k) from source m |
| C_{ik} | Capacity of link (i, k) |
| K_s | Capacity of resource s |
| ν_{ik}^m | Flow scale factor for source m and link (i, k) |

$$\sum_{(i,k) \in \mathcal{S}_s} \sum_m \nu_{ik}^m h_{ik}^m \leq K_s \quad (7)$$

To motivate constraint (7), consider the decentralised information gathering network diagram in Fig. 3 which is a subset of the diagram in Fig. 1. If the two robots in this example exclusively use wireless communications to share observations, then all four links that cross the robot boundaries share a common resource (the wireless communication medium). This constraint is indicated in the figure by the dashed link. Moreover, if each robot only uses one computer for all processing requirements then the links from both sensors to each of the object detection modules share another common resource, the on-board processing computer. These constraints are indicated in the figure by dotted links. This class of constraints, which we call *inter-link* constraints, is represented in (7), where K_s is a fixed upper bound on resource s and \mathcal{S}_s is the set of links sharing resource s . Again, the flow rates are weighted because inter-link constraints impose bounds on the total flow across different links. These links could either be emanating from different nodes, as shown in the example in Fig. 3 or from the same node, when a node sends to many nodes using the same medium.

For convenience, the main terms used in the DIF problem formulation are listed in Table I. We note that i , k , m and j all correspond to nodes in the network graph. We use m to denote a source node, j to denote a destination node, i to denote the incumbent node and k to denote the incident node of an edge.

D. Sensor Utility

Sensor utility in the DIF formulation is a measure of the relative importance of sensor data with respect to a specific estimator. The importance of sensor data is evaluated based on the improvement it induces in the estimate. Hence, sensor utility depends both on the sensor and on the current state of the estimator. Sensor utility can be computed exactly using the POMDP formulation of the information gathering problem; however, this is intractable since Dec-POMDPs are NEXP-complete (Bernstein et al., 2002). In Sec. IV-C, we present a

myopic approximation to sensor utility that was used in our experiments. A further discussion of the issue of sensor utility approximation is provided in Sec. VII.

IV. MIN-COST DYNAMIC INFORMATION FLOW

In this section, we present a message-passing algorithm that solves the min-cost-DIF problem. We introduce a mapping that transforms an instance of min-cost-DIF into an instance of multicast network routing, prove equivalence and show that an algorithm that was originally developed for multicast network routing also finds an optimal solution to min-cost-DIF. We then describe our implementation of this algorithm in the context of min-cost-DIF along with empirical analysis in simulation that evaluates scalability. We also provide experimental demonstrations both in simulation and using a system of one mobile robot and one fixed ground station. Finally, we present results from a Monte Carlo simulation that statistically validates the performance advantage of our method.

A. Min-Cost-DIF Using Multicast Network Routing

An instance of min-cost-DIF can be transformed into an instance of multicast network routing (Cui et al., 2007; Xi and Yeh, 2010) as follows. The flow variable $x_{ik}(j)$ is replaced with $t_i(j)\phi_{ik}(j)$, where $t_i(j)$ is the total flow passing through i and destined to j while $\phi_{ik}(j)$ is the routing variable for link (i, k) ; more specifically, it is the fraction of $t_i(j)$ that is routed to k .

Following this change of variables, the resulting formulation is given by the optimisation problem (8-12). Constraint (10) states that the sum of the routing variables for each node is equal to one, while (11) and (12) are equivalent to (4) and (5).

$$\text{minimise} \quad \sum_{(i,k) \in E} c_{ik} h_{ik} \quad (8)$$

$$\text{subject to} \quad \phi_{ik}(j) \geq 0 \quad (9)$$

$$\sum_{k \in \mathcal{C}(i)} \phi_{ik}(j) = 1 \quad (10)$$

$$h_{ik} = \max_j t_i(j) \phi_{ik}(j) \quad (11)$$

$$t_i(j) = r_i(j) + \sum_{l \in \mathcal{P}(i)} t_l(j) \phi_{li}(j) \quad (12)$$

Given this mapping, existing algorithms for multicast network routing can be applied. Here we summarise one such algorithm, originally presented in (Cui et al., 2007). The algorithm is based on message passing and relies on obtaining the *marginal cost* $\delta_{ik}(j)$ for each link. The marginal cost is the rate at which the total cost increases due to a unit increase in flow along that link and is given by (13).

$$\delta_{ik}(j) = \begin{cases} c_{ik}/n + \sum_{l \in \mathcal{C}(k)} \phi_{kl}(j) \delta_{kl}(j) & \text{if } t_i(j) \phi_{ik}(j) \text{ and } n-1 \text{ other flows on link } (i, k) \text{ are the maximum} \\ \sum_{l \in \mathcal{C}(k)} \phi_{kl}(j) \delta_{kl}(j) & \text{otherwise} \end{cases} \quad (13)$$

Min-cost-DIF can be solved for each source m independently and in parallel. The full problem can be decomposed into independent sub-problems, one for each source, since the objective is additive and there are no inter-source constraints. This is evident from the problem formulation (2-5). Therefore, for simplicity of notation the subscript m is dropped from all variables in this section.

At the start of the algorithm, the routing variables $\{\phi_{ik}(j)\}$ are initialised arbitrarily such that they obey constraints (9) and (10). The routing variables are then repeatedly updated such that after iteration t the routing variables are set as $\phi_{ik}^{t+1}(j) = \phi_{ik}^t(j) + \Delta\phi_{ik}(j)^t$. The update direction $\Delta\phi_{ik}(j)^t$ is defined in (14). E_j is the set of edges belonging to the subgraph containing the ancestors of destination j and $\delta_{i,\min}(j) = \min_k \delta_{ik}(j)$.

$$\Delta\phi_{ik}(j)^t = \begin{cases} 0 & \text{if } (i, k) \in E_j \\ -\min\left\{\phi_{ik}^t(j), \frac{\alpha(\delta_{ik}(j) - \delta_{i,\min}(j))}{t_i(j)}\right\} & \text{if } \delta_{ik}(j) \neq \delta_{i,\min}(j) \\ \sum_{\substack{\delta_{ip}(j) \neq \\ \delta_{i,\min}(j)}} \Delta\phi_{ip}(j)^t & \text{if } \delta_{ik}(j) = \delta_{i,\min}(j) \end{cases} \quad (14)$$

The algorithm runs synchronously. First, the head nodes send messages with their flow contributions to their children. Once a node receives messages from all of its parents, it passes the message to its own children and so forth. The purpose of this downward sweep is to allow nodes to compute the flow of the current routing configuration. The flow values are necessary to compute the marginal costs required in the upward sweep. The downward sweep is followed by an upward sweep during which the marginal costs are computed according to (13) and the routing variables are updated according to (14). The downward and upward sweeps are decentralised, synchronous and are guaranteed to visit every node. Their sequence is dictated by Algorithm 1. The synchronicity property of Algorithm 1 is proved in Lemma 1.

Due to possible changes in link costs, this message passing optimisation runs continuously throughout system operation. As the system configuration changes, link costs are updated with new values. To ensure convergence, the interval between updates is set to an adequate time period. Further details on the appropriate length of the interval between updates can be found in Sec. IV-C.

Lemma 1. *In Algorithm 1, after node i has performed its t -th downward update and before forwarding this update:*

- 1) *Node i and all of its successors would have performed exactly $t - 1$ upward updates*
- 2) *All of its ancestors would have performed exactly t downward updates*

Proof: Suppose one of node i 's successors has performed $t' > t - 1$ upward updates. This means that at least one tail node in the successors has performed t' downward updates. This is impossible because node i has only yet forwarded

Algorithm 1 Synchronous message passing on DAGs

```

1: For node  $i$ 
2: if  $i$  is a head node then
3:   Perform a downward update and send downward mes-
     sage to children
4: end if
5: loop
6:   if a downward message is received from all parents
     then
7:     Perform a downward update and send downward
       message to children
8:     if  $i$  is a tail node then
9:       Perform an upward update and send upward mes-
        sage to parents
10:    end if
11:  end if
12:  if an upward message is received from all children then
13:    Perform an upward update and send upward message
      to parents
14:    if  $i$  is a head node then
15:      Perform a downward update and downward mes-
        sage to children
16:    end if
17:  end if
18: end loop

```

$t - 1$ downward messages. Since node i has performed t updates then its ancestors have performed at least t updates. Now, suppose one of node i 's ancestors has performed t'' updates where $t'' > t$. Then, the head nodes in the ancestry have performed at least t'' updates. This in turn means that they have performed $t'' - 1$ upward updates which means the tail nodes have performed $t'' - 1 > t - 1$ updates, which is impossible as just shown. This means that node i has forwarded at least $t' - 1 \geq t$ downward updates, which leads to a contradiction. ■

Due to observation rewards, negative costs may be assigned to links incident to an estimator $j \in V_e$. These negative costs are handled within the framework of the multicast network routing algorithm by solving an equivalent problem. A large enough constant \bar{c} is added to all $\{c_{ij} : j \in V_e\}$ to obtain a set of non-negative cost variables $\{c'_{ik}\}$ defined in (15).

The equivalence of solving the optimisation problem (8-12) to solving the problem with cost variable c'_{ik} instead of c_{ik} is proved in Theorem 1.

$$c'_{ik} = \begin{cases} c_{ik} + \bar{c} & \text{if } k \in V_e \\ c_{ik} & \text{otherwise} \end{cases} \quad (15)$$

Theorem 1. *Replacing link cost c_{ik} in problem (8-12) with c'_{ik} defined in (15) results in a problem equivalent to (8-12).*

Proof: For every link (i, j) such that $j \in V_e$, constraint (11) turns into the equality relation (16) instead since the only flow that should run along that link is the flow destined to estimator j .

$$h_{ij} = t_i(j)\phi_{ij}(j), \quad \forall j \in V_e, \quad \forall (i, j) \in E \quad (16)$$

After adding \bar{c} to the cost of these links to obtain $\{c'_{ij}\}$, a total of $\bar{c} \sum_{(i,j)} h_{ij}$ is added to the problem objective. Since \bar{c} is constant, we now proceed to prove that $\sum_{(i,j)} h_{ij}$ is constant. After substituting h_{ij} from (16), we obtain (17).

$$\sum_{(i,j)} h_{ij} = \sum_{j \in V_e} \sum_{i \in \mathcal{P}(j)} t_i(j) \phi_{ij}(j) \quad (17)$$

The right hand side of the equation is equal to the total flow arriving at a destination node summed over all destinations. By definition, this flow is equal to the source flow multiplied by the number of destinations and hence is constant. ■

B. Analysis

Subject to the choice of step size parameter α , the multicast routing algorithm is guaranteed to converge to the global optimum (Cui et al., 2007). Since a DAG contains no loops by definition, no contingencies are required to avoid routing loops. By Theorem 1, a min-cost-DIF instance with negative costs on links to an estimator can still be solved using the multicast routing algorithm by adding a sufficiently large positive constant to all such links.

The running time of multicast network routing with network coding is not explicitly provided in (Xi and Yeh, 2010; Cui et al., 2007) but is implied to be polynomial in the size of the network. In practice, we have observed a polynomial rate of increase as a function of network size, as shown in Sec. IV-C below.

C. Implementation and Scalability

Implementing multicast routing for min-cost-DIF involves three main challenges. First, we need to allow all nodes to find the available sources and destinations in a decentralised manner. Second, we need to ensure that the nodes have a suitable mechanism to compute any changing input parameters. Finally, we must choose a suitable multicast policy to implement the chosen flow rates $\{x_{ik}(j)\}$.

Each node must find the set of sources and destinations to which it is connected in the network. Initially, each node is aware of its direct neighbours only. By performing only one downward sweep and one upward sweep of message passing described in Algorithm 1, each node can obtain the list of sources and destinations to which it is connected. In our implementation, the downward messages contain the set of source identities received so far and the upward messages contain the set of destination identities received so far.

Link costs are continuously computed due to changes in the team configuration throughout the progress of its mission. In our implementation, communication costs are simply set proportional to inter-robot distance. Processing costs are assumed to be constant throughout the system operation. Sensor utility, on the other hand, can be more difficult to compute since each estimator must receive observations from a sensor in order to evaluate this utility. We maintain sensor utility values dynamically through an exploration-exploitation model. Each node obeys the chosen flow rate $x_{ik}(j)$ with probability $(1-\epsilon)$ (exploitation) and switches to another randomly selected flow rate with probability ϵ (exploration). The value of ϵ is set

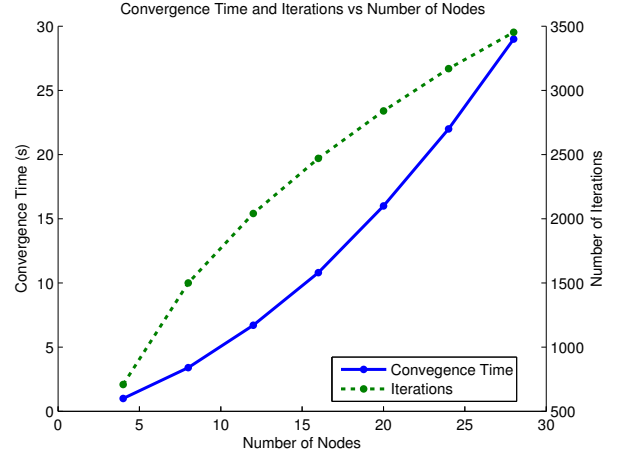


Fig. 4. Convergence time and convergence iterations of the message passing optimisation as a function of the number of nodes in a simulated network. The simulated network includes one sensor and an increasing number of processors and estimators where the number of processors is always one more than the number of estimators.

to a small positive number less than one. More sophisticated alternatives using machine learning methods are possible, such as in (Xu et al., 2013a). Section VII provides further discussion on this issue.

The chosen flow rates $\{x_{ik}(j)\}$ are implemented using a multicast policy that determines how the inward flow of messages at a node is distributed amongst its children. In our problem formulation, we assume that network coding is used. For small-sized networks, network coding can introduce unnecessary complication with little performance advantage in practice (Keshavarz-Haddad and Riedi, 2008). As an alternative, multicast routing can be implemented without network coding by using randomisation. The probability of sending a given inward message along a given outward edge is set proportional to flow variable $x_{ik}(j)$. In this case, the average total flow h_{ik} through the link for a source flow rate of r is given by (18) instead of the network coding relation given in (1).

$$\bar{h}_{ik} = r - \prod_j (r - x_{ik}(j)) \quad (18)$$

For a given set of flow variables, randomisation will result in higher total flow through a link. The extra capacity required in comparison to network coding is given by the relation in (19). From the relation, we deduce that the gap is zero if the maximum flow variable over a link is equal to either zero or the source flow and that the gap is less significant when there are fewer destinations. Therefore, for ease of implementation we use randomisation to implement the multicast policy. However, the total flow is still approximated by (1) since (18) otherwise leads to a non-convex problem. We found this approximation to be valid in practice.

$$\begin{aligned}
\bar{h}_{ik} - h_{ik} &= r - h_{ik} - \prod_j (r - x_{ik}(j)) \\
&\leq r - h_{ik} - (r - \max_{j \in V_e} x_{ik}(j))^{N_j} \\
&= r - \max_{j \in V_e} x_{ik}(j) - (r - \max_{j \in V_e} x_{ik}(j))^{N_j}
\end{aligned} \tag{19}$$

To demonstrate the scalability of the algorithm, we performed an empirical study of the convergence time for a given set of link costs. This study, which gives a convergence time estimate, also gives further insight into the time required between link cost updates.

We evaluated the convergence time in min-cost-DIF through a simulated network using randomised but fixed link costs. Our simulated network includes one sensor and a variable number of processors and estimators where the number of processors is always one more than the number of estimators. Results of the simulation for an increasing number of nodes are shown in Fig. 4. The convergence condition is satisfied when the change in the solution variables is below a certain threshold.

Convergence time depends both on the number of iterations required until convergence and the time expended in each iteration. The number of iterations to convergence is hardware independent and the results shown in Fig. 4 indicate that the number of iterations to convergence is a sub-linear function of the network size. The time required for each iteration involves computing routing updates and transmitting the updated values. The time complexity of the routing updates in (13) and (14) is polynomial in the size of the network. Transmission time, on the other hand, is typically linear in the size of the transmitted message which in turn is proportional to the size of the network.

The convergence times shown do not account for communication delay. To estimate such delay in practice, we observed from the experiments shown later in the paper (Secs. VI-C and VI-D) that the time for one iteration over wireless networks is typically less than 100 milliseconds. This value corresponds to networks concurrently being utilised for transmission of sensor data and processed observations. Based on this estimate and the number of iterations at convergence, we can estimate typical values for convergence time for networks from 5 to 10 nodes to be between 50 and 150 seconds.

This empirical analysis indicates that the running time of our algorithm is polynomial in the size of the network for typical problem instances of interest. We observed that the number of iterations required until convergence is sub-linear in the number of nodes, and the time complexity of each iteration is polynomial in the number of nodes. In practice, we found iteration times to be dominated by communication delay resulting in the convergence time indicated above. We envisage that this period can be reduced by employing various quality of service protocols, yet this solution is deferred to future work.

Based on the above analysis, we can now specify nominal values for the interval between link cost updates. The frequency of link cost updates is set such that the multicast routing algorithm has sufficient time to converge. In our implementation, we chose fixed values between 50 and 150

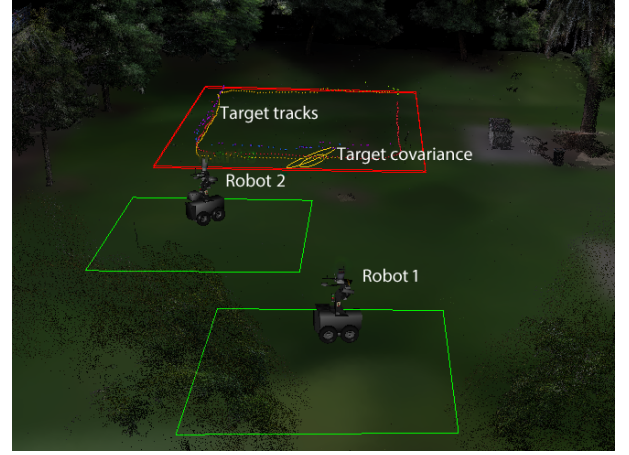


Fig. 5. Demonstration setting of the two-robot simulation and experiment. Robots 1 and 2 are shown with their boundaries. Sample target tracks are shown making a square pattern inside the region of interest.

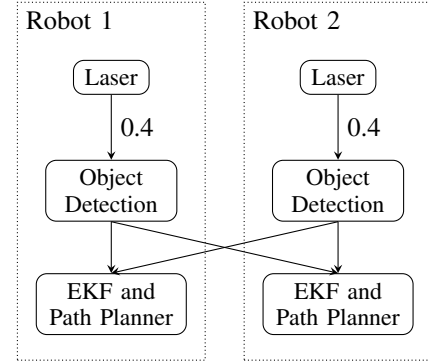


Fig. 6. The min-cost-DIF network diagram for the two-robot simulation. Virtual links are omitted for clarity.

seconds. However, we note that while the optimisation algorithm is iterating, a valid routing is available and information can continuously flow through the network. The frequency of link cost updates determines how reactive the algorithm is to changes in estimated sensor utility, and thus the importance of a higher update frequency would be to handle situations where sensor utility changes rapidly. We leave consideration of this case to future work.

D. Two-Robot Simulation

We evaluated our solution approach to min-cost-DIF in simulation for the case of two mobile robots tracking a moving target. The implementation is decentralised with nodes running in separate processes. The aim of this simulation is to demonstrate the multicast behaviour of dynamic information flow and to show the improvement in information gain realised in the min-cost-DIF setting in comparison to a control condition with uniform-rate communication.

The simulation setup is shown in Fig. 5. Two mobile robots are assigned to separate workspaces. The first robot has a 360° field-of-view sensor and the second robot has a 180° field-of-view sensor. The sensors are bearing-only sensors, forcing cooperation between the robots. A moving target tracked by

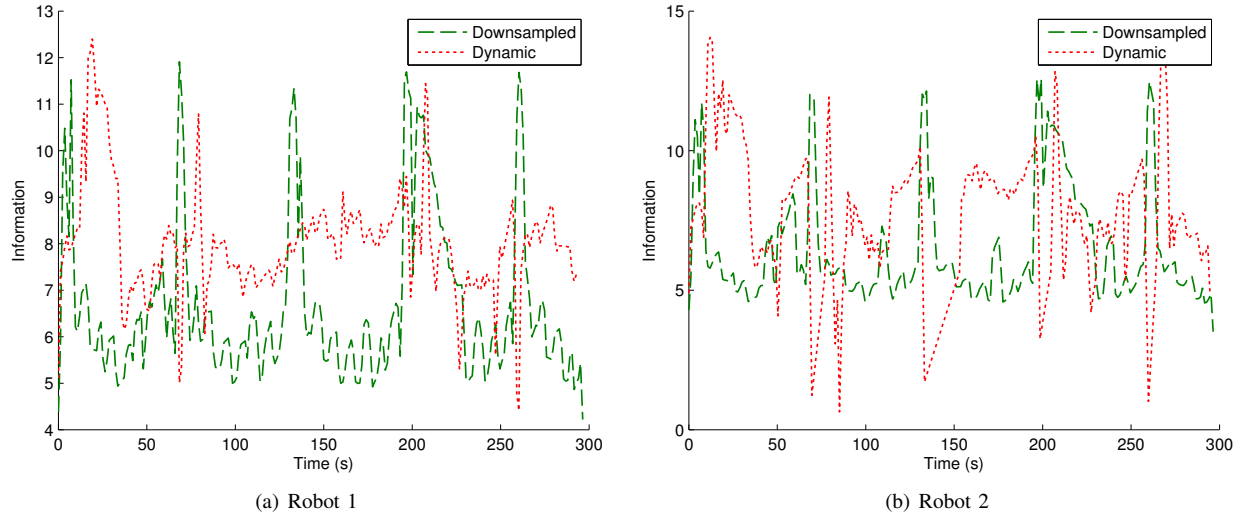


Fig. 7. Information value (negative entropy) of the robots' target estimate over time for the two-robot simulation. The plots are shown for two communication methods, down-sampled and dynamic, with both requiring the same amount of computation on average.

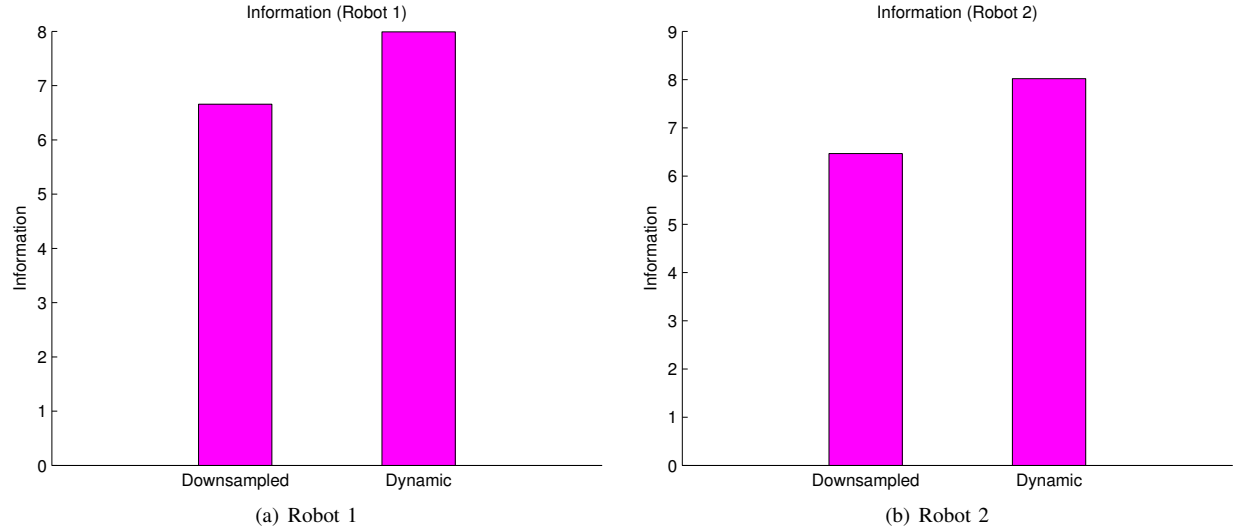


Fig. 8. Time averages of the plots in Fig. 7 shown in bar format. The improvement for the dynamic case over down-sampling is clearly observed.

TABLE II
AVERAGE FLOW RATES AND SENSOR UTILITY FOR THE TWO-ROBOT SIMULATION.

| Link | Flow Rate | Sensor Utility |
|---|-----------|----------------|
| Robot 1's Sensor to Robot 1's Estimator | 44.82 | 0.10 |
| Robot 1's Sensor to Robot 2 | 55.03 | 1.64 |
| Robot 2's Sensor to Robot 2's Estimator | 26.12 | 0.04 |
| Robot 2's Sensor to Robot 1 | 45.01 | 0.97 |

the robots moves in a circular pattern within a square region of interest outside the robots' workspaces.

The network diagram of the system is shown in Fig. 6. The object detection routine on each robot imposes a processing cost. Due to the multicast property, the processing cost of each processing module is distributed among the receiving estimators. Therefore, if the estimators collectively evaluate

an observation utility greater than the processing cost, then the sensor raw data should be processed and sent forward. Virtual links, not shown in the figure, allow for the no-send policy. The cost of the links incident onto the estimators is adjusted throughout the simulation by the robots' sensor utility evaluations.

To validate the performance of the algorithm, a control test was also run. In the control test, the sensor rates were reduced to the same average rate used in the dynamic case.

The information for each of the robots' estimate over time is shown in Fig. 7 and the corresponding average bars are shown in Fig. 8. In Fig. 7(a), the information in Robot 1's estimate was higher on average for the dynamic flow case. In contrast to the down-sampled case, the dynamic flow case boosted inter-robot flow when required preventing long periods of poor information gathering performance. The advantage of the dynamic flow case can also be seen more clearly in Fig. 8(a). The dynamic case for Robot 2 exhibits some

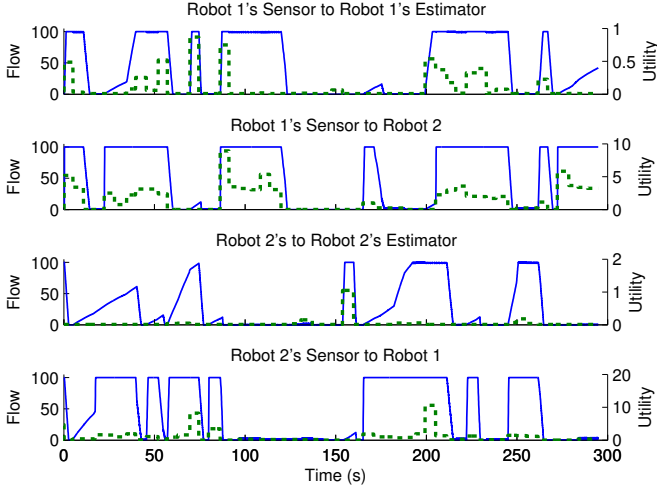


Fig. 9. Flow rates and sensor utility for the two-robot simulation. Flow rates are shown in solid lines while the evaluated sensor utility is shown in dashed lines.

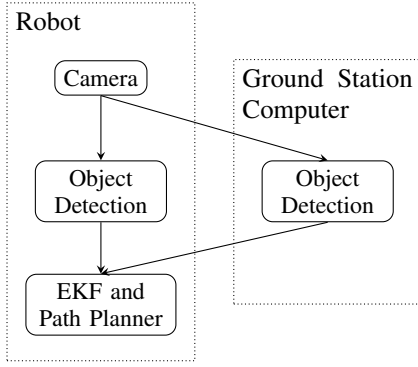


Fig. 10. The min-cost-DIF diagram of the robot and ground station demonstration scenario.

lag in information gathering performance but it eventually outperformed the down-sampled case as shown in Fig. 7(b). This is also confirmed in Fig. 8(b).

The policies and estimated sensor utilities for the dynamic flow case are plotted against time in Fig. 9 with the corresponding averages shown in Table II. The plots show consistency between sensor utility and flow. In particular, the oblique parts of the flow curve correspond to time periods where a robot would receive sensor observations freely due to raw data already being processed for the other robot. This free reception of data is a feature of multicast routing.

As noted in Sec. IV-C, discrete flow decisions minimise the error from our maximum flow approximation of the total flow in each link. As seen in Fig. 9, with the exception of transient periods, the flow variables were mainly either 0 or 100.

E. Robot and Ground Station Simulation

We also evaluated the case of one mobile robot and one ground station. The main purpose of this simulation is to show the generality of the dynamic information flow approach. The simulation was run using ROS and Gazebo.

The setting includes one Pioneer 2DX robot model with an on-board camera and one off-board processing station.

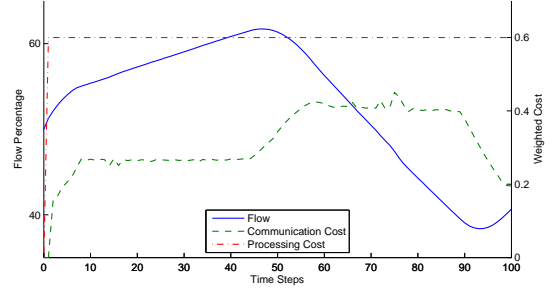


Fig. 11. Flow, communication cost and processing cost of the link from the sensor to the ground station processor for the one robot/one ground station simulation.

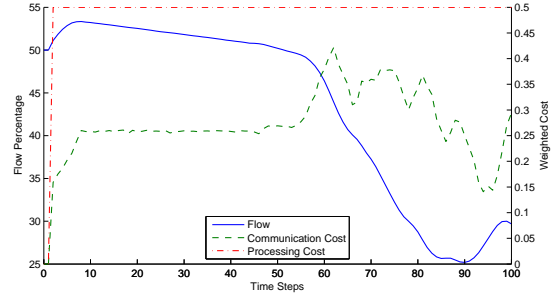


Fig. 12. Experimental results for the one robot/one ground station case.

The robot's on-board processing is computationally expensive, however, it has wireless access to the off-board processing station. The demonstration process is as follows. The robot begins near the processing station. It then proceeds to gain information about a moving target.

The network diagram of the demonstration is depicted in Fig. 10. Subject to communication cost, which is set proportional to distance, the robot's decision is expected to vary between on-board and off-board processing. The maximum flow assumption is valid for this scenario since there is only one destination node.

Figure 11 shows the flow between the robot and the ground station, the communication cost and the robot's processing cost over time. The processing cost is assumed to be fixed over time as shown in the figure. The communication cost varies based on the distance between the robot and the ground station and the flow is adjusted accordingly. In Fig. 11, at around time step 50, communication cost outweighs the cost of on-board processing. This leads to a switch from off-board processing to on-board processing. It should be noted that the communication cost displayed in the plot is only for transmission from the sensor to the processor. When compared with on-board processing cost, the total communication cost is doubled. The results of this simulation show the generality of the dynamic flow formulation. Even though the expected behaviour here is as expected, the aim is to show that this behaviour was achieved within the min-cost-DIF framework without modification.

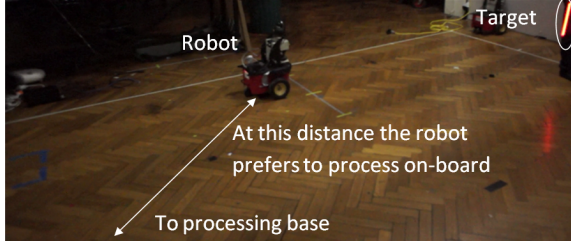


Fig. 13. Snapshot from the one robot/one ground station experiment. The image shows the robot having moved away to follow the target. At this distance, the robot prefers to perform processing on-board.

F. Robot and Ground Station Experiment

We also performed an experiment in hardware for the one robot/one ground station case. A Pioneer 2DX robot equipped with an on-board computer using an Intel Atom processor N270 1.6 GHz was used. The robot is equipped with a SICK LMS291 2D lidar used exclusively for localisation. The robot uses an on-board webcam as a 2D bearing-only sensor to track a moving target. With the exception of the off-board processing ground station, all processes were executed on-board the robot including localisation, image processing (when required), estimation, decision making and the information flow control algorithm.

The change in flow, communication cost and processing cost over time is shown in Fig. 12. These results are consistent with the simulation results for the analogous case. We observe that the robot initially chooses to send images to be processed off-board by the ground station. The ground station performs object detection and sends point observations back to the robot. As the robot moved away to track the target, the communication cost increased and thus the robot chose to perform processing on-board. A snapshot of the moment when the robot decided to switch to on-board processing is shown in Fig. 13.

G. Monte-Carlo Simulation

We performed a Monte Carlo simulation for the experimental setting of Sec. IV-D comparing the down-sampled and the dynamic communication cases. The aim of the simulation is to analyse the statistical significance of the performance advantage introduced by our solution to min-cost-DIF.

Each method was tested in twenty randomly initialised trials running for one minute each. The down-sampling rate was chosen to match the average rate resulting from the dynamic case.

The results of the Monte Carlo simulation are shown in Fig. 14 in box-plot format. The results shown assume each trial as one sample. Dynamic communication clearly outperforms down-sampling. A Welch's t -test for statistical significance resulted in a p -value less than 0.001.

V. DISTRIBUTED OPTIMISATION FOR THRESHOLD-DIF

In this section we introduce a distributed optimisation method that will form the basis of our solution to threshold-DIF. The method is a distributed version of ADMM which we

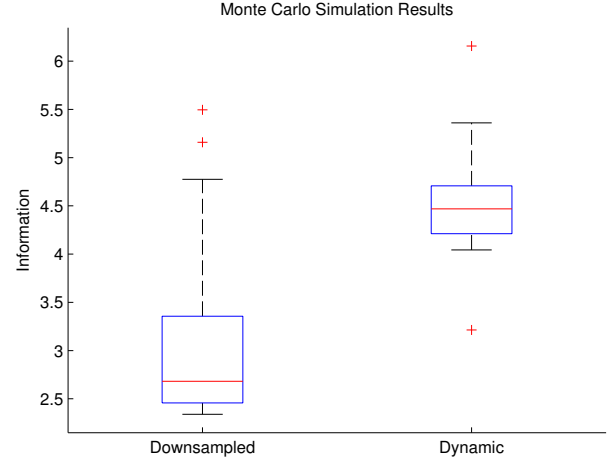


Fig. 14. Monte Carlo simulation results comparing down-sampling with dynamic information flow. Each method was tested on the two-robot scenario for twenty trials running for one minute each. In the results depicted, a trial acts as one sample. The box extents represent the first and third quartile, while the whiskers represent the extrema. The median is represented by the horizontal line inside the box.

call *DADMM*. The DADMM method solves distributed non-smooth constrained convex optimisation problems with a DAG structure. Threshold-DIF is a distributed optimisation problem since its objective is a sum of local objectives and it only has neighbour-to-neighbour constraints. It is non-smooth due to the linear objective, and it has the structure of a DAG by definition.

We begin with a brief introduction to ADMM. Then, DADMM is presented along with complexity analysis. We then present the mapping from threshold-DIF to DADMM later in Sec. VI.

A. ADMM

For convenience, we provide a brief summary of ADMM. A detailed description can be found in (Boyd et al., 2011).

ADMM solves optimisation problems of the form given by (20). The objective is assumed to be a sum of two proper convex functions f_1 and f_2 where the first is a function of the vector z_1 and the other is a function of the vector z_2 . We refer to z_1 as the *primary* vector variable and to z_2 as the *secondary* vector variable. The Lagrangian of the problem is shown in (21).

$$\begin{aligned} & \text{minimise} && f_1(z_1) + f_2(z_2) \\ & \text{subject to} && A_1 z_1 + A_2 z_2 = b \end{aligned} \quad (20)$$

$$L(z_1, z_2, y) = f_1(z_1) + f_2(z_2) + y^T (A_1 z_1 + A_2 z_2 - b) + (\rho/2) \|A_1 z_1 + A_2 z_2 - b\|_2^2 \quad (21)$$

ADMM is summarised in (22). Each iteration involves three updates. The primary update minimises the Lagrangian about z_1 , the secondary update minimises the Lagrangian about z_2 and the third updates the Lagrangian variable y . Convergence is shown in (Boyd et al., 2011).

$$\begin{aligned}
z_1^{k+1} &:= \underset{z_1}{\operatorname{argmin}} L(z_1, z_2^k, y^k) \\
z_2^{k+1} &:= \underset{z_2}{\operatorname{argmin}} L(z_1^{k+1}, z_2, y^k) \\
y^{k+1} &:= y^k + \rho(A_1 z_1^{k+1} + A_2 z_2^{k+1} - b)
\end{aligned} \tag{22}$$

B. Problem Formulation

The general form of optimisation problems that can be solved by DADMM is described as follows. Consider the DAG $G = \{V, E\}$ defined in Sec. III. Attach to each node $i \in V$ a vector variable x_i and a proper convex function $f_i(x_i)$, which is not necessarily smooth. Node i can have constraints with its parents as per (24) where g_i is an affine function. The notation x_U where $U = \{i_1, \dots, i_n\} \subset V$ is defined as the concatenation of all vectors x_i such that $i \in U$, i.e. $x_U = (x_{i_1}, \dots, x_{i_n})$. Function g_i is interpreted as a vector valued function with its dimension indicating the number of constraints n_g^i . The goal of DADMM is to solve the optimisation problem (23-24).

$$\text{minimise} \quad \sum_{i \in V} f_i(x_i) \tag{23}$$

$$\text{subject to} \quad g_i(x_i, x_{\mathcal{P}(i)}) = 0, \forall i \in V \tag{24}$$

C. Inequality Constraints

The standard form of ADMM does not include inequality constraints. Therefore, we have only included equality constraints g_i in the problem definition. This is a non-restrictive assumption since by adding extra variables, inequality constraints can be transformed into equality constraints as we will show. Suppose that instead of g_i we have a function \bar{g}_i that is required to satisfy the inequality constraint (25) instead.

$$\bar{g}_i \leq 0 \tag{25}$$

By adding a slack variable p_i , this inequality constraint becomes an equality constraint plus a non-negative constraint on p_i as shown in (26). The slack variable p_i can be viewed as a variable belonging to a virtual parent node whose objective is an indicator function that is zero when p_i is non-negative and infinity otherwise.

$$\begin{aligned}
g_i &= \bar{g}_i + p_i = 0 \\
p_i &\geq 0
\end{aligned} \tag{26}$$

In DADMM, p_i can be optimised independently. The solution of the optimisation over p_i is given by (27).

$$p_i := \max\{-\bar{g}_i, 0\} \tag{27}$$

D. DADMM

DADMM consists of a preliminary decentralisation step followed by the main optimisation process. The decentralisation step is only performed once during which the optimisation problem is modified, through the addition of variables and constraints, such that it only requires neighbour-to-neighbour communication. In the optimisation process, message passing and optimisation updates run in a sequence that enforces

decentralisation while retaining equivalence to centralised ADMM.

The decentralisation step modifies constraints (24). For every vector x_i where $i \in V$, a mirror vector \bar{x}_i is introduced. The vector \bar{x}_i acts as an interface for all other nodes. Any child node k that has a constraint including x_i replaces x_i with a local copy \tilde{x}_i^k and an equality constraint between \tilde{x}_i^k and \bar{x}_i is added. Symmetrically, from node i 's perspective $x_{\mathcal{P}(i)}$ is replaced with $\tilde{x}_{\mathcal{P}(i)}$. Therefore, from node i 's perspective, (24) is replaced with (28-30).

$$g_i(x_i, \tilde{x}_{\mathcal{P}(i)}) = 0 \tag{28}$$

$$x_i - \bar{x}_i = 0 \tag{29}$$

$$\tilde{x}_{\mathcal{P}(i)} - \bar{x}_{\mathcal{P}(i)} = 0 \tag{30}$$

The new constraints are assigned the following Lagrangian multiplier vectors. The Lagrangian multiplier λ_i is associated with constraint (28), μ_i is associated with constraint (29) and $\eta_{\mathcal{P}(i)}^i$ is associated with constraint (30).

All the above constraints and variables except $\bar{x}_{\mathcal{P}(i)}$ are attached to node i . This means that $\bar{x}_{\mathcal{P}(i)}$ is node i 's only dependency on its parent nodes and \bar{x}_i is the interface variable that is shared with node i 's children. We note that constraint (28) is now an internal constraint. This decentralisation has decoupled the parents of node i . The decoupling is evident by noting that (30) is a decoupled set of equality constraints: $\tilde{x}_l^i = \bar{x}_l, \forall l \in \mathcal{P}(i)$. The decentralisation step is shown schematically in Fig. 15.

From the ADMM perspective, the sets of vector variables x_i and $\tilde{x}_{\mathcal{P}(i)}^i$ are mapped to z_1 in (20) and the sets of variables \bar{x}_i are mapped to z_2 . The sets of constraints (28-30) are collectively mapped to the equality constraint in (20). Define $\hat{x}_i = (x_i, \tilde{x}_{\mathcal{P}(i)}^i)$. Based on the mapping to ADMM, \hat{x}_i is the primary vector variable while \bar{x}_i is the secondary vector variable.

The main optimisation process consists of message passing and optimisation updates defined in a sequential and decentralised manner that is equivalent to the centralised version (22). The process begins with the head nodes and then proceeds to traverse the graph according to Algorithm 1. Algorithm 1 refers to two types of updates and two types of messages, upward and downward. We will now proceed to define what takes place during each update and what each message contains.

At the outset, each node $i \in V$ is initialised with 1x_i , $^1\bar{x}_i$, $^1\tilde{x}_{\mathcal{P}(i)}^i$, $^0\lambda_i$, $^0\mu_{ik}$ and $^0\eta_{\mathcal{P}(i)}^i$. In the t -th downward update, node i updates its Lagrangian multipliers to obtain $^t\lambda_i$, $^t\mu_i$ and $^t\eta_{\mathcal{P}(i)}^i$. It then updates the primary variables to obtain $^{t+1}x_i$ and $^{t+1}\tilde{x}_{\mathcal{P}(i)}^i$. The node's downward message contains $^t\bar{x}_i$ that is required by its children nodes to update their primary variables. In the t -th upward update node i updates its secondary variables to obtain $^{t+1}\bar{x}_i$. It then sends its upward message containing variables $^{t+1}\tilde{x}_{\mathcal{P}(i)}^i$ and Lagrangian multipliers $^t\eta_{\mathcal{P}(i)}^i$.

The decentralised nature of the process is evident from the definition of the updates at the node level. We need to show that the process is in fact equivalent to performing the

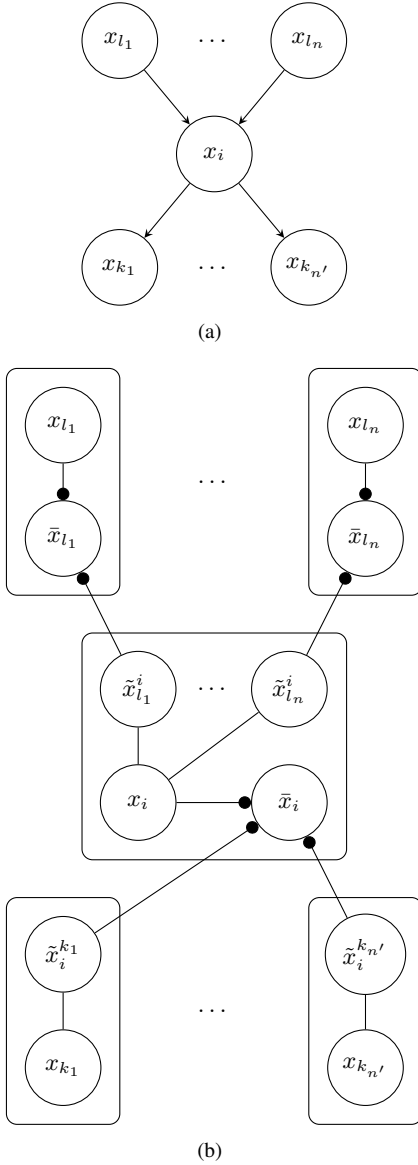


Fig. 15. Decentralisation step of DADMM. In (a), node i is shown with its parents and children in a DAG. It is assumed that node i has constraints that include all of its parents. By transforming the network into that of (b), each of node i 's parents only needs to communicate with node i given that they are not coupled elsewhere.

centralised version of ADMM on the entire system. A proof of this equivalence is provided in Theorem 2 following Lemma 2.

Lemma 2. *In Algorithm 1, every downward update t of node i is followed by an upward update t . Moreover, every upward update t of node i is followed by a downward update $t + 1$.*

Proof: From Lemma 1, we know that if node i has just performed the t -th update then it has performed $t - 1$ upward updates. Now, suppose that the next update is the $t + 1$ -th downward update. This is impossible because according to Lemma 1, node i would have performed t upward updates. The second part of the statement can be proved through a similar argument. ■

Theorem 2. *For each node i , after the t -th update of the*

Lagrangian multipliers in the downward update, the variables owned by the node are equal to the ADMM update t of those variables.

Proof: The proof is by induction. Before the start of the algorithm, the variables of node i are set to $^1x_i, ^1\bar{x}_i, ^1\tilde{x}_{\mathcal{P}(i)}^i, ^0\lambda_i, ^0\mu_{ik}$ and $^0\eta_{\mathcal{P}(i)}^i$. During the first downward update, node i 's Lagrangian variables are updated according to (31). The node would have received $^1\bar{x}_{\mathcal{P}(i)}$ from its parents' downward messages. At this stage, all variables belong to the ADMM update at $t' = 1$.

$$\begin{aligned} ^1\lambda_i &:= ^0\lambda_i + g(^1x_i, ^1\tilde{x}_{\mathcal{P}(i)}^i) \\ ^1\mu_i &:= ^0\mu_i + (^1x_i - ^1\bar{x}_i) \\ ^1\eta_{\mathcal{P}(i)}^i &:= ^0\eta_{\mathcal{P}(i)}^i + (^1\tilde{x}_{\mathcal{P}(i)}^i - ^1\bar{x}_{\mathcal{P}(i)}) \end{aligned} \quad (31)$$

Assume that after node i 's $t - 1$ -th Lagrangian update, all variables belong to the ADMM update at $t' = t - 1$. We now prove the statement for $t' = t$. After the $t - 1$ -th Lagrangian variable update, node i directly updates its primary variables according to (32).

$$\begin{aligned} (^{t+1}x_i, ^{t+1}\tilde{x}_{\mathcal{P}(i)}^i) &:= \\ \operatorname{argmin}_{x_i, \tilde{x}_{\mathcal{P}(i)}^i} \mathcal{L}_i^1(x_i, \tilde{x}_{\mathcal{P}(i)}^i, ^t\bar{x}_i, ^t\bar{x}_{\mathcal{P}(i)}, ^t\lambda_i, ^t\mu_i, ^t\eta_{\mathcal{P}(i)}^i) \end{aligned} \quad (32)$$

According to Lemma 2, the downward update is followed by an upward update. In the upward update, node i would have received $^{t+1}\tilde{x}_i^{C(i)}$ from its children as well as the corresponding Lagrangian variables $^t\eta_i^{C(i)}$. After receiving these variables, node i updates the secondary variables according to (33).

$$^{t+1}\bar{x}_i := \operatorname{argmin}_{\bar{x}_i} \mathcal{L}_i^2(^{t+1}x_i, ^{t+1}\tilde{x}_i^{C(i)}, \bar{x}_i, ^{t-1}\mu_i, ^t\eta_i^{C(i)}) \quad (33)$$

The minimisation over \bar{x}_i can be written explicitly as shown in (34).

$$\begin{aligned} ^{t+1}\bar{x}_i &:= \\ \frac{1}{|\mathcal{C}(i)| + 1} &\left[^{t+1}x_i + ^t\mu_i + \sum_{k \in \mathcal{C}(i)} [^{t+1}\tilde{x}_i^k + ^t\eta_i^k] \right] \end{aligned} \quad (34)$$

Finally, from Lemma 2, we have given that the upward update is followed by downward update $t + 1$ during which the Lagrangian variables are updated in an analogous manner to (31) to obtain $^{t+1}\lambda_i, ^{t+1}\mu_{ik}$ and $^{t+1}\eta_{\mathcal{P}(i)}^i$. Hence, the $t + 1$ -th ADMM update is complete. ■

E. Analysis

In this section, we analyse the computational complexity of one iteration of DADMM. Analysis of the full problem depends on its convergence rate, which we consider for the special case of threshold-DIF in Sec. VI.

First, we define the following terms:

$$\begin{aligned}
n_g^i &: \text{number of constraints for node } i \\
n_g^{\max} &:= \max_{i \in V} n_g^i \\
n_i &:= |x_i| = |\bar{x}_i| \\
n'_i &:= |\tilde{x}_{\mathcal{P}(i)}^i| \\
n_{\max} &:= \max_{i \in V} n_i
\end{aligned} \tag{35}$$

The worst-case complexity of one iteration of DADMM is given by Theorem 3.

Theorem 3. *Each iteration of DADMM over a DAG $G = \{V, E\}$ runs in $\mathcal{O}(\kappa(G)(n_{\max}|V|)^2(n_g^{\max} + n_{\max}|V|))$ time.*

Proof: Each ADMM iteration consists of three updates: the primary update, secondary update and the Lagrangian update. Since the algorithm is decentralised and synchronous, its running time is dominated by the time to update a single node multiplied by the depth of G .

We now develop an upper bound on the computation performed by an arbitrary node i . The primary update involves solving $\nabla_{\hat{x}_i} L = 0$. This equation is a linear equation, since the Lagrangian is quadratic and has the form $A\hat{x}_i = d$, where $A \in \mathbb{R}^{(n_i+n'_i)^2}$ and $d \in \mathbb{R}^{n_i+n'_i}$ since the primary vector \hat{x}_i has dimension equal to $n_i + n'_i$. Each element in A potentially contains a term from the objective and a term from each constraint in which the corresponding element in \hat{x}_i is involved. Each element of \hat{x}_i is involved in an equality constraint (with the secondary variables) and may appear in the constraints of g_i . Therefore, the elements of A can be computed in $\mathcal{O}((n_i + n'_i)^2(2 + n_g^i)) = \mathcal{O}((n_i + n'_i)^2 n_g^i)$ time. Each element in d can include a secondary variable and a Lagrangian multiplier from each constraint. The time complexity of computing the elements of d is $\mathcal{O}((n_i + n'_i)(1 + n_g^i))$. Solving for \hat{x}_i , assuming a dense matrix A , takes $\mathcal{O}((n_i + n'_i)^3)$ time. Therefore, the time complexity of the primary update is $\mathcal{O}((n_i + n'_i)^2(n_i + n'_i + n_g^i))$.

The secondary update computes an average over x_i and $\tilde{x}_i^{C(i)}$ to obtain \bar{x}_i . Thus, its time complexity is $\mathcal{O}(n_i(1 + |\mathcal{C}(i)|)) = \mathcal{O}(n_i|\mathcal{C}(i)|)$.

The Lagrangian update involves an evaluation of all constraints given by Equations (28), (29) and (30). The evaluation of each element of g_i involves at most $n_i + n'_i$ variables. The evaluation of the other constraints involves two variables each. Therefore, the time complexity of the Lagrangian update is $\mathcal{O}(n_g^i(n_i + n'_i) + n_i + n'_i) = \mathcal{O}(n_g^i(n_i + n'_i))$.

The total time complexity is the sum of these three updates. Thus, we have $\mathcal{O}((n_i + n'_i)^2(n_i + n'_i + n_g^i)) + \mathcal{O}(n_i|\mathcal{C}(i)|) + \mathcal{O}(n_g^i(n_i + n'_i)) = \mathcal{O}((n_i + n'_i)^2(n_i + n'_i + n_g^i) + n_i|\mathcal{C}(i)|)$.

We now restate the bound as a function of n_{\max} , n_g^i and $|V|$. Since n_{\max} is an upper bound for n_i , $n_{\max}|\mathcal{P}(i)|$ is an upper bound on n'_i . An upper bound for the number of children or the number of parents is simply the number of nodes $|V|$. Hence, a more conservative upper bound for n'_i is $n_{\max}|V|$. Therefore, the overall time complexity for the work performed by node i can be rewritten as $\mathcal{O}((n_{\max}|V|)^2(n_g^{\max} + n_{\max}|V|))$. The total time complexity for one iteration of the algorithm is thus $\mathcal{O}(\kappa(G)(n_{\max}|V|)^2(n_g^{\max} + n_{\max}|V|))$. ■

VI. THRESHOLD-DIF

In this section, we show how DADMM can be applied to the threshold-DIF problem. The mapping from threshold-DIF to the general problem formulation of DADMM is presented in detail. A detailed complexity analysis of DADMM in terms of the threshold-DIF problem size is provided. We present several experimental results in simulation, including a 15-node example and a Monte Carlo simulation, and results from an experimental system with two outdoor ground robots and a stationary camera.

A. Threshold-DIF Using DADMM

The threshold-DIF problem can be solved using DADMM. However, we first must reformulate constraints (4) and (7) such that they are compatible with the DADMM framework.

The *maximum* function in (4) is replaced by the set of inequalities (39). The maximum function is non-smooth and cannot be optimised in one step. With the set of inequalities in (39), the objective and all equality and inequality constraints of the optimisation become linear as required by DADMM.

The set of inequalities in (39) is equivalent to the maximum relation in (4) as long as one of the constraints is active. One constraint will always be active for h_{ik}^m if the link cost c_{ik}^m is positive. From the problem formulation, we know that the link cost c_{ij}^m can only be negative if $j \in V_e$, i.e. if the link is incident to an estimator j . For these links, we replace the set of inequalities in (39) with the set of equalities given in (36).

$$\begin{aligned}
x_{ik}^m(j) &= 0, & \text{if } k \neq j \\
x_{ik}^m &= h_{ik}^m, & \text{if } k = j
\end{aligned} \tag{36}$$

The inter-link constraint (7) is replaced by (42) where \mathcal{S}_{is} is the set of links emanating from node i involved in the inter-link constraint s . The DADMM format only permits constraints between a node and its parents. Therefore, constraint (42) only applies between links from node i and links from node i 's parents. This condition is not restrictive since an extra link can be added between non-neighbouring nodes with inter-link constraints while retaining the directed acyclic property of the graph. The graph remains acyclic by preserving any ordering between the two nodes between which the extra link is added. Since the network is a connected DAG, then, by definition, for any two nodes i and k either k is a successor of node i , $k \in \bar{\mathcal{C}}(i)$ or k is an ancestor of i , $k \in \bar{\mathcal{P}}(i)$ or neither. If $k \in \bar{\mathcal{C}}(i)$, the link should extend from i to k . If $k \in \bar{\mathcal{P}}(i)$, the link should extend from k to i . If there is no directed path between the nodes, then either direction retains the acyclic property.

After transforming the constraints, we obtain the problem (37-42) shown below.

$$\text{minimise} \quad \sum_{(i,k) \in E} c_{ik}^m h_{ik}^m \quad (37)$$

$$\text{subject to} \quad x_{ik}^m(j) \geq 0 \quad (38)$$

$$x_{ik}^m(j) \leq h_{ik}^m \quad (39)$$

$$\sum_{l \in \mathcal{P}(i)} x_{li}^m(j) - \sum_{k \in \mathcal{C}(i)} x_{ik}^m(j) + r_i^m(j) = 0 \quad (40)$$

$$\sum_m \nu_{ik}^m h_{ik}^m \leq C_{ik} \quad (41)$$

$$\sum_{k \in \mathcal{S}_{is}} \sum_m \nu_{ik}^m h_{ik}^m + \sum_{l \in \mathcal{P}(i)} \sum_{k' \in \mathcal{S}_{ls}} \sum_m \nu_{lk'}^m h_{lk'}^m \leq K_s \quad (42)$$

To solve threshold-DIF, all that is required at this stage is that the threshold-DIF variables and constraints be mapped to the variables of the distributed optimisation problem (23-24). This can be done as follows. The sets of variables $\{h_{ik}^m : m \in V_s, k \in \mathcal{C}(i)\}$ and $\{x_{ik}^m(j) : m \in V_s, j \in V_e, k \in \mathcal{C}(i)\}$ are mapped to x_i . The objective function f_i in (23) is represented by $\sum_{k \in \mathcal{C}(i)} \sum_m c_{ik}^m h_{ik}^m$ and the indicator functions resulting from the inequality constraints. The constraint g_i in (24) is represented by the equality constraints and the equality versions of the inequality constraints involving node i in (37-42).

DADMM runs continuously throughout system operation. As the system configuration changes, link costs and weights are updated with new values. To ensure convergence, the interval between updates is set to an adequate time period. In practice, this interval was found to be of similar length to that of min-cost-DIF in Sec. IV-C.

B. Analysis

In this section, we provide time complexity analysis of DADMM when applied to threshold-DIF. The complexity is expressed in terms of the size of the threshold-DIF input parameters.

Complexity analysis is provided for the entire optimisation process including the number of iterations required for convergence. We first determine the complexity of one iteration following directly from Theorem 3. We then find a bound on the number of iterations based on the algorithm's convergence rate.

The complexity of an DADMM iteration was determined in Sec. V-B. The complexity of one iteration in terms of threshold-DIF problem specification can be determined by substituting the appropriate values for n_{\max} and n_g^{\max} . To begin, we denote the number of sources, destinations and inter-link constraints in the network as follows:

- N_m : number of sources in the network.
- N_j : number of destinations in the network.
- N_s : number of inter-link constraints.

The maximum number of primary variables n_{\max} is proportional to the maximum number of routing variables which, in turn, is proportional to the number of sources multiplied

by the number of destinations multiplied by the number of children. The number of children is bounded from above by the number of nodes. Therefore, n_{\max} is bounded such that $n_{\max} \leq N_m N_j |V|$.

The maximum number of constraints n_g^{\max} is bounded by the maximum number of flow consistency constraints (40) and the maximum number of inter-link constraints (42). The number of consistency constraints is proportional to the number of sources multiplied by the number of destinations. Therefore, n_g^{\max} is bounded such that $n_g^{\max} \leq N_m N_j + N_s$.

Substituting the obtained bounds into the result of Lemma 3, the complexity of one iteration of DADMM for threshold-DIF becomes $\mathcal{O}(\kappa(G)(|V|^2 N_m N_j)^2 (|V|^2 N_m N_j + N_s))$. In threshold-DIF, the depth of the underlying graph is a function of processor cascading which is independent of the number of robots. Therefore, the depth is assumed to be constant. Hence, the time complexity of one iteration can be restated as $\mathcal{O}((|V|^2 N_m N_j)^2 (|V|^2 N_m N_j + N_s))$.

To determine the complexity of the whole optimisation process, the convergence rate is required. A convergence rate in an ergodic sense is established in (He and Yuan, 2012) with relatively mild assumptions.

The result is restated here after establishing the appropriate notation. Define the primal vector of the k -th iteration as $z^k = (z_1^k, z_2^k)$ where z_1^k and z_2^k are the ADMM primary and secondary vectors defined in Sec. V-A. Define the ergodic average $\bar{z}^k = \sum_{k'=1}^{k+1} z^{k'}$ and define z^* and y^* as the optimal primal and dual vectors. Then, if we assume that $z^0 = 0$ and $y^0 = 0$, the convergence result is given by (43). The positive constants α and β are independent of the dimension and value of both the primal and dual variables.

$$L(\bar{z}^k, y^*) - L(z^*, y^*) \leq \frac{\alpha \|z^*\|^2 + \beta \|y^*\|^2}{(k+1)} \quad (43)$$

From (43) it is clear that in order to obtain a bound on the convergence rate, we need to find an upper bound on the norm of the primal and dual optimal vectors. The absolute value of the elements in the primal vector have an upper bound u_z which follows from the problem definition in Sec. VI-A. Hence, an upper bound on the norm of the primal vector is given by (44).

$$\|z^*\|^2 \leq n_z u_z^2 \quad (44)$$

We now seek a bound for the norm of the dual vector $\|y^*\|^2$. To simplify the analysis, we note that the centralised ADMM version of the threshold-DIF problem has the form of the optimisation problem defined in (45) where the indicator function $I_{\geq 0}$ is defined in (46) below. The set \mathcal{I} contains the indices of the variables added to convert any inequality constraint into an equality constraint as described in Sec. V-B. These variables need to satisfy the inequality constraint $z_{(i)} \geq 0$ and they only appear in one row of the set of equality constraints $Az = b$.

$$\begin{aligned} \text{minimise} \quad & c^T z + \frac{\rho}{2} \|Az - b\|^2 + \sum_{i \in \mathcal{I}} I_{\geq 0}(z_{(i)}) \\ \text{subject to} \quad & Az = b \\ & A \in \mathbb{R}^{n_g \times n_z}, b \in \mathbb{R}^{n_g} \end{aligned} \quad (45)$$

$$I_{\geq 0}(z_{(i)}) = \begin{cases} 0 & \text{if } z_{(i)} \geq 0 \\ \infty & \text{otherwise} \end{cases} \quad (46)$$

An upper bound on $\|y^*\|^2$ can be obtained from the following lemma.

Lemma 3. *Assume that the equality constraints and the inequality constraints $z_i \geq 0$ active at x^* are all linearly independent. Then, there exists a positive constant γ independent of z , n_z and n_g such that $\|y^*\|^2 \leq \gamma \|c\|^2$.*

Proof: At optimality, we have $Az - b = 0$ and zero belongs to the subdifferential of the Lagrangian as shown in (47). The i -th element of the vector $b_I \in \mathbb{R}^{n_z}$ is defined in (48) where $\partial I_{\geq 0}$ is the subgradient of the non-smooth indicator function.

$$0 \in c + A^T y^* + b_I \quad (47)$$

$$b_{I(i)} = \begin{cases} \partial I_{\geq 0} & \text{if } i \in \mathcal{I} \text{ and the constraint } z_i \geq 0 \text{ is active} \\ 0 & \text{otherwise} \end{cases} \quad (48)$$

The subgradient $\partial I_{\geq 0}$ evaluated at 0 is an unbounded set and therefore cannot be used to directly bound y^* . However, the optimality condition (47) has n_z rows while y^* has dimension n_g . Furthermore, the maximum number of active inequality constraints, i.e. constraints where $b_{I(i)} \neq 0$, is equal to $(n_z - n_g)$ since otherwise, x^* would be over-defined by the constraints due to the linear independence assumption. Consequently, if all rows in (47) such that $b_{I(i)} \neq 0$ are removed, there will remain at least n_g rows.

To this end, we need to make sure that the matrix \bar{A}^T obtained after removing the rows from A^T remains full rank. When active, the inequality constraint $z_{(i)} \geq 0$ becomes $z_{(i)} = 0$. If this equality is augmented as a row vector to the matrix A , due to the linear independence assumption, the rank of A becomes $n_z + 1$. Through elementary row operations, any non-zero element on the column corresponding to $z_{(i)}$ can be changed to zero with no change in the rank of the matrix. At this stage, the row $z_{(i)} = 0$ can then be removed with the rank of the matrix dropping back to n_g . Once the row is removed, the column corresponding $z_{(i)}$ column is now all zeros and can hence be removed with no change in rank. This proves that \bar{A}^T has full rank n_g .

Therefore, the optimality condition (47) can be restated as (49) where \tilde{c} is the vector obtained after removing all the corresponding rows from c . The vector b_I becomes a zero vector after removing these rows.

$$\bar{A}^T y^* = -\tilde{c} \quad (49)$$

From (49) we obtain (50) where $\sigma_{\min}(\bar{A}\bar{A}^T)$ is the minimum eigenvalue of $\bar{A}\bar{A}^T$ and is greater than zero since \bar{A} is full rank.

$$\frac{\|\tilde{c}\|^2}{\|y^*\|^2} = \frac{y^{*T} \bar{A} \bar{A}^T y^*}{\|y^*\|^2} \geq \sigma_{\min}(\bar{A} \bar{A}^T) \quad (50)$$

The proof is established by setting $\gamma = 1/\sigma_{\min}(\bar{A}\bar{A}^T)$ and noting that $\|\tilde{c}\|^2 \leq \|c\|^2$ since \tilde{c} is obtained by removing elements from c . ■

We assume that all elements in c are upper-bounded by a constant value u_c . This is a reasonable assumption since c represents the link cost vector and only the relative cost is of importance. Consequently, from Lemma 3, we have the upper bound given in (51) for the norm of the dual vector.

$$\|y^*\|^2 \leq \gamma u_c^2 n_g \quad (51)$$

We can now state the main complexity result given by Theorem 4. The complexity is polynomial as expected since the problem is convex and the number of variables is polynomial in the number of nodes.

Theorem 4. *Obtaining an ϵ -optimal solution for the threshold-DIF problem using DADMM has a computational complexity of $\mathcal{O}((|V|^2 N_m N_j)^2 (|V|^2 N_m N_j + N_s) (|V|^3 N_m N_j + N_s) / \epsilon)$*

Proof: Bounds (44) and (51) mean that the left hand side of (43) is bounded by a constant weighted sum of n_z and n_g . Therefore, an upper bound on the number of iterations k required to produce an error ϵ is given as $\mathcal{O}((n_z + n_g)/\epsilon)$.

Note that the number of primary variables n_z is bounded by $\mathcal{O}(|V|^2 N_m N_j)$ multiplied by the number of nodes. Therefore, we have $n_z \leq \mathcal{O}(|V|^3 N_m N_j)$. The number of constraints n_g , on the other hand, can be bounded such that $n_g \leq \mathcal{O}(|V|^3 N_m N_j + N_s)$. Thus, the resulting number of iterations of the optimisation process is given by (52).

$$k \leq \mathcal{O}((|V|^3 N_m N_j + N_s)/\epsilon) \quad (52)$$

The complexity of the whole optimisation process is obtained by multiplying the number of iterations by the complexity of each iteration. Thus, for an ϵ -optimal solution, DADMM for threshold-DIF runs in $\mathcal{O}((|V|^2 N_m N_j)^2 (|V|^2 N_m N_j + N_s) (|V|^3 N_m N_j + N_s) / \epsilon)$ time. ■

C. Two-Robot Experiment

We implemented our approach and performed an experiment where two mobile robots communicate to track a moving target. The aim of this experiment is to show the information gain advantage of dynamic information flow in the case of limited inter-robot communication bandwidth.

1) *Experimental Setup:* The experimental system consists of two modified Segway RMP 400 robots. An image of the robots is shown in Fig. 16. The first robot is equipped with a Velodyne 3D Lidar with a 360° field of view. The second robot is equipped with a 2D SICK LMS291 horizontally mounted laser scanner with a 180° field of view. Although these sensors provide range as well as bearing, they were treated as bearing-only sensors to force cooperation between the robots. Each robot is also equipped with a server-class computer with an eight-core processor. For localisation, the two robots rely on high-accuracy Inertial Measurement Unit (IMU) and Differential Global Positioning System (DGPS) modules.



Fig. 16. The Segway RMP 400 robots used in the experiments.

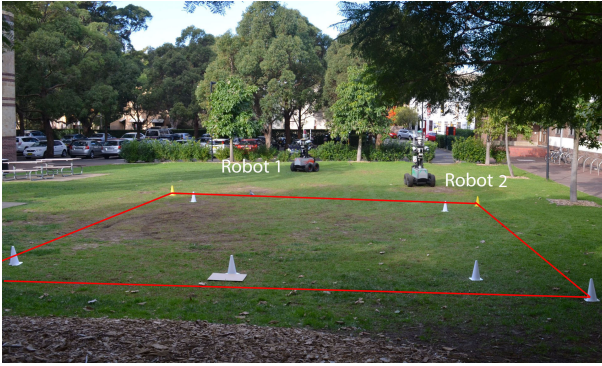


Fig. 17. The outdoor experimental setup, with robots visible outside the tracking region of interest. The border of the tracking region is designated by solid lines.

An image of the experimental setting is shown in Fig. 17. The two robots were placed in two separate areas with virtually bounded geographical regions to avoid collision. Target tracking was limited to a geographically bounded region of interest and for the purpose of the demonstration, tracking was limited to one target performing circular patterns.

The network diagram for this demonstration is shown in Fig. 18. It is assumed that maximum communication bandwidth is limited and does not allow both robots to send sensor data at the full rate. We also assume that communication throughput decreases with inter-robot distance. Therefore, the robots are required to share the available bandwidth. The bandwidth sharing constraint is indicated by the dashed lines drawn between the two inter-robot links. Virtual links, not shown in the figure, allow for the no-send decision. It is expected that through dynamic information flow the bandwidth will be shared efficiently with respect to sensor utility. The maximum flow assumption is valid in this scenario since no processing costs are assigned.

To validate the performance of dynamic information flow, two control tests were run for the purpose of comparison. The first control test allows unconstrained communication between all nodes and shall be referred to as the *unconstrained*

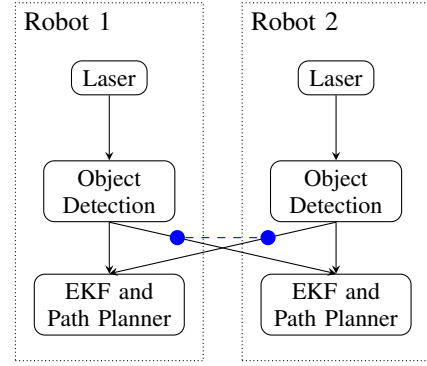


Fig. 18. The threshold-DIF diagram for the two-robot scenario. Virtual links are omitted for clarity.

TABLE III
AVERAGE FLOW RATES AND SENSOR UTILITY FOR THE TWO-ROBOT EXPERIMENT.

| Link | Flow Rate | Sensor Utility |
|--------------------|-----------|----------------|
| Robot 1 to Robot 2 | 44.80 | 3.42 |
| Robot 2 to Robot 1 | 10.30 | 2.10 |

case. This test mainly acts as a benchmark since it violates bandwidth constraints. The second control test involves a reduced communication rate that obeys bandwidth bounds. This test shall be referred to as the *down-sampled* case.

2) *Results*: The information value for the robots' target estimates over time is shown in Fig. 19 with the corresponding average bars shown in Fig. 20. In both figures, we observe minimal difference in information value across the three communication methods for Robot 1. Because Robot 1 has a 360° field-of-view sensor, the target is always visible and therefore tracking does not depend on observations received from Robot 2. However, we do observe a difference in information value for Robot 2. Fig. 20(b) shows that the down-sampled method results in reduced information gathering performance when compared to our method. This effect is also evident in Fig. 19(b) where there is a clear decline in information for the down-sampled case at times 20 and 60. This decline occurs because the target drops outside the robot's sensor field-of-view. Dynamic flow ensured that information was directed from Robot 1 to Robot 2, but down-sampling naively shared the communication medium.

The advantage of dynamic information flow is further confirmed in Fig. 21. The bottom two plots show the approximate sensor observation utilities and data flow between robots over time for the dynamic flow case. The flow from Robot 1 to Robot 2 dominates bandwidth usage when the target is not in Robot 2's sensor field-of-view. However, at times 40 and 80, the flow was directed from Robot 2 to Robot 1 because the target was closer to Robot 2. The top plot shows the distance between the robots over time. At the average inter-robot distance, the available bandwidth is limited to approximately half of the maximum bandwidth. As expected, the sum of the information flows obeys this reduced capacity. The flow rate and sensor utility averages are shown in Table III.

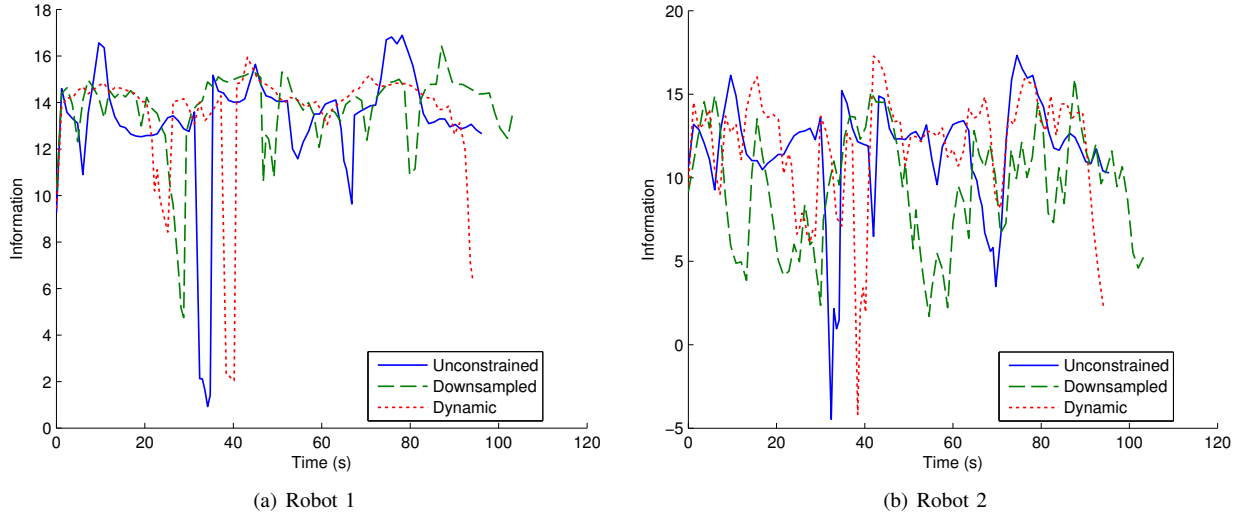


Fig. 19. The information value (negative entropy) of the robots' target estimate for the two-robot hardware experiment. Plots shown are for all three communication methods: unconstrained, down-sampled and dynamic. The sudden drops in information are due to target loss.

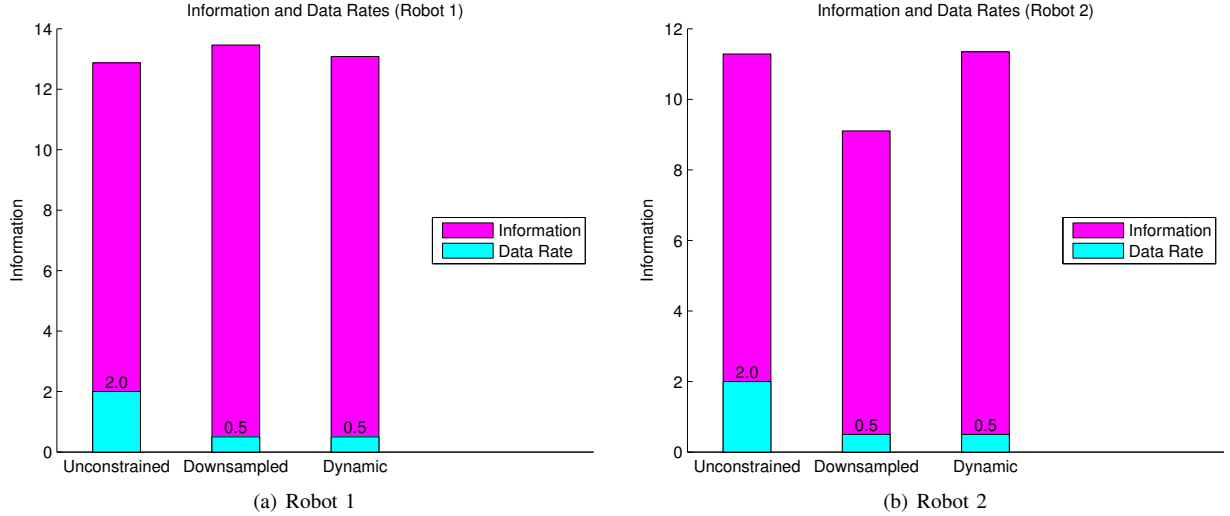


Fig. 20. Time averages of the plots in Fig. 19 shown in bar format. Data rates are shown superimposed. The dynamic case shows a clear improvement in information gain in comparison to down-sampling for Robot 2.

D. Three-Sensor-Node Experiment

We also evaluated our approach in a more complex scenario. This scenario involves the two robots from the previous experiment with the addition of a stationary camera. The aim of this three-sensor-node scenario is to show the generality of our method and to emphasise the multicast behaviour of dynamic information flow.

1) *Experimental Setup:* In this scenario, the two robots are aided in tracking by a Prosilica GC2450 camera acting as a bearing-only sensor. The camera is positioned outside the tracking region of interest opposite the robots. The camera sends raw images to Robot 2 which processes the images and shares the observations with Robot 1. Hence, wireless communication is required for three links: 1) the link from the camera to Robot 2, 2) the link from Robot 1 to Robot 2 and 3) the link from Robot 2 to Robot 1. In the experiment, the robots remained stationary. This does not affect the results of

the demonstration since separation distance is ignored in this scenario. Simulation results with moving robots in a similar setup are shown later in Sec. VI-E.

The network diagram is shown in Fig. 22. The experiment assumes that available bandwidth is sufficient for the two mobile robots to share observations. However, if images are received from the static camera then the wireless network becomes congested. The camera provides accurate tracking when the target is in its proximity and inside its field-of-view. Dynamic flow is expected to allow images to be sent from the camera in such a situation. The extra flow from the camera has to be accompanied by a simultaneous reduction of flow in the other links sharing the medium.

All three communication methods were tested. In the unconstrained communication case, images from the camera were sent to Robot 2 for processing and caused congestion in the wireless network. The bandwidth limits on the wireless

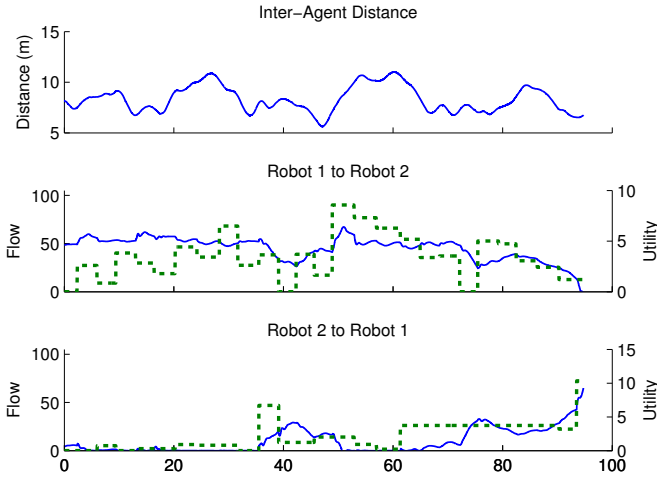


Fig. 21. Inter-robot flow rates and sensor utility over time for the dynamic flow case of the two-robot experiment. In the first plot, the inter-robot distance is shown. In the lower plots, flow rate is shown as solid lines and sensor utility is shown as dotted lines. Flow rate varies with utility and available bandwidth varies with inter-robot distance.

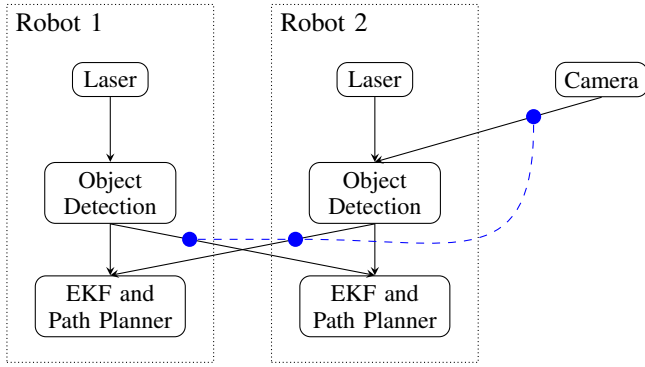


Fig. 22. The threshold-DIF diagram for the three-sensor-node scenario. Virtual links are omitted for clarity.

network effectively reduced the image transfer rate and consequently diminished the advantage of the unconstrained case.

2) *Results*: The information value for each of the robots' estimates is plotted over time in Fig. 24 with the corresponding average bars shown in Fig. 25. The figures show that the dynamic case outperforms down-sampling for both robots. They also show better performance for the dynamic case in comparison to the unconstrained case for Robot 1, since the unconstrained case would have failed to produce the desired communication due to infrastructure bandwidth limitations. The results for Robot 2 show similar performance between the dynamic case and the unconstrained case.

One of the main objectives of this experiment is to highlight the multicast behaviour. Multicast behaviour can be observed by analysing the bottom two plots of Fig. 23. At times 80, 150 and 190, the flow from the camera to Robot 2 retains a high value even though the camera utility for Robot 2 is low during those times. Robot 2 receives these observations without inducing additional cost since observations destined to Robot 1 are processed on-board Robot 2. Robot 1 receives these observations due to their high utility for Robot 1 but

TABLE IV
AVERAGE FLOW RATES AND SENSOR UTILITY FOR THE THREE-SENSOR-NODE EXPERIMENT.

| Link | Flow Rate | Sensor Utility |
|-----------------------------|-----------|----------------|
| Robot 1's Sensor to Robot 2 | 51.34 | 2.85 |
| Robot 2's Sensor to Robot 1 | 24.68 | 0.50 |
| Camera to Robot 1 | 26.29 | 3.40 |
| Camera to Robot 2 | 24.89 | 1.88 |

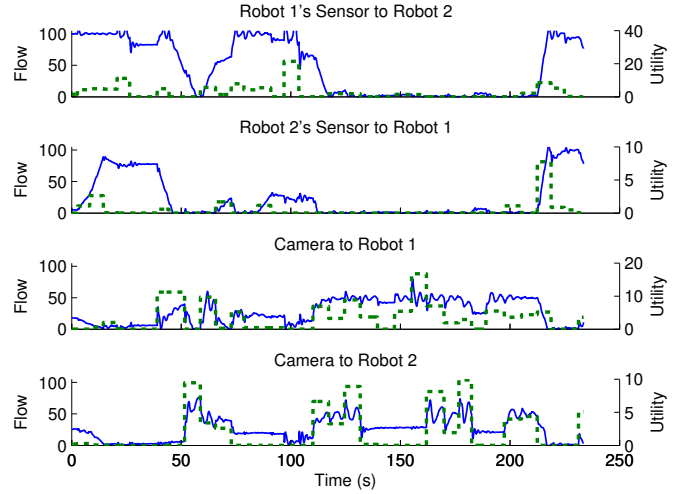


Fig. 23. Inter-robot information flow and sensor utility over time for the dynamic flow case of the three-sensor-node experiment. Flow rates are shown in solid lines; sensor utility is shown in dashed lines. Based on the bandwidth constraints and the camera's data rate, 50 is the maximum flow available for the data sourced from the camera. At approximately 120 seconds, communication from the camera interrupts communication between robots due to the increased utility of camera observations.

these observations must be processed on-board Robot 2 according to the system architecture. Multicast routing ensures that such flow does not get tallied twice. The top two plots confirm that the system obeys the bandwidth limits as the drop in flow takes place concurrently with the rise in flow of from the camera. It should be noted that the maximum possible flow from the off-board stationary camera is only 50 units according to bandwidth bounds. The flow rate and sensor utility averages are shown in Table IV.

These results also validate the maximum flow approximation of total flow. The flow in each of the inter-robot links shown in the top two plots in Fig. 23 holds data to only one destination. Therefore, there is no error arising from the maximum flow approximation for those links. The link from the camera holds data destined to both robots. At instances when the camera link is at zero or at maximum flow, there is no loss due to the maximum flow approximation. Also, the number of destinations in this case is two and hence the loss calculated from (19) is at most a 25% of maximum flow for all other instances.

E. Three-Sensor-Node Simulation

We repeated the experiment presented in Sec. VI-E in simulation. Here, we allow robots to move in order to improve

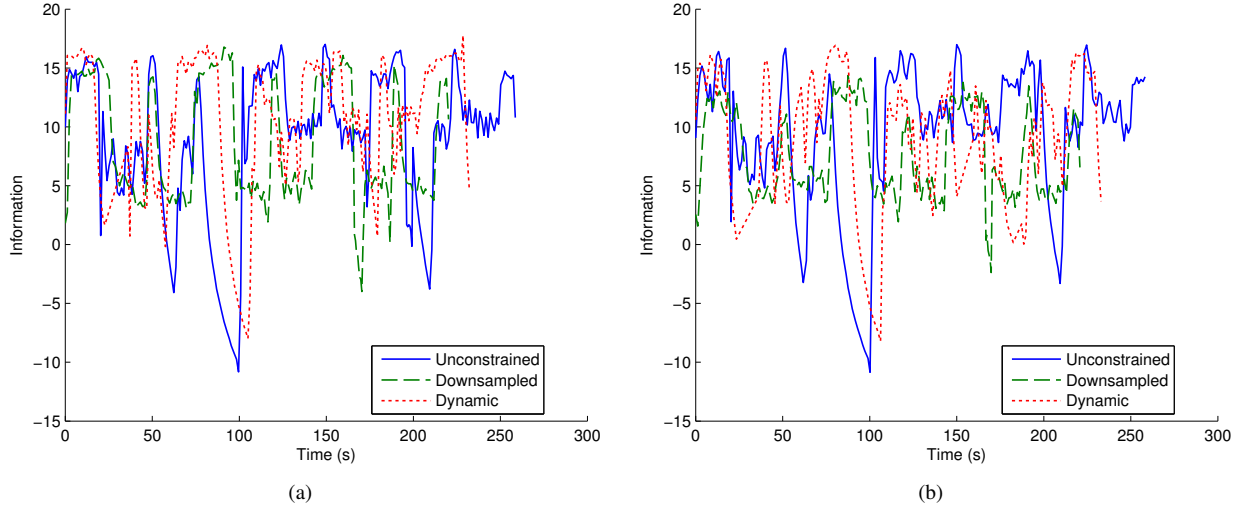


Fig. 24. The information value (negative entropy) of the robots' target estimate for the three-sensor-node experiment. Plots are shown for all three communication methods: unconstrained, down-sampled and dynamic.

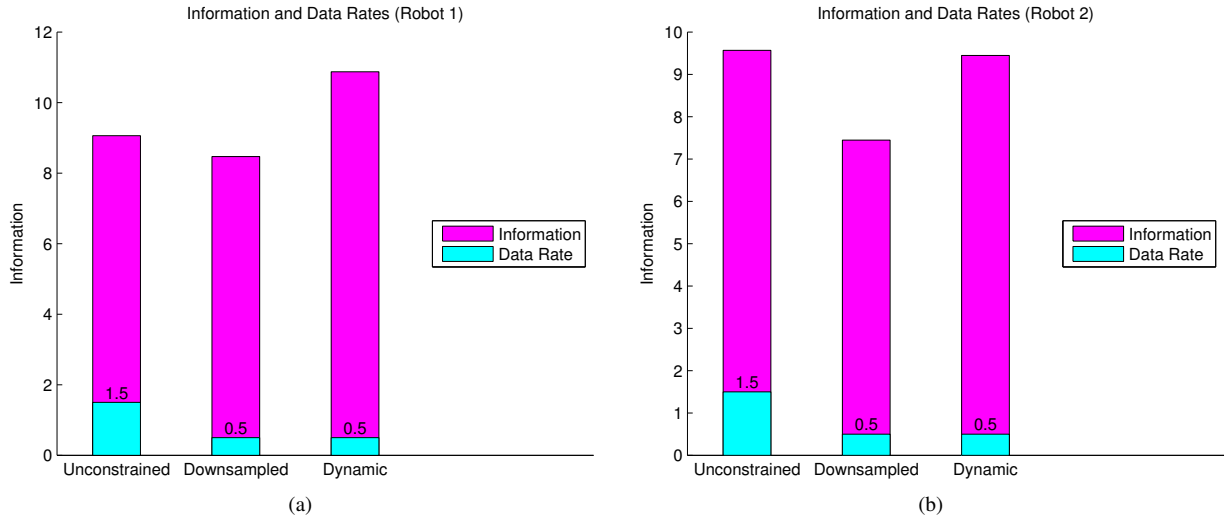


Fig. 25. Time averages of the plots in Fig. 24 shown in bar format. Data rates are shown superimposed. The dynamic information flow case has high average information gain performance with low average data rate.

tracking performance.

1) *Experimental Setup*: The experimental setting is similar to that of Sec. VI-D. Sensor output was only simulated through point observations. Therefore, no raw images were involved and the raw-data communication rate was fictitious. In a similar manner to the hardware case, communication is required for the links between the robots as well as the link from the camera.

The network diagram for this experiment is the same as shown earlier in Fig. 22. Through dynamic information flow, the camera is expected to selectively interrupt communication between the two robots in order to send its images based on the benefit of its observations as evaluated by the robots.

The simulation was run for all three communication methods. The unconstrained communication method is naturally expected to produce higher information gain since the bandwidth bounds are not enforced due to the fictitious data rates.

TABLE V
AVERAGE FLOW RATES AND SENSOR UTILITY FOR THE THREE-SENSOR-NODE SIMULATION.

| Link | Flow Rate | Sensor Utility |
|-----------------------------|-----------|----------------|
| Robot 1's Sensor to Robot 2 | 89.74 | 3.01 |
| Robot 2's Sensor to Robot 1 | 81.21 | 1.14 |
| Camera to Robot 1 | 5.33 | 1.50 |
| Camera to Robot 2 | 6.79 | 0.94 |

2) *Results*: The bottom two plots shown in Fig. 26 highlight an aspect of multicast behaviour different to that highlighted by the hardware analogue of this simulation. When only one robot evaluates a higher utility for camera observations, such as at times 120, 150 and 220, no significant change in flow is observed. However, when both robots evaluate an improvement in the utility, the increase in the flow from the

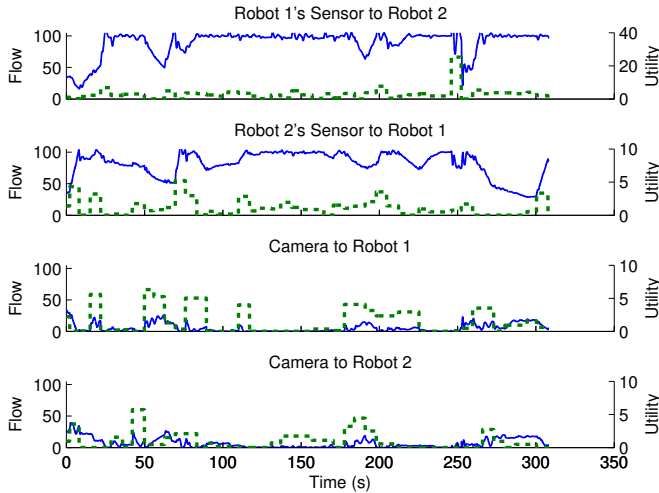


Fig. 26. Information flow and sensor utility for the three-sensor-node simulation. Flow rates and sensor utility are represented as in Fig. 23. The maximum flow rate available for the data sourced from the camera is assumed to be 50. Flow rates vary with sensor utility.

camera can be clearly seen.

The information value of the robots' target estimates over time is shown in Fig. 27 with the corresponding average bars shown in Fig. 28. As expected, unconstrained communication results in higher information on average. However, this communication setting violates the bandwidth bounds. Nevertheless, Fig. 28 shows that the dynamic case outperforms the down-sampled case for both robots. Figure 27 shows that the dynamic case dominates the down-sampled case at time 70, between times 150 and 200 and between times 250 and 300. Observations were received from the camera at these times for the dynamic flow case (as can be seen in Fig. 26). These times correspond to the configuration where the target enters the camera's field-of-view and becomes closer to the camera than the mobile robots. The flow rate and sensor utility averages are shown in Table V.

Similar to the hardware case, we note there is no error arising from the maximum flow assumption in the links between the two robots. The loss occurring in the link from the camera is negligible.

F. Multiple-Node Simulation

We demonstrated our decentralised algorithm on a simulated 15-node threshold-DIF network. The purpose of the simulation is to demonstrate our approach for a problem that is not amenable to manually designed communication protocols.

The simulated network comprises five agents each equipped with a sensor, a processor and an estimator. The network has a fully connected topology such that each sensor is connected to all processors and each processor is connected to all estimators. The agents are spatially distributed evenly in a linear manner. Each sensor is assumed to produce data at a rate of 100 units. A global communication constraint of 500 units was applied. In addition, a per-link capacity constraint of 200 units was applied to each processor-estimator link.

Sensor utilities were externally randomised and provided to the estimators.

Figure 29 displays, in chronological order, the routing state of the network at various time instances throughout the simulation. Each column represents one agent equipped with a sensor, processor and estimator. This configuration is a matter of choice rather than a restriction of the algorithm. The links shown in the figure represent those that carried more than 10 units of data during the simulation. The figure demonstrates the shift of flow from one part of the network to another as the sensor utilities change. More importantly, the routing states shown would be difficult to determine based merely on intuition.

G. Monte-Carlo Simulation

To validate the performance of dynamic information in comparison to down-sampling, we conducted a Monte Carlo simulation for the experimental setting of Sec. VI-C. The aim of the simulation is to analyse the statistical significance of performance improvement due to our solution to threshold-cost-DIF.

The results of the Monte Carlo simulation are shown in Fig. 30 in box-plot format. For each communication method, we ran twenty randomly initialised trials running for one minute each. The results shown assume each trial as one sample. Dynamic communication outperforms down-sampling with p -value less than 0.03 based on the Welch's t -test.

VII. DISCUSSION OF SENSOR UTILITY ESTIMATION

An essential component in addressing dynamic information flow is the estimation of sensor utility. In this section, we discuss our implementation of sensor utility estimation, its limitations and possibilities for improvement.

In our implementation, an estimator approximates sensor utility by evaluating the most recent sensor observation received. The value of the observation is computed as the reduction in entropy realised by fusing the observation into the estimator. This approach is advantageous due to the simplicity of implementation. Entropy reduction can be computed efficiently without additional data storage. The disadvantage of this approximation is that it is myopic; it only reflects the instantaneous effect of a sensor observation on the information gathering performance of a single robot. Myopic approximations such as this are commonly used since the long-term value of a sensor observation can be difficult to compute in the general case (Williamson et al., 2009). Alternatively, robots could learn the utility of an observation from other robots, as in (Xu et al., 2013a). This approach is suited to homogeneous systems with sufficient capacity for the extra computational overhead necessitated by the learning process.

For non-myopic sensor utility estimation, the value of a shared observation should be evaluated according to that observation's impact on the actions of other robots. Consider the following example where two moving robots track two moving targets. Suppose that the targets are far apart and each robot is only tracking one target. The observation from the first robot still induces an information gain in the second robot's

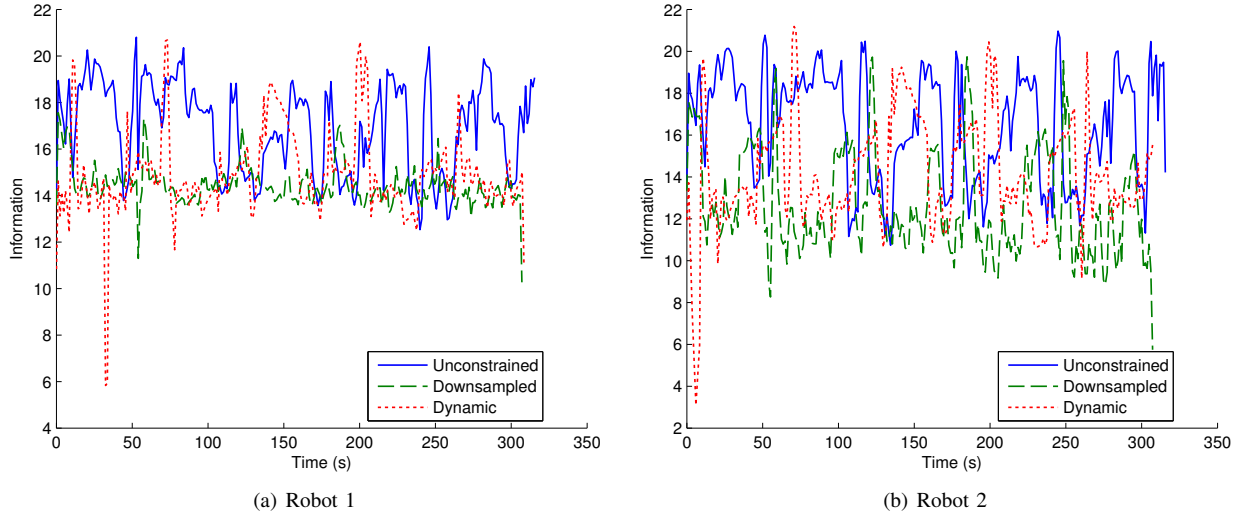


Fig. 27. The information value (negative entropy) of the robots' target estimate for the three-sensor-node simulation. Plots are shown for all three communication methods: unconstrained, down-sampled and dynamic.

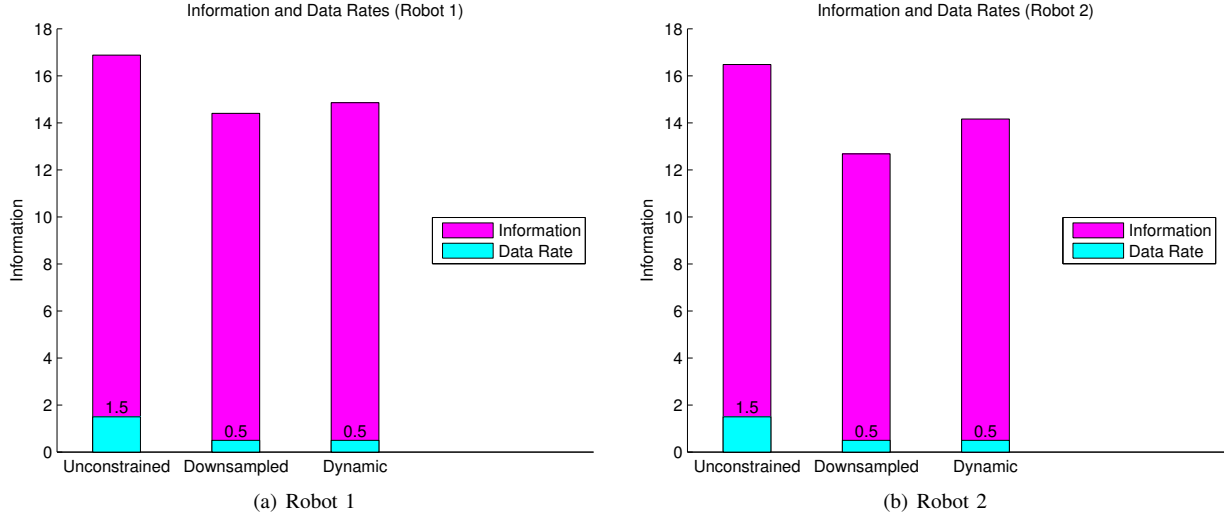


Fig. 28. Time averages of the plots in Fig. 27 shown in bar format. The data rates are shown superimposed. The improvement for Robot 2 achieved by dynamic communication over down-sampling using the same communication rates is clearly observed. The unconstrained method violates the bandwidth constraints and is only included as a benchmark.

estimator in this case. However, the observation is of little value to the second robot because this additional information does not affect the second robot's choice of viewpoint in the immediate future. The second robot can always receive the latest target estimate, which is an accumulation of past observations, from the first robot when needed. Therefore, in a non-myopic approach the value of an observation should not be based purely on information gain but rather on its effect on decision-making. Non-myopic estimation can improve the performance of DIF algorithms but is beyond the scope of the paper.

The submodularity of mutual information has been recently used to provide near-optimal solutions to the sensor selection problem (Golovin and Krause, 2011). However, the submodularity property does not hold for information gathering tasks with dynamic environments (Williams, 2007), which is the case of interest in this paper.

VIII. CONCLUSIONS AND FUTURE WORK

We have introduced the dynamic information flow problem and defined two variants, min-cost-DIF and threshold-DIF. We proposed an efficient decentralised solution for each variant that allows heterogeneous decentralised information gathering teams to dynamically decide when and where sensor information should be transmitted. We expect this work to be of value in a broad range of application areas in and beyond robotics. For example, agriculture and environmental monitoring robotics involves problems in coordination and estimation for multiple robots with high-data-rate sensors that would benefit from efficient use of limited communication bandwidth. Our work can also be beneficial to automated personnel and equipment tracking systems in geographically large areas such as ports and mines.

In min-cost-DIF, robots determine flow rates based on in-

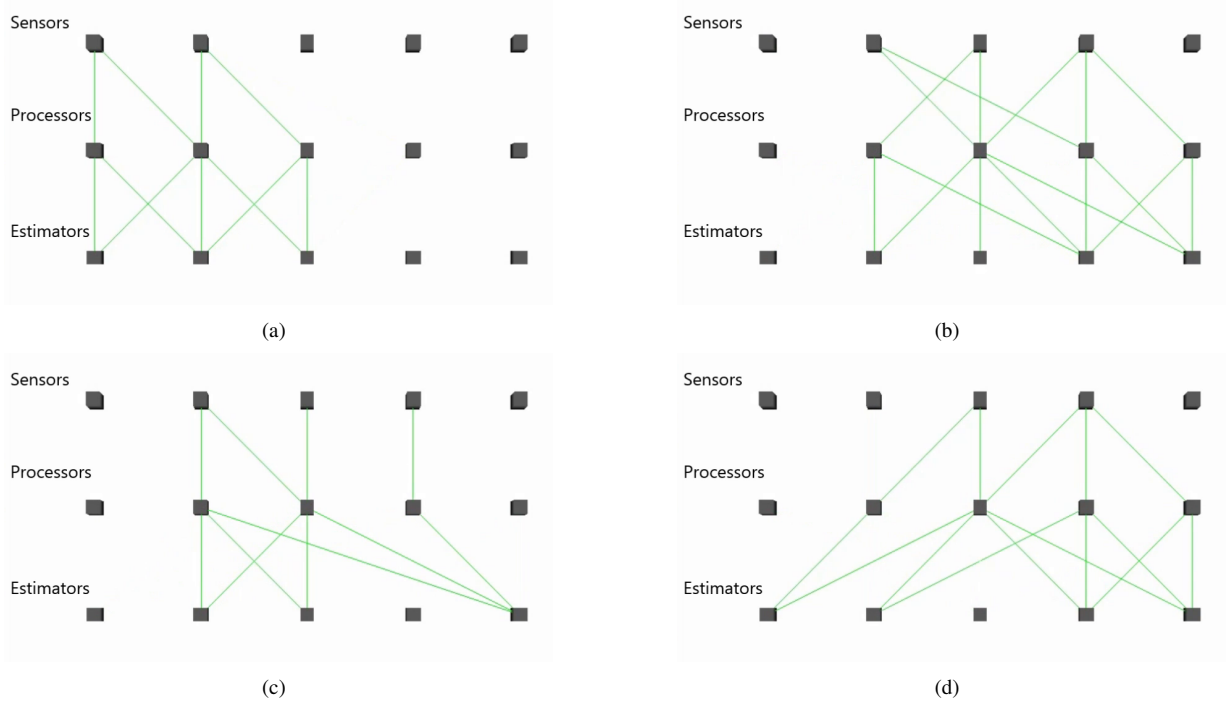


Fig. 29. Routing state of multiple-node simulation at various times depicting active wireless links. Each column represents one agent equipped with a sensor, processor and estimator. Active links are represented by green lines. A link is considered active if it holds more than 10 units of data flow.

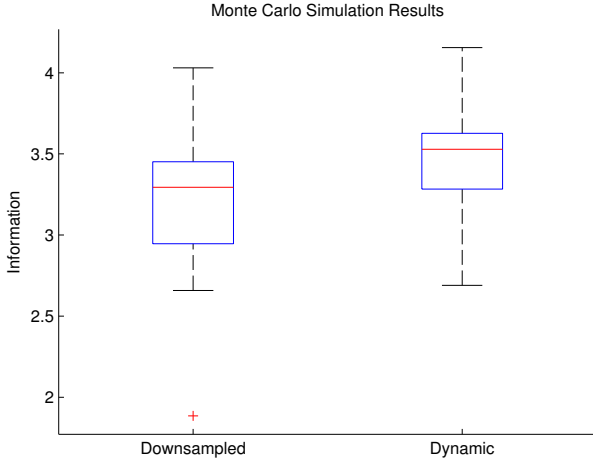


Fig. 30. Monte Carlo simulation results comparing down-sampling with dynamic information flow. Each method was tested on the two-robot scenario for twenty trials running for one minute each. In the results depicted, a trial acts as one sample. The box extents represent the first and third quartile, while the whiskers represent the extrema. The median is represented by the horizontal line inside the box.

formation value, communication costs and computation costs. Our solution to min-cost-DIF was adapted from recent results in multicast routing, which we extended to allow for negative link costs that represent sensor utility. This solution was demonstrated in simulation for a system of two mobile robots and was also demonstrated experimentally and in simulation for the case of one mobile robot with access to an off-board processing station. The scalability of our solution was empirically evaluated in a simulation with up to 28 nodes.

In threshold-DIF, flow rates are optimised based on the

value of information while obeying local computation limits and global communication limits. Our solution to threshold-DIF is based on a distributed version of ADMM that requires neighbour-to-neighbour communication only. We proved that the convergence time of our solution is polynomial in the size of the network. The benefits of this solution were demonstrated in simulation and hardware demonstrations with two mobile robots and two robots plus a stationary camera.

Our results have shown empirically that judicious adjustment of flow rates can improve information gathering performance. We focused on experimental systems with high-data-rate sensors that can easily overwhelm a standard wireless network. We have begun to explore scalability to larger networks in simulation, but the general issue of scalability and performance in large real-world networks (greater than 10 nodes) is an important area of future work.

Another interesting avenue to pursue is to consider further variants of the general DIF problem. For example, we would like to extend the cases considered in this paper, which address sensor-to-estimator communication, to include estimator-to-controller communication. This extension would introduce a new DIF variant for unified communication-aware information gathering and is of particular importance to applications that involve large amounts of data exchange for both estimation and control. The challenges of communication efficiency for control have traditionally been studied in the context of team decision theory in terms of information structure optimisation (Ho, 1980). The difficulty of information structure optimisation would complicate its integration with DIF, mainly due to the non-linearity of the control objective as a function of communication decisions. However, if communication costs

are readily specified, as assumed in the case of min-cost-DIF, then the integration of our previous communication-efficient control method (Kassir et al., 2012), which also takes communication costs as input, can be made possible. This integration would require simplifying assumptions such as the separation between estimation and control, for example.

The performance of our solutions to DIF could be further improved through an accurate non-myopic sensor utility estimate. A possible solution to non-myopic sensor utility estimation might include employing machine learning techniques that learn the sensor utility as a function of the platform state. It is also interesting to consider cases where sensor utility estimates change rapidly, and to explore the sensitivity of DIF algorithms to such high frequency changes.

Another desirable future advancement would be to extend our current distributed solution to threshold-DIF to obtain an any-time feasible solution. Any-time feasibility might be achieved through the adaptation of feasibility projection methods from optimisation theory.

In our current implementation, the identities of the nodes in the network need to be globally known. We envisage that this is not a necessary requirement and can be replaced by an automated discovery process. Nodes can advertise their types and then run a handshaking procedure with appropriate nodes before establishing connections.

ACKNOWLEDGEMENTS

This work is supported in part by the Australian Centre for Field Robotics and the New South Wales State Government. We would like to thank Professor Stephen Boyd for his helpful advice on ADMM. We would also like to thank James Underwood, Zhe Xu, Suchet Bargoti, Joseph Nguyen, Marcos Castro, Tim Patten and Mark Calleija for their help in conducting the experiments.

REFERENCES

- Ahlswede, R., Cai, N., Li, S.-Y., and Yeung, R. (2000). Network information flow. *IEEE Trans. Inform. Theory*, 46(4):1204–1216.
- Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Prentice hall.
- Bernstein, D. S., Givan, R., Immerman, N., and Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Math. Oper. Res.*, 27(4):819–840.
- Bourgault, F., Furukawa, T., and Durrant-Whyte, H. (2004). Decentralized bayesian negotiation for cooperative search. In *Proc. of IEEE/RSJ IROS*, Sendai, Japan.
- Boyd, S., Parikh, N., and Chu, E. (2011). *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Now Publishers Incorporated.
- Carlin, A. and Zilberstein, S. (2008). Value-based observation compression for DEC-POMDPs. In *Proc. of AAMAS*, Estoril, Portugal.
- Chen, L., Wang, Z., Szymanski, B., Branch, J. W., Verma, D., Damarla, R., and Ibbotson, J. (2010). Dynamic service execution in sensor networks. *Comput. J.*, 53(5):513–527.
- Chung, T. H., Burdick, J. W., and Murray, R. M. (2006). A decentralized motion coordination strategy for dynamic target tracking. In *Proc. of IEEE ICRA*, Orlando, USA.
- Cui, Y., Xue, Y., and Nahrstedt, K. (2007). Optimal distributed multicast routing using network coding. In *Proc. of IEEE ICC*, Glasgow, Scotland.
- Delle Fave, F., Rogers, A., Xu, Z., Sukkarieh, S., and Jennings, N. R. (2012). Deploying the max-sum algorithm for coordination and task allocation of unmanned aerial vehicles for live aerial imagery collection. In *Proc. of IEEE ICRA*, St. Paul, USA.
- Gan, S. K. and Sukkarieh, S. (2011). Multi-UAV target search using explicit decentralized gradient-based negotiation. In *Proc. of IEEE ICRA*, Shanghai, China.
- Ghaffarkhah, A. and Mostofi, Y. (2011). Communication-aware motion planning in mobile networks. *IEEE Trans. Automat. Contr.*, 56(10):2478–2485.
- Gmytrasiewicz, P. J. and Durfee, E. H. (2001). Rational communication in multi-agent environments. *Auton. Agent. Multi-Ag.*, 4(3):233–272.
- Goldman, C. V. and Zilberstein, S. (2003). Optimizing information exchange in cooperative multi-agent systems. In *Proc. of AAMAS*, Melbourne, Australia.
- Golovin, D., Faulkner, M., and Krause, A. (2010). Online distributed sensor selection. In *Proc. of ACM/IEEE IPSN*, Stockholm, Sweden.
- Golovin, D. and Krause, A. (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Intell. Res.*, 42:427–486.
- Gupta, P. and Kumar, P. (2000). The capacity of wireless networks. *IEEE Trans. Inform. Theory*, 46(2):388–404.
- Gupta, V., Chung, T. H., Hassibi, B., and Murray, R. M. (2006). On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260.
- He, B. and Yuan, X. (2012). On the $o(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.*, 50(2):700–709.
- Ho, Y.-C. (1980). Team decision theory and information structures. *Proc. IEEE*, 68(6):644 – 654.
- Hollinger, G. and Singh, S. (2012). Multirobot coordination with periodic connectivity: Theory and experiments. *IEEE Trans. Robot.*, 28(4):967–973.
- Hollinger, G., Singh, S., Djugash, J., and Kehagias, A. (2009). Efficient multi-robot search for a moving target. *Int. J. Robot. Res.*, 28(2):201–219.
- Hsieh, M. A., Cowley, A., Kumar, V., and Taylor, C. J. (2008). Maintaining network connectivity and performance in robot teams. *J. Field Robot.*, 25(1-2):111–131.
- Kassir, A., Fitch, R., and Sukkarieh, S. (2012). Decentralised information gathering with communication costs. In *Proc. of IEEE ICRA*, St. Paul, USA.
- Keshavarz-Haddadt, A. and Riedi, R. (2008). Bounds on the benefit of network coding: Throughput and energy saving in wireless networks. In *Proc. of IEEE INFOCOM*, Phoenix, USA.
- Kulik, J., Heinzelman, W., and Balakrishnan, H. (2002). Negotiation-based protocols for disseminating infor-

- mation in wireless sensor networks. *Wirel. Netw.*, 8(2/3):169–185.
- Kuo, V. and Fitch, R. (2014). Scalable multi-radio communication in modular robots. *Robot. Auton. Syst.*, 62(7):1034–1046.
- Lindhé, M. and Johansson, K. H. (2013). Exploiting multipath fading with a mobile robot. *Int. J. Robot. Res.*, 32(12):1363–1380.
- Malmirchegini, M. and Mostofi, Y. (2012). On the spatial predictability of communication channels. *IEEE Trans. Wireless Commun.*, 11(3):964–978.
- Molin, A. and Hirche, S. (2009). On LQG joint optimal scheduling and control under communication constraints. In *Proc. of IEEE CDC/CCC*, Shanghai, China.
- Mostofi, Y. (2009). Decentralized communication-aware motion planning in mobile networks: An information-gain approach. *J. Intell. Robot. Syst.*, 56(1-2):233–256.
- Ramanathan, S. (1996). Multicast tree generation in networks with asymmetric links. *IEEE/ACM Trans. Networking*, 4(4):558–568.
- Sadler, B. M., Rus, D., and Sukhatme, G. S., editors (2013). *Int. J. Robot. Res., Special Issue on Robotic Communications and Collaboration in Complex Environments*. SAGE Publications. 32(12).
- Semsar-Kazerooni, E. and Khorasani, K. (2009). Multi-agent team cooperation: A game theory approach. *Automatica*, 45(10):2205 – 2213.
- Smith, R. N., Schwager, M., Smith, S. L., Jones, B. H., Rus, D., and Sukhatme, G. S. (2011). Persistent ocean monitoring with underwater gliders: Adapting sampling resolution. *J. Field Robot.*, 28(5):714–741.
- Speyer, J., Seok, I., and Michelin, A. (2008). Decentralized control based on the value of information in large vehicle arrays. In *Proc. of ACC*, Seattle, USA.
- Stachura, M. and Frew, E. W. (2011). Cooperative target localization with a communication-aware unmanned aircraft system. *AIAA J. Guid. Control Dynam.*, 34:1352–1362.
- Twigg, J. N., Fink, J. R., Yu, P. L., and Sadler, B. M. (2013). Efficient base station connectivity area discovery. *Int. J. Robot. Res.*, 32(12):1398–1410.
- Williams, J. L. (2007). *Information Theoretic Sensor Management*. PhD thesis, Massachusetts Institute of Technology.
- Williamson, S., Gerding, E., and Jennings, N. (2008). A principled information valuation for communications during multi-agent coordination. In *Proc. of AAMAS, Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, Estoril, Portugal.
- Williamson, S. A., Gerding, E. H., and Jennings, N. R. (2009). Reward shaping for valuing communications during multi-agent coordination. In *Proc. of AAMAS*, Budapest, Hungary.
- Wu, S.-L., Lin, C.-Y., Tseng, Y.-C., and Sheu, J.-P. (2000). A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Proc. of I-SPAN*, Dallas/Richardson, USA.
- Xi, Y. and Yeh, E. (2010). Distributed algorithms for minimum cost multicast with network coding. *IEEE/ACM Trans. Networking*, 18(2):379 –392.
- Xing, K., Cheng, X., Ma, L., and Liang, Q. (2007). Superimposed code based channel assignment in multi-radio multi-channel wireless mesh networks. In *Proc. of ACM MobiCom*, Montreal, Canada.
- Xu, Z., Fitch, R., and Sukkarieh, S. (2013a). Decentralised coordination of mobile robots for target tracking with learnt utility models. In *Proc. of IEEE ICRA*, Karlsruhe, Germany.
- Xu, Z., Fitch, R., Underwood, J., and Sukkarieh, S. (2013b). Decentralized coordinated tracking with mixed discrete-continuous decisions. *J. Field Robot.*, 30(5):717–740.
- Yan, Y. and Mostofi, Y. (2013). Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *IEEE Trans. Wireless Commun.*, 12(4):1562–1572.