

DISCRETE PARTITIONING AND COVERAGE CONTROL WITH GOSSIP COMMUNICATION

Joseph W. Durham

Ruggero Carli

Department of Mechanical Engineering
University of California
Santa Barbara, California, 93106
(joey, carlirug)@engineering.ucsb.edu

Paolo Frasca

Istituto per le Applicazioni del Calcolo
Consiglio Nazionale delle Ricerche
Rome, Italy
paolo.frasca@gmail.com

Francesco Bullo

Department of Mechanical Engineering
University of California
Santa Barbara, California, 93106
bullo@engineering.ucsb.edu

ABSTRACT

In this paper we propose distributed algorithms to automatically deploy a group of robotic agents and provide coverage of a discretized environment represented by a graph. We present a discrete coverage algorithm which converges to a centroidal Voronoi partition while requiring only pairwise “gossip” communication between the agents. Our theoretical analysis is based on a dynamical system on partitions of the graph’s vertices. We also establish bounds on the computational requirements of the algorithm and demonstrate its functionality with simulations.

1 INTRODUCTION

This paper deals with distributed partitioning and coverage control problems for a network of robotic agents in a potentially non-convex environment. The distributed *partitioning problem* for robotic networks consists of designing control and communication laws to divide an environment into territories. Typically, partitioning is done so as to optimize a cost function that measures the quality of the partitions. *Coverage control* algorithms are usually designed in a similar way, with an additional criterion of optimizing the placement of the agents. In this paper we describe a partitioning and coverage control algorithm which optimizes the configuration of a group of agents in a discrete environment represented by a graph. Optimality is defined with reference to a cost function which depends on the locations of the agents and geodesic distances in the graph. As with all multi-agent coordination applications, the challenge comes from reducing the communication requirements: the proposed algorithm only requires pairwise “gossip” communication.

A broad discussion of partitioning and coverage control is presented in [1] which builds on the classic work of Lloyd [2]

on algorithms for optimal quantizer design through “centering and partitioning.” The relationship between discrete and continuous coverage control laws based on Euclidean distances is studied in [3]. Coverage control and partitioning of discrete sets are also related to the literature on the facility location or k -center problem [4]. Coverage control algorithms for non-convex environments are discussed in [5–7] while equitable partitioning is studied in [8]. In [9] the authors have showed how a group of robotic agents can optimize the partition of a given environment using pairwise “gossip” communication: only one pair of regions is updated at each step of the algorithm. In that work the environment is assumed to be a convex bounded subset of \mathbb{R}^d . This assumption is not suitable for implementation on physical robots which inherently sense and memorize a quantized environment.

There are three main contributions of this paper. First, we extend the results for gossip coverage from [9] to perform coverage of a connected graph using geodesic distances. In the usage we envision, the connected graph represents a discretization of an environment, allowing physical robots to cover complex non-convex environments with holes. We discuss important properties of this setup, namely that the centroid of a robot’s region always belongs to the region (which is a subset of the vertices), and moreover each robot’s region remains connected during the evolution of the algorithm. Second, we prove that, under suitable assumptions on the sequence of updating pairs, the discrete gossip coverage algorithm converges in finite time to a single centroidal Voronoi partition. This result, stronger than that in [9], is a consequence of the discrete nature of the problem. Third, we discuss the implementation of the core computations of the algorithm and provide bounds on their computational complexity. We show that the computations of the algorithm scale well with the size of the environment and the number of robots used,

meaning the algorithm is truly distributed. Combined with the extension to a discrete environment, this efficient computation enables the algorithm to be implemented on a large group of real robots.

This paper is organized as follows. In Section 2 we formally describe the discrete coverage control problem and provide definitions of discrete Voronoi partitions. We present the discrete gossip coverage algorithm in Section 3 and discuss some of its properties. Section 4 contains the main convergence results. In Section 5 we discuss bounds on the computational complexity of the algorithm. Simulation results are shown in Section 6 and some conclusions are given in Section 7.

In our notation, $\mathbb{R}_{\geq 0}$ denotes the set of non-negative real numbers and $\mathbb{Z}_{\geq 0}$ the set of non-negative integers. Given a set X , $|X|$ denotes the number of elements in X . Given sets X, Y , their difference is $X \setminus Y = \{x \in X \mid x \notin Y\}$. A set-valued map, denoted by $T : X \rightrightarrows Y$, associates to an element of X a subset of Y .

2 PROBLEM FORMULATION

We are given a group of robotic agents with limited sensing and communication capabilities, and a discretized environment. We want to apportion the environment into smaller regions, each assigned to one of the agents. Our approach is to iteratively update the partition in such a way as to minimize a cost functional which depends on the current partition and the positions of the agents.

Let the finite set Q be the discretized environment. We assume that the elements of Q , which can be thought as locations, are connected by weighted edges. In other words, we let $G = (Q, E, w)$ be an (undirected) weighted graph with edge set $E \subset Q \times Q$ and weight map $w : E \rightarrow \mathbb{R}_{>0}$; we let $w_e > 0$ be the weight of edge e . We assume that G is connected and think of the edge weights as distances between locations.

In any weighted graph G there is a standard notion of distance between nodes defined as follows. A *path* in G is an ordered sequence of nodes such that any pair of consecutive nodes in the sequence is an edge of G . The *weight of a path* is the sum of the weights of all the edges in the path. Given two vertices h and k in G , the *distance* between h and k , denoted by $d_G(h, k)$, is the smallest weight of any path from h to k , or $+\infty$ if there is no path from h to k . In other words, the distance between two nodes is the weight of the shortest path between them. By convention, $d_G(h, k) = 0$ if $h = k$. Note that $d_G(h, k) = d_G(k, h)$, for any $h, k \in Q$. If the graph is connected, then the distance between any two nodes is finite.

Analogously, we define local distances on induced subgraphs of $G = (Q, E, w)$. Given $I \subset Q$, the *subgraph induced by the restriction of G to I* , denoted by $G \cap I$, is the graph with the set of nodes equal to I and with the set of weighted edges containing all weighted edges of G where both vertices belong to I . In other words, we set $(Q, E, w) \cap I = (Q \cap I, E \cap (I \times I), w|_{I \times I})$. The induced subgraph is a weighted graph that is equipped with a notion of distance between nodes. Given $h, k \in I$, we write $d_I(h, k) := d_{G \cap I}(h, k)$. Note that $d_I(h, k) \geq d_G(h, k)$.

Remark 1 (Discretization of an environment) In the motivational scenario we are considering, a continuous environment is substituted by an *occupancy grid map*, where each grid cell is either free space or an obstacle (occupied) [10]. We define two free cells as adjacent if they border each other in the grid map. The group of robots is then tasked with partitioning the graph of the free cells. However, there are many other methods to discretize a space beyond the use of a grid, including triangularization and other approaches studied in computational geometry [11]. Any such method which produces a connected graph with positive edge weights can be used.

In this paper we will deal with partitioning Q into connected subsets. Given the graph $G = (Q, E, w)$, we define *connected subset of Q* as a subset $S \subset Q$ such that $S \neq \emptyset$ and $G \cap S$ is connected. Moreover let $C(Q)$ denote the set of such subsets. We can then define partitions of Q into connected sets as follows.

Definition 1 (Connected partitions) *Given the graph $G = (Q, E, w)$, we define a connected N -partition of Q as a collection $p = \{p_i\}_{i=1}^N$ of N subsets of Q such that*

1. $\bigcup_{i=1}^N p_i = Q$;
2. $p_i \cap p_j = \emptyset$ if $i \neq j$;
3. $p_i \neq \emptyset$ for all $i \in \{1, \dots, N\}$;
4. $p_i \in C(Q)$ for all $i \in \{1, \dots, N\}$.

Let \mathcal{P} to be the set of such partitions.

On Q , we define a *weight function* to be a bounded positive function $\phi : Q \rightarrow \mathbb{R}_{>0}$ which assigns a positive weight to each element of Q . Given $R \in C(Q)$, we define the *one-center function* $\mathcal{H}_1 : R \rightarrow \mathbb{R}_{\geq 0}$ as

$$\mathcal{H}_1(h; R) = \sum_{k \in R} d_R(h, k) \phi(k).$$

A technical assumption is then needed to define the *generalized centroid* of a connected subset. In what follows, we assume that a *total order* relation, $<$, is defined on Q ; hence, we can also denote $Q = \{1, \dots, |Q|\}$.

Definition 2 (Centroid) *Let Q be a totally ordered set, and $R \in C(Q)$. We define the set of generalized centroids*

$$C(R) := \operatorname{argmin}_{h \in R} \mathcal{H}_1(h; R),$$

and the map $\operatorname{Cd} : C(G) \rightarrow Q$ such that $\operatorname{Cd}(R) := \min\{c \in C(R)\}$. We call $\operatorname{Cd}(R)$ the generalized centroid of R .

In subsequent use we will drop the word ‘‘generalized’’ for brevity. Note that with this definition the centroid is well defined, thanks to the ordering assumption, and the centroid of a region belongs to that region. With a slight notational abuse, we

define $\text{Cd} : \mathcal{P} \rightarrow \mathcal{Q}^N$ as the map which associates to a partition the vector of the centroids of its elements.

With these notions we can introduce a performance function; let the *multicenter function* $\mathcal{H}_{\text{multi-center}} : \mathcal{P} \times \mathcal{Q}^N \rightarrow \mathbb{R}$ be defined by

$$\mathcal{H}_{\text{multi-center}}(p, c) = \sum_{i=1}^N \sum_{x \in p_i} d_{p_i}(x, c_i) \phi(x).$$

We aim to minimize this function with respect to both the partitions p and the points c .

Among all the possible ways of partitioning \mathcal{Q} , there is one which is worth of special attention. Given $c \in \mathcal{Q}^N$ such that if $i \neq j$, then $c_i \neq c_j$, the partition $p \in \mathcal{P}$ is said to be a *Voronoi partition of \mathcal{Q} generated by c* if, for each p_i and all $k \in p_i$, we have $c_i \in p_i$ and $d_G(k, \text{Cd}(p_i)) \leq d_G(k, \text{Cd}(p_j))$, $\forall j \neq i$ (see Section 6 for visual examples of Voronoi partitions).

Proposition 1 (Properties of multi-center function) *Let $p \in \mathcal{P}$, $c \in \mathcal{Q}^N$, and let p^* be a Voronoi partition generated by c . Then*

$$\begin{aligned} \mathcal{H}_{\text{multi-center}}(p, \text{Cd}(p)) &\leq \mathcal{H}_{\text{multi-center}}(p, c), \\ \mathcal{H}_{\text{multi-center}}(p^*, c) &\leq \mathcal{H}_{\text{multi-center}}(p, c). \end{aligned}$$

This cost function can be interpreted as the expected distance from a cell selected uniformly at random in the environment to the centroid of the partition it is a member of. Our motivation for using this cost function is to serve as a base for a task servicing setup. After partitioning the environment, the team of robots will be asked to service tasks which appear at nodes on the graph. By minimizing $\mathcal{H}_{\text{multi-center}}$, we minimize the expected time-to-service of a task in the light load case where each robot has at most one task to service at a time.

These statements motivate the following definition: a partition $p \in \mathcal{P}$ is a *centroidal Voronoi partition* if p is a Voronoi partition generated by $\text{Cd}(p)$. Based on the multicenter function, we define $\mathcal{H}_{\text{expected}} : \mathcal{P} \rightarrow \mathbb{R}$ by

$$\mathcal{H}_{\text{expected}}(p) = \mathcal{H}_{\text{multi-center}}(p, \text{Cd}(p)) = \sum_{i=1}^N \sum_{x \in p_i} d_{p_i}(x, \text{Cd}(p_i)) \phi(x).$$

Observe that $\mathcal{H}_{\text{expected}}$ has the following property as an immediate consequence of Proposition 1: given $p \in \mathcal{P}$, if p^* is a Voronoi partition generated by $\text{Cd}(p)$ then

$$\mathcal{H}_{\text{expected}}(p^*) \leq \mathcal{H}_{\text{expected}}(p).$$

We are now in a position to state the goal of our algorithm: solving the distributed optimization problem $\min_{p \in \mathcal{P}} \mathcal{H}_{\text{expected}}(p)$, subject to the constraint that the agents use only gossip communication, namely that only a pair of agents can communicate at each iteration.

3 GOSSIP COVERAGE ALGORITHM

Here we present a partition-based coverage algorithm that, at each iteration, requires only a pair of adjacent regions to communicate. The following definition describe the admissible pairwise communications.

Definition 3 (Adjacency graph) *Let $p \in \mathcal{P}$. Using the associated graph $G = (\mathcal{Q}, E, w)$, we can define a graph of adjacency between regions of the partition, as follows. Let $\mathcal{G}(p) = (\{1, \dots, N\}, \mathcal{E}(p))$, then $(i, j) \in \mathcal{E}(p)$ if $p_i \cup p_j \in \mathcal{C}(\mathcal{Q})$.*

The *discretized gossip coverage algorithm* is stated as follows.

At each time $t \in \mathbb{Z}_{\geq 0}$, each agent $i \in \{1, \dots, N\}$ maintains in memory a connected subset $p_i(t)$. The collection $p(0) = \{p_1(0), \dots, p_N(0)\}$ is an arbitrary connected N -partition of \mathcal{Q} . At each $t \in \mathbb{Z}_{\geq 0}$ a communicating pair, say $(i, j) \in \mathcal{E}(p(t))$, is selected by a deterministic or stochastic process to be determined. Assume that $i < j$. Every agent $k \notin \{i, j\}$ sets $p_k(t+1) = p_k(t)$, while agents i and j perform the following tasks:

- 1: agent i transmits to agent j its subset $p_i(t)$ and vice-versa
- 2: both agents compute the centroids $\text{Cd}(p_i(t))$ and $\text{Cd}(p_j(t))$ and the sets

$$\begin{aligned} W_{i \rightarrow j} &= \{x \in p_i : d_{p_i \cup p_j}(x, \text{Cd}(p_j)) < d_{p_i \cup p_j}(x, \text{Cd}(p_i))\} \\ W_{j \rightarrow i} &= \{x \in p_j : d_{p_i \cup p_j}(x, \text{Cd}(p_i)) < d_{p_i \cup p_j}(x, \text{Cd}(p_j))\} \\ W_{i \cong j} &= \{x \in p_i \cup p_j : d_{p_i \cup p_j}(x, \text{Cd}(p_i)) = d_{p_i \cup p_j}(x, \text{Cd}(p_j))\} \end{aligned}$$
- 3: **if** $W_{i \rightarrow j} \cup W_{j \rightarrow i} = \emptyset$ **then**
- 4: $p_i(t+1) := p_i(t)$ and $p_j(t+1) := p_j(t)$
- 5: **else**
- 6: $p_i(t+1) := ((p_i \setminus W_{i \rightarrow j}) \cup W_{j \rightarrow i}) \cup W_{j \cong i}$,
 $p_j(t+1) := ((p_j \setminus W_{j \rightarrow i}) \cup W_{i \rightarrow j}) \setminus (W_{j \cong i} \cap p_j)$
- 7: **end if**

Observe that $W_{i \rightarrow j}$ (resp. $W_{j \rightarrow i}$) contains the cells of p_i (resp. p_j) which are closer to $\text{Cd}(p_j)$ (resp. $\text{Cd}(p_i)$), whereas $W_{i \cong j}$ represents the set of the tied cells. In other words, when two robots exchange territory, then their updated regions $\{p_i(t+1), p_j(t+1)\}$ are the Voronoi partition of the set $p_i(t) \cup p_j(t)$ generated by the centroids $\text{Cd}(p_i)$ and $\text{Cd}(p_j)$, with all tied cells assigned to the agent with the lower index.

Now, for any pair $(i, j) \in \{1, \dots, N\}^2$, $i \neq j$, we define the map $T_{ij} : \mathcal{P} \rightarrow \mathcal{P}$ by

$$T_{ij}(p) = \begin{cases} p, & \text{if } (i, j) \notin \mathcal{E}(p) \text{ and } W_{i \rightarrow j} \cup W_{j \rightarrow i} = \emptyset \\ (p_1, \dots, \hat{p}_i, \dots, \hat{p}_j, \dots, p_N), & \text{otherwise,} \end{cases}$$

where $\hat{p}_i = ((p_i \setminus W_{i \rightarrow j}) \cup W_{j \rightarrow i}) \cup W_{j \cong i}$, and $\hat{p}_j = ((p_j \setminus W_{j \rightarrow i}) \cup W_{i \rightarrow j}) \setminus (W_{j \cong i} \cap p_j)$.

The dynamical system on the space of partitions is therefore described by, for $t \in \mathbb{Z}_{\geq 0}$,

$$p(t+1) = T_{ij}(p(t)), \quad \text{for some } (i, j) \in \mathcal{E}(p(t)),$$

together with a rule describing what edge (i, j) is selected at each time. We also define the set-valued map $T : \mathcal{P} \rightrightarrows \mathcal{P}$ by

$$T(p) = \{T_{ij}(p) \mid (i, j) \in \mathcal{E}(p)\}. \quad (1)$$

We claim that (1) is well-defined in the sense that $T_{ij}(p) \in \mathcal{P}$ for all $(i, j) \in \{1, \dots, N\}^2$, if $p \in \mathcal{P}$. We will prove this fact in Proposition 2; first we need the following technical lemma.

Lemma 1 *Let $p^+ = T_{ij}(p)$. If $p_i^+ \neq p_i$, then for all $k \in p_i^+$, every shortest path in $p_i \cup p_j$ from $\text{Cd}(p_i)$ to k is a subset of p_i^+ .*

Proof. Let $u = p_i \cup p_j$. We prove this lemma by contradiction by assuming that, given a shortest path from $\text{Cd}(p_i)$ to k in u , there exists $m \in Q$ which belongs both to the shortest path and to p_j^+ . Consider the case where $i < j$. For m to be in p_j^+ means that $d_u(m, \text{Cd}(p_i)) > d_u(m, \text{Cd}(p_j))$. This implies that

$$\begin{aligned} d_u(k, \text{Cd}(p_i)) &= d_u(m, \text{Cd}(p_i)) + d_u(k, m) \\ &> d_u(m, \text{Cd}(p_j)) + d_u(k, m) \\ &\geq d_u(k, \text{Cd}(p_j)) \end{aligned}$$

This is a contradiction, for $k \in p_i^+$. Similar considerations hold for the case where $i > j$. \square

Proposition 2 *Let T be the pairwise algorithm map defined by (1). If $p \in \mathcal{P}$, then $T(p) \subset \mathcal{P}$.*

Proof. Without loss of generality we assume that $p^+ = T_{ij}(p)$. To prove the statement of the Proposition we need to show that p^+ satisfies points 1 – 4 from Definition 1. Observe, from the definition of the map T_{ij} that $p_i^+ \cup p_j^+ = p_i \cup p_j$ and $p_i^+ \cap p_j^+ = \emptyset$. Moreover, since $\text{Cd}(p_i) \in p_i^+$ and $\text{Cd}(p_j) \in p_j^+$, it follows also that $p_i^+ \neq \emptyset$ and $p_j^+ \neq \emptyset$. These observations imply the validity of the points 1, 2, and 3 for p^+ . Finally, Lemma 1 guarantees that p_i^+ and p_j^+ are connected, i.e. p^+ satisfies also point 4. \square

A couple of remarks are now in order.

Remark 2 (Connected regions) Observe that the previous Proposition guarantees the connectivity of the regions in the partition during the evolution of the algorithm. The key piece which keeps the regions connected is the fact that distances are taken in $p_i \cup p_j$: if distances were taken in G then connectivity would no longer hold.

Remark 3 (Tie breaking rules) It is worth noting that there are many potential variations in how to handle the tied cells in $W_{i \cong j}$. One option would be to give $W_{i \cong j}$ to the agent with the lower \mathcal{H}_i to achieve a form of load balancing. In fact, any method which gives all of $W_{i \cong j}$ to either i or j can be used. However, for a general graph with non-uniform positive edge weights w_e , exact ties between sums of different paths are highly unlikely. The

tie breaking policy only plays a significant role in small graphs with regular edge weights, such as those derived from coarsely gridding the environment.

4 CONVERGENCE

The strength of our results depends on the possibility of enforcing a partition to converge to a centroidal Voronoi partition by means of pairwise communication and territory exchange. We start our convergence analysis with the following concept.

Definition 4 *Consider the connected graph $G = (Q, E, w)$. Let $p \in \mathcal{P}$ and let $\mathcal{G}(p)$ be the associated adjacency graph. Then $p \in \mathcal{P}$ is said to be centroidal Voronoi in pairs if, for any $i \in \{1, \dots, N\}$ we have that*

$$d_{p_i \cup p_j}(k, \text{Cd}(p_i)) \leq d_{p_i \cup p_j}(k, \text{Cd}(p_j))$$

for all $k \in p_i$ and for all j such that $(i, j) \in \mathcal{E}(p)$.

By definition, partitions which are centroidal Voronoi in pairs are fixed points for the map T_{ij} , for all $(i, j) \in \mathcal{E}$. The following Lemma states the equivalence between the set of Centroidal Voronoi in pairs partitions and the set of Centroidal Voronoi partitions.

Lemma 2 *A partition $p \in \mathcal{P}$ is centroidal Voronoi in pairs if and only if it is centroidal Voronoi.*

Proof. We start by proving the “if” part. Assume that p is a centroidal Voronoi partition. Let $x \in p_i$. Observe that at least one of the shortest path connecting x to $\text{Cd}(p_i)$ is all within p_i , otherwise it would exist a region p_j such that $d_G(x, \text{Cd}(p_j)) < d_G(x, \text{Cd}(p_i))$. Hence we can write that $d_G(x, \text{Cd}(p_i)) = d_{p_i}(x, \text{Cd}(p_i)) = d_{p_i \cup p_j}(x, \text{Cd}(p_i))$ for all p_j such that $(i, j) \in \mathcal{E}(p)$. Moreover observe that $d_{p_i \cup p_j}(x, \text{Cd}(p_j)) \geq d_G(x, \text{Cd}(p_j))$. Hence $d_{p_i \cup p_j}(x, \text{Cd}(p_i)) \leq d_{p_i \cup p_j}(x, \text{Cd}(p_j))$.

We prove now the “only if” part by contradiction. To this aim we assume that there exists $p = \{p_1, \dots, p_N\} \in \mathcal{P}$ which is centroidal Voronoi in pairs but not centroidal Voronoi. In other words, there exist two components p_i and p_j of p and an element x of one of such components, say $x \in p_i$, such that $d_G(x, \text{Cd}(p_i)) > d_G(x, \text{Cd}(p_j))$.

Given two nodes a, b of a graph H , let $s_{a,b}^H$ be a shortest path in H connecting a to b . Consider $s_{x, \text{Cd}(p_j)}^G$, and observe that it can be rewritten, for a suitable $m \in \mathbb{N}$ and for a suitable $2m$ -uple of nodes y_1, \dots, y_m , as

$$s_{\text{Cd}(p_j), x}^G = s_{\text{Cd}(p_j), y_1}^{p_j} \cup \left(\bigcup_{h=1}^{m-1} s_{y_{2h}, y_{2h+1}}^{p_{i_h}} \right) \cup s_{y_{2m}, x}^{p_i}$$

where the union of two paths means their concatenation. Let now $s_{\text{Cd}(p_i), y_3}^{p_{i_1}}$ be a shortest path connecting $\text{Cd}(p_{i_1})$ to y_3 . Since

$(j, i_1) \in \mathcal{E}(p)$ and since p is Centroidal Voronoi in pairs, we have that

$$d_{p_{i_1}}(\text{Cd}(p_{i_1}), y_3) \leq d_{p_j}(\text{Cd}(p_j), y_1) + d_{p_j \cup p_{i_1}}(y_1, y_2) + d_{p_{i_1}}(y_2, y_3).$$

Similarly we can prove that $d_{p_{i_2}}(\text{Cd}(p_{i_2}), y_5) \leq d_{p_{i_1}}(\text{Cd}(p_{i_1}), y_3) + d_{p_{i_1} \cup p_{i_2}}(y_3, y_4) + d_{p_{i_2}}(y_4, y_5)$ and iterating this reasoning that $d_{p_{i_{m-1}}}(\text{Cd}(p_{i_{m-1}}), y_{2m-1}) \leq d_{p_{i_{m-2}}}(\text{Cd}(p_{i_{m-2}}), y_{2m-3}) + d_{p_{i_{m-2} \cup p_{i_{m-1}}}}(y_{2m-3}, y_{2m-2}) + d_{p_{i_{m-1}}}(y_{2m-2}, y_{2m-1})$. Plugging together the above inequalities it turns out that

$$\begin{aligned} & d_{p_{i_{m-1}}}(\text{Cd}(p_{i_{m-1}}), y_{2m-1}) + d_{p_{i_{m-1} \cup p_i}}(y_{2m-1}, y_{2m}) + d_{p_i}(y_{2m}, x) \\ & \leq d_{p_j}(\text{Cd}(p_j), y_1) + d_{p_j \cup p_i}(y_1, y_2) + \sum_{h=1}^{m-1} d_{p_{i_h}}(y_{2h+1}, y_{2h}) \\ & \quad + \sum_{h=1}^{m-2} d_{p_{i_h \cup p_{i_{h+1}}}}(y_{2h+1}, y_{2h+2}) + d_{p_{i_{m-1} \cup p_i}}(y_{2m-1}, y_{2m}) \\ & \quad + d_{p_i}(y_{2m}, x) \\ & < d_{p_i}(\text{Cd}(p_i), x) \end{aligned}$$

This contradicts the fact that, since $(i_{m-1}, i) \in \mathcal{E}(p)$ and p is Centroidal Voronoi in pairs,

$$d_{p_{i_{m-1}}}(\text{Cd}(p_{i_{m-1}}), y_{2m}) + d_{p_i}(y_{2m}, x) \geq d_{p_i}(\text{Cd}(p_i), x). \quad \square$$

The following property guarantees that the objective function decreases along the trajectories of T .

Proposition 3 *Let $T : \mathcal{P} \rightrightarrows \mathcal{P}$ be the point-to-set map defined for the pairwise algorithm. If $p \in \mathcal{P}$ and $p^+ \in T(p) \setminus p$, then $\mathcal{H}_{\text{expected}}(p^+) < \mathcal{H}_{\text{expected}}(p)$.*

Proof. Without loss of generality we assume that $p^+ = T_{ij}(p)$. To prove the statement of the proposition we show that

$$\begin{aligned} & \sum_{x \in p_i} d_{p_i}(x, \text{Cd}(p_i)) + \sum_{y \in p_j} d_{p_j}(y, \text{Cd}(p_j)) \\ & > \sum_{x \in p_i^+} d_{p_i^+}(x, \text{Cd}(p_i^+)) + \sum_{y \in p_j^+} d_{p_j^+}(y, \text{Cd}(p_j^+)) \end{aligned}$$

First, since $W_{i \rightarrow j} \cup W_{j \rightarrow i} \neq \emptyset$, we have that

$$\begin{aligned} & \sum_{x \in W_{i \rightarrow j}} d_{p_i \cup p_j}(x, \text{Cd}(p_i)) + \sum_{x \in W_{j \rightarrow i}} d_{p_i \cup p_j}(x, \text{Cd}(p_j)) \\ & > \sum_{x \in W_{i \rightarrow j}} d_{p_i \cup p_j}(x, \text{Cd}(p_j)) + \sum_{x \in W_{j \rightarrow i}} d_{p_i \cup p_j}(x, \text{Cd}(p_i)). \end{aligned}$$

Thus, we further have that

$$\begin{aligned} & \sum_{x \in p_i} d_{p_i}(x, \text{Cd}(p_i)) + \sum_{y \in p_j} d_{p_j}(y, \text{Cd}(p_j)) \\ & \geq \sum_{x \in p_i} d_{p_i \cup p_j}(x, \text{Cd}(p_i)) + \sum_{y \in p_j} d_{p_i \cup p_j}(y, \text{Cd}(p_j)) \\ & > \sum_{x \in (p_i \setminus W_{i \rightarrow j}) \cup W_{j \rightarrow i}} d_{p_i \cup p_j}(x, \text{Cd}(p_i)) + \sum_{x \in W_{i \rightarrow j}} d_{p_i \cup p_j}(x, \text{Cd}(p_j)) \\ & \quad + \sum_{x \in p_j \setminus (W_{j \rightarrow i} \cup W_{j \rightarrow i})} d_{p_i \cup p_j}(y, \text{Cd}(p_j)) + \sum_{x \in W_{j \rightarrow i}} d_{p_i \cup p_j}(x, \text{Cd}(p_i)). \end{aligned}$$

Using the definition of p_i^+ , we can write that

$$\begin{aligned} & \sum_{x \in (p_i \setminus W_{i \rightarrow j}) \cup W_{j \rightarrow i}} d_{p_i \cup p_j}(x, \text{Cd}(p_i)) + \sum_{x \in W_{i \rightarrow j}} d_{p_i \cup p_j}(x, \text{Cd}(p_j)) \\ & = \sum_{x \in p_i^+} d_{p_i \cup p_j}(x, \text{Cd}(p_i)). \end{aligned}$$

Moreover, by Lemma 1 we have that $\sum_{x \in p_i^+} d_{p_i \cup p_j}(x, \text{Cd}(p_i)) = \sum_{x \in p_i^+} d_{p_i^+}(x, \text{Cd}(p_i))$. Similarly we can see that

$$\begin{aligned} & \sum_{x \in p_j \setminus (W_{j \rightarrow i} \cup W_{j \rightarrow i})} d_{p_i \cup p_j}(y, \text{Cd}(p_j)) + \sum_{x \in W_{i \rightarrow j}} d_{p_i \cup p_j}(x, \text{Cd}(p_j)) \\ & = \sum_{x \in p_j^+} d_{p_j^+}(y, \text{Cd}(p_j)). \end{aligned}$$

To conclude the proof observe that from the definition of centroid we have that $\sum_{x \in p_i^+} d_{p_i^+}(x, \text{Cd}(p_i)) \geq \sum_{x \in p_i^+} d_{p_i^+}(x, \text{Cd}(p_i^+))$, and that $\sum_{x \in p_j^+} d_{p_j^+}(x, \text{Cd}(p_j)) \geq \sum_{x \in p_j^+} d_{p_j^+}(x, \text{Cd}(p_j^+))$. \square

We can now state the main convergence results.

Theorem 1 (Convergence under uniformly persistent gossip)

Consider the discrete gossip coverage algorithm T defined in (1) and let $p : \mathbb{Z}_{\geq 0} \rightarrow \mathcal{P}$ be an evolution of T . Assume that for each pair $(i, j) \in \{1, \dots, N\}^2$, $i \neq j$, there exists an increasing sequence of times $\{t_k\}_{k \in \mathbb{Z}_{\geq 0}} \subset \mathbb{Z}_{\geq 0}$ such that $(t_{k+1} - t_k)$ is bounded and $p(t_k + 1) = T_{ij}(p(t_k))$. Then there exist $\hat{t} \in \mathbb{Z}_{\geq 0}$ and a centroidal Voronoi partition \hat{p} such that $p(t) = \hat{p}$ for all $t \geq \hat{t}$.

Proof. Proof is based on verifying the assumptions (i), (ii), (iii), (iv) of Theorem IV.1 in [9]. We start with the following topological consideration.

Let $A, B \subset \mathcal{Q}$, $A \Delta B$ be their symmetric difference, and let $|A|$ be the ‘‘points counting measure’’ of A , that is the number of elements of A . Then if we define $d_{\Delta}(A, B) = |A \Delta B|$, we have that d_{Δ} is a distance on the set of the subsets of \mathcal{Q} , that we denote by $2^{\mathcal{Q}}$. Note that the point counting measure takes integer values. This makes $2^{\mathcal{Q}}$ a discrete topological space. Consider now the product space $(2^{\mathcal{Q}})^N$ and let $q = \{q_1, \dots, q_N\}$ and $\bar{q} = \{\bar{q}_1, \dots, \bar{q}_N\}$ be two

its elements. Then, thanks to results valid for product topological spaces we have that the function $d_{\Delta}^{(N)}(q, \bar{q}) = \sum_{i=1}^N d(q_i, \bar{q}_i)$ is a distance on $(2^Q)^N$. Hence also $(2^Q)^N$ is a discrete topological space. Observe that \mathcal{P} is a subset of $(2^Q)^N$. Since 2^Q , $(2^Q)^N$ and \mathcal{P} are finite, they are also trivially compact. Moreover, since the algorithm $T : \mathcal{P} \rightarrow \mathcal{P}$ is well-defined we have that \mathcal{P} is strongly positively invariant.¹

Now, the fact that all maps from a discrete space are continuous, implies the continuity of the maps T_{ij} and $\mathcal{H}_{\text{expected}}$. Furthermore, thanks to Proposition 3, $\mathcal{H}_{\text{expected}}$ is guaranteed to decrease along the trajectories of T .

The above remarks guarantee the validity of assumptions (i), (ii) and (iii). Assumption (iv) holds true for the hypothesis of uniformly persistent gossip made in the statement of Theorem 1.

Hence we are in the position to apply Theorem IV.1, and conclude convergence to the intersection of the equilibria of the maps T_{ij} , which, according to the definition of the maps T_{ij} coincides with the set of the centroidal Voronoi partitions in pairs. Thanks to Lemma 2, the set of the centroidal Voronoi partitions in pairs is equivalent to set of the centroidal Voronoi partitions.

Hence, since that set is finite, we can argue that the system converges in finite time to one centroidal Voronoi partition. \square

We can also state a probabilistic version of the above result. Its proof is based on applying Theorem IV.2 in [9].

Theorem 2 (Convergence under persistent random gossip)

Consider the discrete gossip coverage algorithm T defined in (1). Given a stochastic process $J : \mathbb{Z}_{\geq 0} \rightarrow \{(i, j) \in \{1, \dots, N\}^2 \mid i \neq j\}$, consider an evolution $p : \mathbb{Z}_{\geq 0} \rightarrow \mathcal{P}$ of T satisfying, for $t \in \mathbb{Z}_{\geq 0}$, $p(t+1) = T_{J(t)}(p(t))$. Assume that there exists $p \in]0, 1[$ and $k \in \mathbb{N}$ such that, for all $(i, j) \in \{1, \dots, N\}^2$, $i \neq j$, and for all $t \in \mathbb{Z}_{\geq 0}$, there exists $h \in \{1, \dots, k\}$ such that

$$\mathbb{P}[J(t+h) = (i, j) \mid J(t), \dots, J(1)] \geq p.$$

Then almost surely there exist $\hat{t} \in \mathbb{Z}_{\geq 0}$ and a centroidal Voronoi partition \hat{p} such that $p(t) = \hat{p}$ for all $t \geq \hat{t}$.

Proof. The proof follows the line of the previous Theorem. In this case one can see that the assumptions of Theorem IV.2 in [9] are met; the application of this theorem assures the almost surely convergence to the set of the Centroidal Voronoi partitions in pairs, and hence, due to the equivalence between the set of the Centroidal Voronoi partitions in pair and the set of the Centroidal Voronoi partitions and the finiteness of this set, the almost surely convergence in finite time to a centroidal Voronoi partition. \square

5 COMPUTATIONAL COMPLEXITY

In this section, we describe the computational requirements of the algorithm assuming that the environment is discretized using an occupancy grid as per Remark 1. Our primary interest is in

¹ Given a set valued map $T : X \rightarrow X$, a subset Z of X is said to be *strongly positively invariant* for T , if for any $x_0 \in Z$, all $x \in T(x_0)$ satisfy $x \in Z$.

graphs derived from occupancy grids because of their ubiquity in robotics, but the fact that all edge weights are the same for these graphs also results in significant computational savings.

5.1 One-to-all Distances

Computing distances from one vertex to all other vertices in a subgraph of G (i.e., one-to-all distances) is the core computation of the algorithm. For the lattice-like graphs from occupancy grids, each cell can have a maximum of 4 edges, so all computational bounds can be stated in terms of $|Q|$. In addition, the uniform weight of edges means that computing distances in the graph is equivalent to counting hops. We can therefore use a Breadth First Search (BFS) approach to compute one-to-all distances on the fly in $O(|Q|)$ in both time and memory. All of the subsequent complexity bounds will assume the use of BFS on a lattice-like graph for this core computation.

However, if the edge weights in the graph are not all the same, then Dijkstra's algorithm must be used. Dijkstra requires $O(|Q| \log(|Q|))$ in time and $O(|Q|)$ in memory, and all subsequent bounds must be adjusted accordingly. For small graphs it is also possible to pre-compute all pairwise distances between vertices using Johnson's algorithm in $O(|Q|^2 \log(|Q|))$ [12] and then use a constant-time lookup on line. However, the memory requirement for storing the pairwise distances scales as $O(|Q|^2)$ and so is intractable for large environments.

5.2 Exchanging Territory

There are three stages to the pairwise computation of which territory cells to exchange between robots i and j . First, i and j must exchange their current subsets p_i and p_j , requiring a transmission of $O(|p_i| + |p_j|)$ bits of information. Second, each agent must compute the centroids c_i and c_j . As we will describe below, this computation is the most computationally demanding part of the algorithm. Finally, for i to update its subset it must compute which cells in $p_i \cup p_j$ are closer to c_i , and j must perform a similar update. These computations require $O(|p_i| + |p_j|)$ in both time and memory.

5.3 Centroid Computation

From Definition 2, centroid c_i is the vertex in p_i which minimizes cost function \mathcal{H}_i , the sum of the distances to all other vertices in p_i . Thus, when a robot is determining c_i , it is solving a 1-median or 1-center problem on the induced subgraph $G \cap p_i$. The potential for local minima in \mathcal{H}_i is what makes this the most demanding part of exchanging territory. We present three approaches for determining c_i which trade-off fidelity of solution for reduced computational complexity.

1. **Exhaustive Search.** To find a vertex in p_i with the minimum cost requires computing the one-to-all distances for each vertex. This approach finds the true centroid but requires $O(|p_i|^2)$ time.
2. **Linear-time Heuristic.** In [9], the centroid of a polygonal partition is the center of mass of the polygon. If the vertices

of G can be represented as points in a Euclidean space, then averaging their positions can produce an analogous estimate of the centroid of p_i in $O(|p_i|)$. The vertex of p_i closest to the computed average would be the new centroid. While this approach does not guarantee that the cost function always decreases on each iteration, it is a heuristic which works well and fast in practice.

3. **Gradient Descent.** Starting from the prior centroid, a robot can use a discrete version of gradient descent to find a local minimum of \mathcal{H}_1 . This approach will often yield the same cost as an exhaustive search in only $O(|p_i| \log(|p_i|))$, but it may not find the lowest cost for some topologies of p_i . For example, if a robot's territory surrounds an obstacle, then there can be local minima in \mathcal{H}_1 around the obstacle. In practice, however, the final equilibria obtained using gradient descent and exhaustive search are often indistinguishable, as we demonstrate in Section 6. The gossip coverage algorithm modified to use gradient descent to locate centroids can also be proven to converge: its analysis is similar to the one in this paper and will be presented elsewhere.

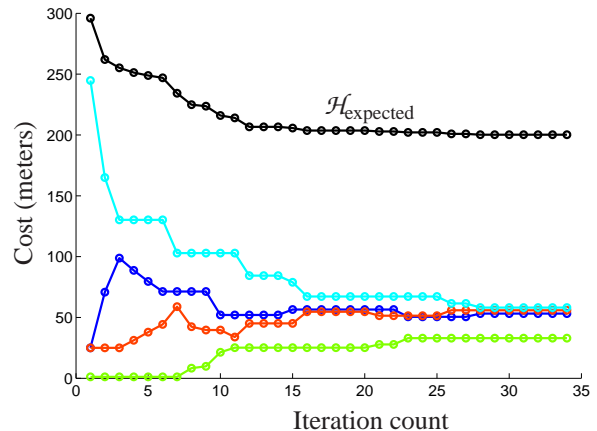


Figure 2. GRAPH OF THE EVOLUTION OF $\mathcal{H}_{\text{expected}}$ FOR THE SIMULATION IN FIG. 1. THE LABELED BLACK LINE SHOWS THE TOTAL COST DECREASES MONOTONICALLY, THE COST FOR EACH ROBOT IS PLOTTED BY COLOR.

a total cost that is only 0.6% lower for this environment.

6 SIMULATION RESULTS

To demonstrate the utility of the proposed gossip coverage algorithm, we implemented it in the open-source Player/Stage robot software system [13], using the Boost Graph Library (BGL) for the graph representing the environment [14]. All results presented here were generated using Player version 2.1.1, Stage 2.1.0, and BGL 1.34.1. Occupancy grid maps were specified as bitmaps with a $0.1m$ resolution, producing a lattice-like graph with all edge weights are equal to $0.1m$. To compute distances on graphs with uniform edge weights we extended the BGL implementation of breadth-first-search with a distance recorder event visitor. To compute distances in subgraphs of the environment we also added an event visitor on vertex discovery to check whether the vertex was a member of the subgraph.

The first simulation shown in Fig. 1 consists of four robots partitioning a square $2m \times 2m$ environment. In the initial setup shown in Fig. 1(a), the robot in the top right controls most of the environment while the robot in the bottom left controls very little. On each iteration, a random pair of robots with adjacent territories will consider exchanging territory. After any change in territory ownership, the pair of robots involved recompute the centroid of their region using exhaustive search and move to their new centroid. The evolution of the cost function over the iterations of the algorithm is shown in Fig. 2. After 33 iterations and 17 pairwise territory exchanges, the robots reach the centroidal Voronoi partition shown in Fig. 1(d).

Fig. 3 shows the second set of simulations, where ten robots partition a non-convex $4m \times 4m$ environment with holes. In the initial partitioning there is a large discrepancy in territory sizes as shown in Fig. 3(a). The next two figures, 3(b) and 3(c), show the final equilibria produced using exhaustive search and gradient descent, respectively, to locate centroids. The final solutions are very similar with the exhaustive search approach producing

7 CONCLUSION

We have described a gossip coverage algorithm on a graph which can guarantee convergence to a centroidal Voronoi partition using only pairwise communication. We presented simulation results to verify these claims. We have also presented three strategies for implementing the algorithm, including a gradient descent heuristic which converges to a stable approximation of a centroidal Voronoi partition in $O(\frac{n}{m} \log(\frac{n}{m}))$, where n is the number of vertices in the map and m is the number of robots. These time complexity bounds allow the algorithm to scale well for large maps with a high number of robots.

In future work, we intend to perform hardware experiments of the algorithm which will require a motion protocol for the robots to guarantee they make contact with their neighbors. In addition, the centroidal Voronoi partition which this algorithm converges to may be only a local minima of the cost function. Extending this approach with some form of annealing would enable the algorithm to better avoid local minima.

8 ACKNOWLEDGEMENTS

The authors would like to thank Dario Cazzaro and Luca Invernizzi for their effort in implementing a version of this algorithm in Player/Stage. This work was supported in part by ARO MURI grant W911NF-05-1-0219, ONR grant N00014-07-1-0721 and NSF grant CNS-0834446.

REFERENCES

- [1] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Applied Mathematics Se-

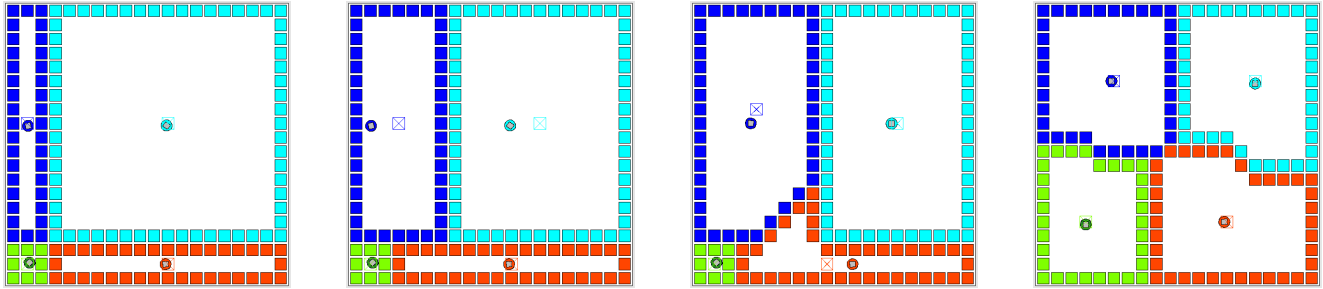


Figure 1. SIMULATION OF GOSSIP COVERAGE ALGORITHM WITH FOUR ROBOTS IN A DISCRETIZED, SQUARE ENVIRONMENT. EACH ROBOT OWNS ALL THE CELLS INSIDE ITS COLORED BOUNDARY, CENTROIDS ARE DRAWN WITH X-BOXES. IN (a), THE ROBOTS ARE AT THE CENTROIDS OF THEIR INITIAL PARTITIONS. (b) AND (c) SHOW THE FIRST AND SECOND TERRITORY SWAP BETWEEN PAIRS OF ROBOTS AND THE RESULTING CHANGE TO THE CENTROIDS OF THE PARTITIONS. THE EQUILIBRIUM CENTROIDAL VORONOI PARTITION IS SHOWN IN (d).

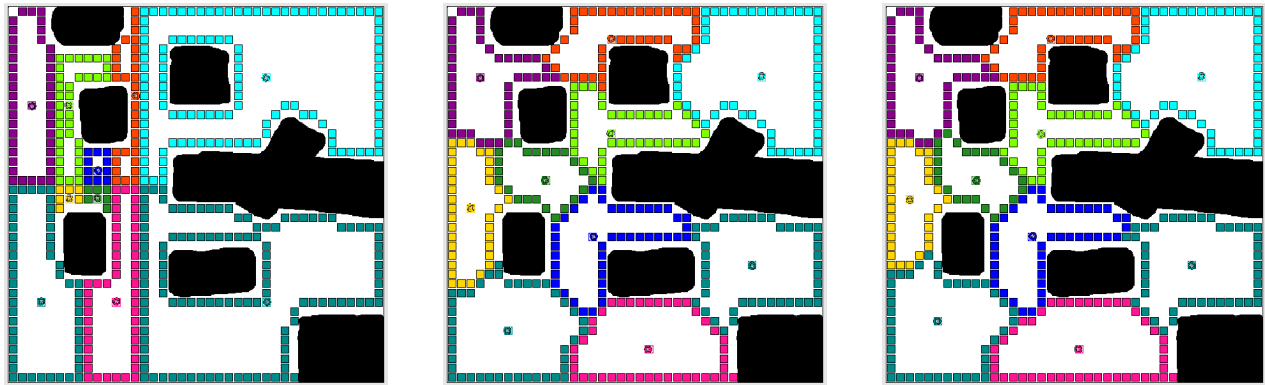


Figure 3. SIMULATION OF GOSSIP COVERAGE ALGORITHM WITH TEN ROBOTS IN A NON-CONVEX ENVIRONMENT. (a) SHOWS THE ROBOTS AT THE CENTROIDS OF THEIR INITIAL PARTITIONS. (b) AND (c) SHOW THE VERY SIMILAR EQUILIBRIA REACHED USING EXHAUSTIVE SEARCH AND GRADIENT DESCENT, RESPECTIVELY, TO LOCATE CENTROIDS.

- ries, Princeton University Press, 2009. Available at <http://www.coordinationbook.info>.
- [2] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982. Presented as Bell Laboratory Technical Memorandum at a 1957 Institute for Mathematical Statistics meeting.
- [3] C. Gao, J. Cortés, and F. Bullo, "Notes on averaging over acyclic digraphs and discrete coverage control," *Automatica*, vol. 44, no. 8, pp. 2120–2127, 2008.
- [4] V. V. Vazirani, *Approximation Algorithms*. Springer, 2001.
- [5] M. Zhong and C. G. Cassandras, "Distributed coverage control in sensor network environments with polygonal obstacles," in *IFAC World Congress*, (Seoul, Korea), pp. 4162–4167, July 2008.
- [6] C. H. Caicedo-Núñez and M. Žefran, "Performing coverage on nonconvex domains," in *IEEE Conf. on Control Applications*, (San Antonio, TX), pp. 1019–1024, Sept. 2008.
- [7] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *IEEE Conf. on Decision and Control*, (Cancún, México), pp. 3947–3952, Dec. 2008.
- [8] O. Baron, O. Berman, D. Krass, and Q. Wang, "The equitable location problem on the plane," *European Journal of Operational Research*, vol. 183, no. 2, pp. 578–590, 2007.
- [9] P. Frasca, R. Carli, and F. Bullo, "Multiagent coverage algorithms with gossip communication: control systems on the space of partitions," in *American Control Conference*, (St. Louis, MO), pp. 2228–2235, June 2009.
- [10] H. Minc, *Nonnegative matrices*. John Wiley, 1988.
- [11] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer, 2 ed., 2000.
- [12] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1–13, 1977.
- [13] B. Gerkey, R. T. Vaughan, A. Howard, and contributors, "The Player/Stage Project." <http://playerstage.sourceforge.net>, June 2008. Version 2.1.1.
- [14] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, "Boost Graph Library." <http://www.boost.org>, July 2007. Version 1.34.1.