Distributed consensus on enclosing shapes and minimum time rendezvous

Giuseppe Notarstefano   Francesco Bullo

***Abstract*— In this paper we introduce the notion of optimization under control and communication constraint in a robotic network. Starting from a general setup, we focus our attention on the problem of achieving rendezvous in minimum time for a network of first order agents with bounded inputs and limited range communication. We propose two dynamic control and communication laws. These laws are based on consensus algorithms for distributed computation of the minimal enclosing ball and orthotope of a set of points. We prove that these control laws converge to the optimal solution of the centralized problem (i.e., when no communication constrains are enforced) as the bound on the control input goes to zero. Moreover, we give a bound for the time complexity of one of the two laws.**

I. Introduction

The interesting aspect of motion coordination consists in combining together problems from control and communication theory. The main difficulty deals with integrating the sensing, computing, communication and control aspects of problems involving groups of mobile agents. A well known problem in control theory is optimal control. Roughly speaking, it consists in finding a feedback law that minimizes some cost functional under some inputs and dynamics constraint. In this paper we introduce the notion of optimal control and communication for a network of robotic agents. We want to study how to solve an optimization problem, in presence of both the usual motion constraints and the communication ones. In particular this paper is a preliminary contribution towards what might be loosely referred to as "distributed geometric optimization." In fact many optimization problems for robotic networks can be shown to be equivalent to the computation of geometric shapes. While in a centralized setting the solution is usually simple, the problem becomes very complicated when it must be solved in a distributed way. Distributed computation over network has been largely studied for fixed topologies; e.g., see [1].

In this paper we point our attention on the well known rendezvous coordination task and look for solutions that solve such task in minimum time. We look for distributed solutions in networks of mobile agents with first order dynamics, bounded inputs and limited-range communication.

The "multi-agent rendezvous" problem and a "circumcenter algorithm" have been introduced by Ando and coworkers in [2]. The algorithm proposed in [2] has been extended to various synchronous and asynchronous stop-and-go strategies in [3]. A related algorithm, in which connectivity constraints are not imposed, is proposed in [4]. In [5] the class of "circumcenter algorithms" has been studied in networks of agents whose state space is $\mathbb{R}^d$, for arbitrary d, and with communication topology characterized by proximity graphs spatially distributed over the disk graph. In [6] the (time and communication) complexity of this and other algorithms has been studied. All these coordination schemes are memoryless (static feedback). In this paper we want to explore dynamic control and communication laws in order to approximate the optimal solution of the minimum time rendezvous. In particular the control and communication laws is based on reaching consensus on some logic variables and at the same time moving toward the current estimation. A similar approach was used in [7] where the agents try to reach a consensus on a set of variables called coordination variables.

Studying the minimum time rendezvous problem in the centralized setting we show that, depending on the norm used to bound the control input, the optimal solution consists of moving toward the center of the minimal enclosing ball (bound on L_2 norm) and toward the center of the minimal enclosing orthotope (bound on the infinity norm) of the points located at the initial position of the agents.

Our main result is the design of a control and communication law based on a consensus algorithm for the distributed computation of the minimal enclosing ball and the minimal enclosing orthotope of a set of points. We prove the correctness of the two consensus algorithms and provide a bound on the time of convergence for the orthotope case. Then we prove that the control and communication law that combines the consensus with the motion law converges to the optimal solution as the control bound goes to zero. Moreover, for the problem with input bounded by the infinity norm (corresponding to the computation of the minimal enclosing orthotope), we prove that the control and communication law is a constant factor approximation of the centralized optimal solution.

In Section II we introduce a formal model of robotic network inspired by the one introduced in [6]. Moreover, we define the optimal control and communication problem. In Section III we characterize the solution of minimum time rendezvous in a centralized setting. In Section IV we define the *FloodMEB* and *FloodMEO* algorithms for the distributed computation of minimal enclosing ball and orthotope, prove their correctness and give bounds on time complexity. Section V contains the control and communication laws based on the consensus algorithms described in Section IV. Finally in

This material is based upon work supported in part by ARO MURI Award W911NF-05-1-0219. The authors would like to thank Sonia Martínez, Jorge Cortés, and Emilio Frazzoli for numerous discussions on robotic networks.

Giuseppe Notarstefano is with the the Department of Information Engineering, Università di Padova, Via Gradenigo 6/b, 35131 Padova, Italy, notarste@dei.unipd.it

Francesco Bullo is with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA 93106, bullo@engineering.ucsb.edu

Section V-C and Section VI we show simulations and draw the conclusions with future perspectives.

II. Preliminary developments

In this section we recall the concepts of network of robotic agents, coordination tasks and complexity measures, and introduce the notion of optimization under motion and communication constraints.

A. Notation

We let $\mathbb{N}$, $\mathbb{N}_0$, and $\mathbb{R}_+$ denote the natural numbers, the non-negative integer numbers, and the positive real numbers, respectively. We let $\prod_{i\in\{1,\dots,n\}} S_i$ denote the Cartesian product of sets $S_1,\dots,S_n$. For $p\in\mathbb{R}$, we let $\lfloor p\rfloor$ and $\lceil p\rceil$ denote the floor and ceil of p. For $r\in\mathbb{R}_+$ and $p\in\mathbb{R}^d$, we let $B(p,r)$ denote the closed ball centered at p with radius r, i.e., $B(p,r)=\{q\in\mathbb{R}^d \mid \|p-q\|_2\le r\}$ and $C(p,r)$ denote the closed hypercube centered at p with sides of length r and parallel to the coordinate axes, i.e., $C(p,r)=\{q\in\mathbb{R}^d \mid \|p-q\|_\infty\le r\}$.

For $f,g:\mathbb{N}\to\mathbb{R}$, we say that $f\in O(g)$ (respectively, $f\in\Omega(g)$) if there exist $n_0\in\mathbb{N}$ and $k\in\mathbb{R}_+$ such that $|f(n)|\le k|g(n)|$ for all $n\ge n_0$ (respectively, $|f(n)|\ge k|g(n)|$ for all $n\ge n_0$). If $f\in O(g)$ and $f\in\Omega(g)$, then we use the notation $f\in\Theta(g)$.

Next, we briefly review some useful proximity graphs. Given $r_{\text{cmm}}\in\mathbb{R}_+$, the *disk graph* $\mathcal{G}_{\text{disk}}(r_{\text{cmm}})$, respectively *cube graph* $\mathcal{G}_{\text{cube}}(r_{\text{cmm}})$, is the state dependent graph on $\mathbb{R}^d$ defined by the following statement: for any pointset $\{p^{[1]},\dots,p^{[n]}\}\subset\mathbb{R}^d$, the pair (i,j) is an edge in $\mathcal{G}_{\text{disk}}(r_{\text{cmm}})\cdot(\{p^{[1]},\dots,p^{[n]}\})$, respectively $\mathcal{G}_{\text{cube}}(r_{\text{cmm}})\cdot(\{p^{[1]},\dots,p^{[n]}\})$, if and only if $i\neq j$ and

$$\|p^{[i]}-p^{[j]}\|_2\le r_{\text{cmm}} \iff p^{[i]}-p^{[j]}\in B(0_d,r_{\text{cmm}}),$$

respectively

$$\|p^{[i]}-p^{[j]}\|_\infty\le r_{\text{cmm}} \iff p^{[i]}-p^{[j]}\in C(0_d,r_{\text{cmm}}).$$

Another useful graph is the complete graph $\mathcal{G}_{\text{cmpl}}$, i.e., the graph with edges between any pair of nodes.

Finally, given a graph $\mathcal{G}$ (even not state dependent), we denote with $\text{dist}_{\mathcal{G}}(i,j)$ the topological distance between i and j, i.e., the minimum number of agents to go from i to j in the graph $\mathcal{G}$. We define $\text{diam}_{\mathcal{G}}$, the *diameter* of $\mathcal{G}$, to be the maximum topological distance, $\text{dist}_{\mathcal{G}}(i,j)$, for all (i,j).

B. Modeling a network of robotic agents

We describe a (uniform) network of robotic agents using the formal model introduced in [6] modified for the discrete time case. The network is modeled as a tuple $(I,\mathcal{A},E_{\text{cmm}})$. $I=\{1,\dots,n\}$ is the *set of unique identifiers (UIDs)*; $\mathcal{A}=\{A^{[i]}\}_{i\in I}=\{(X,U,X_0,f)\}_{i\in I}$ is called the *set of physical agents* and is a set of control systems consisting of a differentiable manifold X (state space), a compact subset U of $\mathbb{R}^m$ (input space), a subset X_0 of X (set of allowable initial states) and a (sufficiently smooth) map $f:X\times U\to X$ describing the dynamics of ith agent; $E_{\text{cmm}}:X^n\to I\times I$ is called the *communication edge map*.

The robotic network evolves according to a discrete-time communication and motion model.

Definition 2.1 (Control and communication law): Let $\mathcal{S}$ be a robotic network. A *(uniform, synchronous, dynamic) control and communication law* $\mathcal{CC}$ for $\mathcal{S}$ consists of the sets:

(i) L, a set containing the `null` element, called the *communication language*; elements of L are called *messages*;
(ii) W, set of values of some *logic variables* $w^{[i]}$, $i\in I$;
(iii) $W_0\subseteq W$, subsets of *allowable initial values*;

and of the maps:

(i) $\text{msg}: X\times W\times I\to L$, *message-generation function*;
(ii) $\text{stf}: W\times L^n\to W$, called *state-transition function*;
(iii) $\text{ctl}: X\times W\times L^n\to U$, called *control function*. □

Roughly speaking this definition has the following meaning: for all $i\in I$, to the ith physical agent corresponds a logic process, labeled i, that performs the following actions. First, at each communication round the ith logic process sends to each of its neighbors in the communication graph a message (possibly the `null` message) computed by applying the message-generation function to the current values of $x^{[i]}$ and $w^{[i]}$. After a negligible period of time, the ith logic process resets the value of its logic variables $w^{[i]}$ by applying the state-transition function to the current value of $w^{[i]}$, and to the messages received at time t. Between communication instants, the motion of the ith agent is determined by applying the control function to the current value of $x^{[i]}$, and the current value of $w^{[i]}$. This idea is formalized as follows.

Definition 2.2 (Evolution of a robotic network): Let $\mathcal{S}$ be a robotic network and $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$. The *evolution* of $(\mathcal{S},\mathcal{CC})$ from initial conditions $x_0^{[i]}\in X_0$ and $w_0^{[i]}\in W_0$, $i\in I$, is the set of curves $x^{[i]}:\mathbb{N}\to X$ and $w^{[i]}:\mathbb{N}\to W$, $i\in I$, satisfying

$$x^{[i]}(t+1)=f\big(x^{[i]}(t),\,\text{ctl}(x^{[i]}(t),w^{[i]}(t+1),y^{[i]}(t))\big),$$

where, for $i\in I$,

$$w^{[i]}(t+1)=\text{stf}(w^{[i]}(t),y^{[i]}(t))\,,$$

with the conventions that $x^{[i]}(t_0)=x_0^{[i]}$ and $w^{[i]}(t_0)=w_0^{[i]}$, $i\in I$. Here, the function $y^{[i]}:\mathbb{N}\to L^n$ (describing the messages received by agent i) has components

$$y_j^{[i]}(t)=\begin{cases}\text{msg}(x^{[j]}(t),w^{[j]}(t),i), & \text{if } (i,j)\in E_{\text{cmm}},\\ \texttt{null}, & \text{otherwise.}\end{cases} \qquad \square$$

In the paper we consider the following network. Each agent i occupies a location $p^{[i]}\in\mathbb{R}^d$, $d\in\mathbb{N}$, and moves according to the first order discrete-time integrator

$$p^{[i]}(t+1)=p^{[i]}(t)+u^{[i]}(t). \tag{1}$$

The communication edge map can be either the one arising according to the *disk graph*, E_{disk}, or the one according to the *cube graph*, E_{cube}. Each control $u^{[i]}$ takes values in a bounded subset of $\mathbb{R}^d$, that can be either $B(0,r_{\text{ctr}})$ or $C(0,r_{\text{ctr}})$, i.e., $\|u^{[i]}\|_2\le r_{\text{ctr}}$ or $\|u^{[i]}\|_\infty\le r_{\text{ctr}}$. Notice that, in general, the type of communication edge map and the type of control bound are not related. Finally the control and communication law will be defined depending on the coordination task.

C. Coordination tasks and time complexity

We are ready to define the notion of task and of task achievement by a robotic network.

Definition 2.3 (Coordination task): Let $\mathcal{S}$ be a robotic network. A *(static) coordination task* for $\mathcal{S}$ is a map $\mathcal{T} : X^n \to \{\text{true, false}\}$. Additionally, let $\mathcal{CC}$ a control and communication law for $\mathcal{S}$. The law $\mathcal{CC}$ *achieves* the task $\mathcal{T}$ if, for all initial conditions $x_0^{[i]} \in X_0$ and $w_0^{[i]} \in W_0$, $i \in I$, the corresponding network evolution $t \mapsto (x(t), w(t))$ has the property that there exists $T \in \mathbb{N}$ such that $\mathcal{T}(x(t)) = \texttt{true}$ for all $t \geq T$. □

We are finally ready to define the notion of time complexity as the minimum number of communication rounds needed by the agents to achieve the task $\mathcal{T}$ with $\mathcal{CC}$.

Definition 2.4 (Time complexity): Let $\mathcal{S}$ be a robotic network and let $\mathcal{T}$ be a coordination task for $\mathcal{S}$. Let $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$ compatible with $\mathcal{T}$. The *time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ from $x_0 \in X_0^n$* is

$$\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0) = \inf\{T \in \mathbb{N} \mid \mathcal{T}(x(t)) = \texttt{true}\,,\ \forall\, t \geq T\}$$

where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0).

The *time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$*, $\mathrm{TC}(\mathcal{T}, \mathcal{CC})$, is the maximum $\mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0)$ over all initial conditions x_0.

D. Optimal control and communication in robotic networks

Having defined a coordination task for a robotic network, we can ask whether such task can be accomplished minimizing some cost functional. In what follows we will introduce the notion of *optimal control and communication problem* and of *optimal control and communication law* as solution of the problem.

Definition 2.5 (Optimal control and communication): Given a task $\mathcal{T}$ and a cost functional $J(u(\cdot), x(T), T)$, an optimal control and communication problem is the following:

$$\text{minimize}_{u(\cdot), x(0), x(T), T} \quad J(u(\cdot), x(T), T)$$

$$J(u(\cdot), x(T), T) = \textstyle\sum_{\tau=0}^{T} (l(x(\tau), u(\tau)) + g(x(T)),$$

subj. to

(i) $(x(\cdot), u(\cdot))$ is an input-state trajectory of $\mathcal{A}$, $\mathcal{A} = \{A^{[i]}\}_{i \in I}$;

(ii) i and j can communicate if and only if $(i, j) \in E_{\text{cmm}}(x^{[1]}(t), \ldots, x^{[n]}(t))$;

(iii) $\mathcal{T}(x(t)) = \texttt{true}$ for all $t \geq T$, $T \in \mathbb{N}$.

where $l : X^n \times U^n \to \mathbb{R}$ is a sufficiently smooth and nonnegative-valued function, called *stage cost*, and $g : X^n \to \mathbb{R}$ has the same properties plus $g(x) = 0$ for all $x \in X^n$ such that $\mathcal{T}(x) = \texttt{true}$ (for an admissible $\mathcal{CC}$). □

We say that a control and communication law $\mathcal{CC}$ is optimal with respect to the coordination task $\mathcal{T}$ and the cost functional J, if it solves the above optimal control and communication problem.

We call $\mathcal{CC}$ a *centralized* optimal control and communication law if it solves the optimization problem for a network of robotic agents that communicate according to the complete graph, i.e., the communication edge map is E_{cmpl}.

Remark 2.6: The centralized solution of an optimal control and communication problem is the classical solution of the optimal control problem for the whole network system without communication constraints. □

III. Centralized minimum time rendezvous

In this section we study the rendezvous problem for a robotic network of first order agents with communication edge map E_{disk} or E_{cube} and look for a control and communication law that solves the problem in minimum time.

More formally, let $\mathcal{S} = (I, \mathcal{A}, E_{\text{cmm}})$ be a uniform robotic network. The *(exact) rendezvous task* $\mathcal{T}_{\text{rndzvs}} : X^n \to \{\text{true, false}\}$ for $\mathcal{S}$ is the static task defined by

$$\mathcal{T}_{\text{rndzvs}}(x) = \begin{cases} \texttt{true}, & \text{if } x^{[i]} = x^{[j]}, \\ & \forall (i,j) \in E_{\text{cmm}}(x), \\ \texttt{false}, & \text{otherwise.} \end{cases}$$

for $x = (x^{[1]}, \ldots, x^{[n]})$.

Thus, given the uniform network $\mathcal{S} = (I, \mathcal{A}, E_{\text{cmm}})$, the *minimum time rendezvous* problem for first order agents with limited-range communication and bounded control input is the following:

$$\text{minimize} \textstyle\sum_{\tau=0}^{T} 1,$$

$$u(\cdot), p(T)$$

subj. to

(i) $(p(\cdot), u(\cdot))$ is an input-state trajectory of $\mathcal{A}$, $\mathcal{A} = \{A^{[i]}\}_{i \in I} = \{(\mathbb{R}^d, U, \mathbb{R}^d, f)\}_{i \in I},\ p(0) = p_0$;

(ii) i and j can communicate if and only if $(i, j) \in E_{\text{cmm}}(p^{[1]}(t), \ldots, p^{[n]}(t))$;

(iii) $\mathcal{T}_{\text{rndzvs}}(p^{[1]}, \ldots, p^{[n]}) = \texttt{true}$ for all $t \geq T$, $T \in \mathbb{N}$.

Here U is either $B(0, r_{\text{ctr}})$ or $C(0, r_{\text{ctr}})$, $f(p^{[i]}(t), u^{[i]}(t)) = p^{[i]}(t) + u^{[i]}(t)$ and the communication edge map E_{cmm} is either E_{disk} or E_{cube}.

We refer to the minimum time rendezvous problem with communication edge map E_{cmm} and input set U as $\mathcal{MTR}(E_{\text{cmm}}, U)$.

Next, we provide some preliminary results for the centralized setting of the above problem, that is, for $\mathcal{MTR}(E_{\text{cmpl}}, U)$. Let $\mathrm{MEB}(p^{[1]} \cdots p^{[n]})$ and $\mathrm{MEO}(p^{[1]} \cdots p^{[n]})$ the minimal enclosing ball and orthotope of points $(p^{[1]} \cdots p^{[n]})$, and let $\mathrm{MBC}(p^{[1]} \cdots p^{[n]})$ and $\mathrm{MOC}(p^{[1]} \cdots p^{[n]})$ the centers of $\mathrm{MEB}(p^{[1]} \cdots p^{[n]})$ and $\mathrm{MEO}(p^{[1]} \cdots p^{[n]})$ respectively. We present the following theorem omitting the proof based on geometric arguments because of space constraints.

Theorem 3.1: For all $r_{\text{ctr}} \in \mathbb{R}_+$, $p_0^{[i]} \in \mathbb{R}^d$, $i \in \{1, \ldots, n\}$ the solution of $\mathcal{MTR}(E_{\text{cmpl}}, U)$, $U = B(0, r_{\text{ctr}})$ or $U = C(0, r_{\text{ctr}})$, is not unique (the problem is not normal). If $u^{[i]} \in B(0, r_{\text{ctr}})$, $i \in \{1, \ldots, n\}$, then

(i) $p(T) = p_{\text{rndzvs, disk}} = \mathrm{MBC}(p^{[1]}(0), \ldots, p^{[n]}(0))$,

$$u^{[i]}(t) = \min\{r_{\text{ctr}}, \|p_{\text{rndzvs}} - p^{[i]}(t)\|_2\} \cdot \mathrm{vers}\,(p_{\text{rndzvs}} - p^{[i]}(t)), \quad i \in \{1, \ldots, n\},$$

is a solution of $\mathcal{MTR}(E_{\text{cmpl}}, B(0, r_{\text{ctr}}))$;

(ii) if $\forall i \in \{1,\dots,n\}$, $\|p^{[i]} - \mathrm{MBC}(p^{[1]}\cdots p^{[n]})\|_2 \leq r_{\text{ctr}}$, then the solution of $\mathcal{MTR}(E_{\text{cmpl}}, B(0, r_{\text{ctr}}))$ is given by $p(T) = p_{\text{rndzvs, disk}}$, $p_{\text{rndzvs, disk}} \in \cap_{i\in\{1,\dots,n\}} B(p^{[i]}, r_{\text{ctr}})$, and $u^{[i]}(t) = p_{\text{rndzvs, disk}} - p^{[i]}(t)$.

Alternatively, if $u^{[i]} \in C(0, r_{\text{ctr}})$, $i \in \{1,\dots,n\}$, then

(iii) $p(T) = p_{\text{rndzvs, cube}}$, $p_{\text{rndzvs, cube}} \in \mathcal{R}_{\text{MT}}$, ${u_a}^{[i]}(t) = \min\{r_{\text{ctr}}, |p_{\text{rndzvs,a}} - {p_a}^{[i]}(t)|\}\,\mathrm{sign}(p_{\text{rndzvs,a}} - {p_a}^{[i]}(t))$, $i \in \{1,\dots,n\}$, $a \in \{1,\dots,d\}$ is a solution of $\mathcal{MTR}(E_{\text{cmpl}}, C(0, r_{\text{ctr}}))$, where

$$\mathcal{R}_{\text{MT}} = \prod_a \Big[\mathrm{MOC}(p^{[1]}(0),\dots,p^{[n]}(0)) - \frac{1}{2}(l_{\max} - l_{\text{a}}), \mathrm{MOC}(p^{[1]}(0),\dots,p^{[n]}(0)) + \frac{1}{2}(l_{\max} - l_{\text{a}})\Big],$$

$l_{\max}$ is the largest side of $\mathrm{MEO}(p^{[1]}(0),\dots,p^{[n]}(0))$ and l_{a} is the side in direction a;

(iv) if $\forall i \in \{1,\dots,n\}$ $\|p^{[i]} - \mathrm{MOC}(p^{[1]}\cdots p^{[n]})\|_\infty \leq r_{\text{ctr}}$ the solution of $\mathcal{MTR}(E_{\text{cmpl}}, C(0, r_{\text{ctr}}))$ is given by $p(T) = p_{\text{rndzvs}}$, $p_{\text{rndzvs}} \in \cap_{i\in\{1,\dots,n\}} C(p^{[i]}, r_{\text{ctr}})$, and $u^{[i]}(t) = p_{\text{rndzvs}} - p^{[i]}(t)$. □

IV. Distributed consensus on minimal enclosing ball and orthotope

In the previous section we have shown that minimal enclosing shapes play a key role in the solution of minimum time rendezvous. In fact if the agents could know the center of such shapes (ball or orthotope) the solution of minimum time rendezvous would be just a control law that drives each agent to this point. Therefore, in this section, we want to explore two consensus algorithms to compute the minimal enclosing ball and the minimal enclosing orthotope of a set of given points in $\mathbb{R}^d$ in a distributed way.

Here is an informal description of what we shall refer to as the *FloodMEB* algorithm:

> *[Informal description]* Each agent initializes the minimal enclosing ball to its initial position, then, at each communication round, performs the following tasks: (i) it acquires from its neighbors (a message represented by) the coordinates of the minimum set of points describing the boundary of their minimal enclosing ball and the coordinates of their initial position; (ii) it computes the minimal enclosing ball of the point set comprised of its and its neighbors' set of points and its initial position (that it maintains in memory); (iii) it updates its logic variables and message as in (i).

Before describing the algorithm more formally, we need to introduce some notation and state some properties of the minimal enclosing ball. Given a set of m points $\{q_1,\dots,q_m\} \subset \mathbb{R}^d$ in generic positions, we denote with $\mathrm{MEB}_{\text{bndry}}(\{q_1,\dots,q_m\})$ the minimum set of points on the boundary of $\mathrm{MEB}(\{q_1,\dots,q_m\})$ that uniquely identify such boundary. When the points are in generic position, we let $\mathrm{MEB}_{\text{bndry}}(\{q_1,\dots,q_m\})$ denote *a* minimum set of points on the boundary of $\mathrm{MEB}(\{q_1,\dots,q_m\})$ that identify such boundary. Moreover, let $\mathrm{MBR} : \mathbb{F}(\mathbb{R}^d) \to \mathbb{R}$ the function that associates to a set of points the radius of the minimal enclosing ball of such points.

Lemma 4.1 (MEB properties): Let Q_n a set of n points. The following statements hold.

(i) there exists a subset $Q_d \subset Q_n$ of $d+1$ elements such that $\mathrm{MEB}(Q_d) = \mathrm{MEB}(Q_n)$;

(ii) for all $Q_{n_1}, Q_{n_2} \subset Q_n$ with $Q_{n_1} \subset Q_{n_2}$, then $\mathrm{MBR}(Q_{n_1}) \leq \mathrm{MBR}(Q_{n_2})$;

(iii) if $\mathrm{MBR}(Q_{n_1}) = \mathrm{MBR}(Q_{n_2})$, then $\mathrm{MBC}(Q_{n_1}) = \mathrm{MBC}(Q_{n_2})$;

(iv) the number of possible values of $\mathrm{MBR}(Q_{n_1})$, for all $Q_{n_1} \subset Q_n$, is finite. □

Remark 4.2: An important implication of Lemma 4.1(i) is that $\mathrm{MEB}_{\text{bndry}}(\{q_1,\dots,q_n\})$ has at most $d+1$ points, then the number of packets in the message sent and stored by each agent is at most $d+1$ and does not depend on n. □

The algorithm is described formally in the following table.

Name:	*FloodMEB* algorithm.
Goal:	Solve the problem of computing minimal enclosing ball of a set of points.
Logic state:	$w^{[i]} = ({P^{[i]}}_{\text{bndry}}, p_0^{[i]})$;
Msg function:	$\mathrm{msg}(x^{[i]}, w^{[i]}, i) = w^{[i]}$;
Initialization:	${P^{[i]}}_{\text{bndry}}(0) = \{p^{[i]}(0)\}$, $p_0^{[i]}(0) = p^{[i]}(0)$.

For $i \in \{1,\dots,n\}$, agent i executes at each time $t \in \mathbb{N}$:

1: acquire $w^{[j]}(t)$, $j \in \mathcal{N}(i)$

2: compute ${P^{[i]}}_{\text{bndry}}(t+1) = \mathrm{MEB}_{\text{bndry}}(W_{\mathcal{N}(i)}(t))$, $W_{\mathcal{N}(i)}(t) = \{w^{[j]}(t) \mid j \in \mathcal{N}(i) \cup i\}$

3: update $w^{[i]}(t+1) = ({P^{[i]}}_{\text{bndry}}(t+1), p_0^{[i]}(t))$

Remark 4.3: For the algorithm to converge it is important that each agent keeps in memory the coordinates of its initial position and thus computes the minimal enclosing ball on the points received from its neighbors together with the point located in its initial position. In fact a point on the boundary on the minimal enclosing ball of n_1 points is not ensured to be on the boundary of the ball of $n_2 \leq n_1$ points. This means that the coordinates of the agents on the boundary could be taken out from the logic variables during the first iterations. This does not happen, for example, for the minimal enclosing orthotope. The result is a simplified consensus algorithm. □

We are now ready to prove the algorithm's correctness.

Theorem 4.4 (Correct MEB *computation):* Let $\mathcal{S}$ be a robotic network such that the agents can communicate according to some communication edge map E_{cmm}. For any $\mathcal{CC}$ such that the graph remains connected along the evolution, the *FloodMEB* algorithm achieves consensus on minimal enclosing ball. □

Proof: In order to prove correctness of the algorithm, observe, first of all, that each law at every node converges in a finite number of steps. In fact, using Lemma 4.1, each sequence corresponds to a ball whose radius is monotone nondecreasing, upper bounded and can assume a finite number of values. Then we proceed by contradiction to prove that all the laws converge to the same ball (same radius and center) and that it is exactly the minimal enclosing ball of the n points. Suppose that the algorithm converges on different balls (different radius or different center) for different agents. Then there must exist two agents that are neighbors and have different logic variables (corresponding to different balls). But this means that, at the following time instant, they have to compute the minimal enclosing ball of a larger set of points, then either one of them will take the value of the other or both of them will change their value and take a common one. Iterating this argument we obtain that all the agents must converge to a common value. Now, the ball at each node contains, by construction, the initial position of that node. Since the ball is the same for each node, it contains all the initial positions, then it is the minimal enclosing ball of the initial positions. ■

Remark 4.5: If we admit that the agents have different priority, the initial positions of the agents can be shared by all the agents in time of order $\Theta(n^2)$. The algorithm is the following. Each agent sends the position of the agent with higher (or equivalently lower) priority that he has in memory. Each position takes time $\Theta(n)$ to spread in the network, therefore the total time complexity is $\Theta(n^2)$. Even if we did not provide any bound for the time complexity of *FloodMEB* algorithm, however simulations suggest that it should be of order $\Theta(n)$. Moreover, while the algorithm for sharing the initial position needs to store a number of packets of order $\Theta(n)$, the *FloodMEB* algorithm needs to store only $d+2$ packets. □

Here is an informal description of what we shall refer to as the *FloodMEO* algorithm:

> *[Informal description]* Each agent initializes the minimal enclosing orthotope to its initial position, then, at each communication round, performs the following tasks: (i) it acquires from its neighbors a message represented by the coordinates of their current minimal enclosing orthotope; (ii) it computes the minimal of its and its neighbors' enclosing orthotopes; (iii) it stores as new message the coordinates of the minimal enclosing orthotope computed at the previous step.

A more formal description of the algorithm is given in the following table.

Name:	*FloodMEO* algorithm.
Goal:	Solve the problem of computing minimal enclosing orthotope of a set of points.
Logic state:	$w^{[i]} = \{w_a^{[i]}\}_{a\in\{1,\dots,d\}} = \{(p_{min,a}^{[i]}, p_{max,a}^{[i]})\}_{a\in\{1,\dots,d\}}$
Msg function:	$\text{msg}(x^{[i]}, w^{[i]}, i) = w^{[i]}$
Initialization:	$p_{min,a}^{[i]}(0) = p_a^{[i]}(0)$, $p_{max,a}^{[i]}(0) = p_a^{[i]}(0)$

For $i \in \{1,\dots,n\}$, agent i executes at each time $t \in \mathbb{N}$:

1: acquire $w^{[j]}$, $j \in \mathcal{N}(i)$
2: compute $\forall a \in \{1,\dots,d\}$
$p_{min,a}^{[i]}(t+1) = \min_{j\in\mathcal{N}(i)\cup\{i\}}\{p_{min,a}^{[j]}(t)\}$
$p_{max,a}^{[i]}(t+1) = \max_{j\in\mathcal{N}(i)\cup\{i\}}\{p_{max,a}^{[j]}(t)\}$
3: update $\forall a \in \{1,\dots,d\}$
$w_a^{[i]}(t+1) = \left(p_{min,a}^{[i]}(t+1), p_{max,a}^{[i]}(t+1)\right)$

In the following theorem we prove the correctness of this algorithm, together with the fact that it reaches consensus in minimum time.

Theorem 4.6 (Correct MEO *computation):* Let $\mathcal{S}$ be a robotic network such that the agents can communicate according to some communication edge map E_{cmm}. For any $\mathcal{CC}$ such that the graph remains connected along the evolution, then the *FloodMEO* algorithm achieves consensus on minimal enclosing orthotope. Moreover, it achieves consensus in minimum number of communication rounds given by

$$T_{\textit{FloodMEO}} = \max_{a\in\{1,\dots,d\}} \max_{i\in\{1,\dots,n\}} \{\text{dist}_{\mathcal{G}}(i, i_{\text{min,a}}), \text{dist}_{\mathcal{G}}(i, i_{\text{max,a}})\},$$

where $i_{\text{min,a}}$ and $i_{\text{max,a}}$ are the agents that characterize the boundary of the orthotope in direction a and minimize the topological distance from i.

The time complexity of the algorithm is of order $\Theta(n)$.□

Proof: In order to prove the correctness and the time complexity of the algorithm described in Table **??**, we need to prove that it is equivalent to $2d$ *FloodMax* algorithms for leader election (two for each direction) running simultaneously. Once we have proven that, the results on correctness and time complexity follow from Chapter 4 in [1].

The algorithm is clearly a set of *FloodMax* algorithms for leader election. In fact the boundary of the orthotope in each direction a is given by the coordinates of the points on such boundary which are characterized by the property of having the maximum and minimum value of the ath coordinate respectively.

In order to prove that the exact number of communication rounds needed is $T_{\textit{FloodMEO}}$, simply observe that it is exactly the minimum time for all the leaders to propagate their information through all the network. Hence this is the minimum time for every possible consensus algorithm to converge. But this is exactly the time taken by $2d$ *FloodMax* algorithms running simultaneously and therefore the time taken by *FloodMEO*. ■

In the following lemma we give, for both *FloodMEB* and *FloodMEO* algorithms, a bound on the time needed by each agent to decide that the algorithm has reached consensus.

Lemma 4.7 (Termination condition): Consider a network $\mathcal{S}$, where the *FloodMEB* (*FloodMEO*) algorithm is running. Each agent can decide that the algorithm has reached consensus if the value of its MEB (MEO) has not changed after $\text{diam}_{\mathcal{G}}$ communication rounds. □

Proof: In order to prove the claim we proceed by contradiction. Suppose that after $\text{diam}_{\mathcal{G}}$ communication rounds the MEB (MEO) of agent i (for some $i \in \{1, \dots, n\}$) has not changed and the algorithm has not converged yet. Then there will exist a $T > \mathrm{diam}_{\mathcal{G}}$ such that the MEB (MEO) of agent i will change to a new value. But this means that the new value, stored T rounds before by some other agent j, took a number of communication rounds greater than $\mathrm{diam}_{\mathcal{G}}$ to arrive from j to i and this contradicts the definition of diameter of $\mathcal{G}$. ■

V. Constant factor approximation of minimum time rendezvous control and communication law

The centralize solution for minimum time rendezvous and the consensus algorithms studied in the previous section suggest a dynamic control and communication law that plays a key role in the minimum time rendezvous problem.

Here is an informal description of what we shall refer to as the *move-toward-*MBC *(*MOC*) control and communication law*, $\mathcal{CC}_{\mathrm{MEB}}$ ($\mathcal{CC}_{\mathrm{MEO}}$):

> Each agent initializes its logic variables to its initial position, then, at each communication round, performs the following tasks: (i) it acquires from its neighbors a message given by their logic variables and positions; (ii) it runs, as state transition function, the *FloodMEB(MEO)* algorithm; (iii) it moves toward the center of the current ball (orthotope) while maintaining connectivity.

Next, we formally define the law as follows. First we assume that each agent operates according to the standard message-generation function, that is $\mathrm{msg}(x^{[i]}, w^{[i]}, i) = (x^{[i]}, w^{[i]})$. Second, before the *FloodMEB* (or *FloodMEO*) algorithm reach consensus, connectivity is maintained by restricting the allowable motion of each agent in some appropriate manner. The exact algorithm can be found for example in [6].

The state transition function implements the *FloodMEB* and *FloodMEO* algorithms respectively, with logic variables as defined in the two tables above.

Define the control function $\mathrm{ctl} : \mathbb{R}^d \times \mathbb{R}^d \times L^n \to \mathbb{R}^d$ for each agent $i \in \{1, \dots, n\}$ by:

$$\mathrm{ctl}(p^{[i]}, w^{[i]}, y^{[i]}) = \max\{\lambda_* \cdot (p_{rndzvs}(w^{[i]}, y^{[i]}) - p^{[i]}), r_{\mathrm{ctr}}\} \cdot \mathrm{vers}(p_{rndzvs}(w^{[i]}, y^{[i]}) - p^{[i]}),$$

with

$$p_{rndzvs}(w^{[i]}, y^{[i]}) = \mathrm{MBC}(w^{[i]}, y^{[i]}) \tag{2}$$

and λ_* is chosen in order to maintain connectivity until consensus is reached.

In a network with communication edge map $E_{\mathrm{cmm}} = E_{\mathrm{cube}}$ the procedure described above is applied separately in every direction $a \in \{1, \dots, d\}$.

The correctness of the two control and communication laws is proven in the following lemma.

Lemma 5.1 (Correctness of $\mathcal{CC}_{\mathrm{MEB}}$ and $\mathcal{CC}_{\mathrm{MEO}}$): On the network $\mathcal{S}$ with communication edge map E_{ball} or E_{cube} and bound on the ith control input $u^{[i]} \in B(0, r_{\mathrm{ctr}})$ or $u^{[i]} \in C(0, r_{\mathrm{ctr}})$, the control and communication laws $\mathcal{CC}_{\mathrm{MEB}}$ and $\mathcal{CC}_{\mathrm{MEO}}$ achieve rendezvous at $\mathrm{MBC}(p^{[1]}(0), \dots, p^{[n]}(0))$ and $\mathrm{MOC}(p^{[1]}(0), \dots, p^{[n]}(0))$ respectively. □

Proof: By the connectivity arguments done before and by Theorem 4.4 and Theorem 4.6 we know that there exists $\overline{T} \in \mathbb{N}$ such that for $t = \overline{T}$ the network is connected and all the agents have reached consensus on $\mathrm{MBC}(p^{[1]}(0), \dots, p^{[n]}(0))$ (or $\mathrm{MOC}(p^{[1]}(0), \dots, p^{[n]}(0))$). Since this instant all the agents can move toward the same point (at maximum speed) without enforcing connectivity constraint anymore. Thus, they can converge to the rendezvous point which is exactly $\mathrm{MBC}(p^{[1]}(0), \dots, p^{[n]}(0))$ (or $\mathrm{MOC}(p^{[1]}(0), \dots, p^{[n]}(0))$). ■

A. Time complexity of $\mathcal{CC}_{\mathrm{MEB}}$ and $\mathcal{CC}_{\mathrm{MEO}}$

In the previous lemma we have proven that the control and communication laws $\mathcal{CC}_{\mathrm{MEB}}$ and $\mathcal{CC}_{\mathrm{MEO}}$ achieve consensus. Now we ask how fast these laws are depending on the control bound r_{ctr} and the number of agents.

Theorem 5.2: For $r_{\mathrm{cmm}} \in \mathbb{R}_+$, $d \in \mathbb{N}$, consider the network S with communication edge map either E_{disk} or E_{cube}. The following statements hold:

(i) for $u^{[i]} \in B(0, r_{\mathrm{ctr}})$, $i \in \{1, \dots, n\}$, the control and communication law $\mathcal{CC}_{\mathrm{MEB}}$ asymptotically converges to the minimum time rendezvous centralized solution $\mathcal{MTR}(E_{\mathrm{cmpl}}, B(0, r_{\mathrm{ctr}}))$ as $r_{\mathrm{ctr}} \to 0^+$ (for all fixed n).

(ii) for $u^{[i]} \in C(0, r_{\mathrm{ctr}})$, $i \in \{1, \dots, n\}$, the control and communication law $\mathcal{CC}_{\mathrm{MEO}}$ converges to the minimum time rendezvous centralized solution $\mathcal{MTR}(E_{\mathrm{cmpl}}, C(0, r_{\mathrm{ctr}}))$ for $r_{\mathrm{ctr}} \to 0^+$ (for all fixed n). Moreover, it is a constant factor approximation of $\mathcal{MTR}(E_{\mathrm{cmpl}}, C(0, r_{\mathrm{ctr}}))$, i.e., $\mathrm{TC}(\mathcal{T}_{\mathrm{rndzvs}}, \mathcal{CC}_{\mathrm{MEO}}) \in \Theta(\frac{n}{r_{\mathrm{ctr}}})$ for $r_{\mathrm{ctr}} \to 0^+$ and $n \to +\infty$. □

Before proving the theorem let us state a useful lemma.

Lemma 5.3 ([8]): For all pointsets $P_1 \subset P$, we have $\mathrm{MBC}(P_1) \in \mathrm{MEB}(P)$ and $\mathrm{MOC}(P_1) \in \mathrm{MEO}(P)$. □

Proof: [Theorem 5.2] The line of proof of the two statements is the same, hence we prove only the second one which is stronger due to the stronger result on the time complexity of the *FloodMEO* algorithm. Using the previous lemma we know that for all $t \in \mathbb{N}_0$, then $p^{[i]}_{rndzvs}(t) \in \mathrm{MEO}(p^{[1]}(0), \dots, p^{[n]}(0))$, where $p^{[i]}_{rndzvs}$ is defined as in (2). This implies that, once the consensus is reached, the time to rendezvous is upper bounded by the time of the centralized solution. Hence the following bound on the time of convergence of $\mathcal{CC}_{\mathrm{MEO}}$, T_{MEO}, holds:

$$T_{\mathrm{MEO}} \leq \left\lceil \mathrm{diam}(p^{[1]}(0), \dots, p^{[n]}(0)) \cdot \frac{1}{r_{\mathrm{ctr}}} \right\rceil + T_{\mathit{FloodMEO}}. \tag{3}$$

The first statement is proven by observing that T_{FloodMEO} does not depend on r_{ctr}, therefore, as $r_{\mathrm{ctr}} \to 0^+$, T_{MEO} converges to the optimal value of the centralized case.

In order to prove the second statement, observe that $\mathrm{diam}(p^{[1]}(0), \dots, p^{[n]}(0)) \leq (n-1) r_{\mathrm{cmm}}$ and $T_{\mathit{FloodMEO}} \in \Theta(n)$. The result follows by substituting these bounds in (3). ■

Remark 5.4: The previous theorem confirms the intuitive idea that, if the communication is much faster than the motion (r_{ctr} small), then the optimal solution in the distributed case converges to the one of the centralized case. □

B. Distributed minimum time rendezvous in one dimension

In one dimension (all the agents spread on a line), we can find a condition on r_{ctr} ensuring that the move-toward-MBC algorithm is the solution of $\mathcal{MTR}(E_{\text{disk}}, B(0, r_{\text{ctr}}))$.

Theorem 5.5: For $d = 1$, let i_{max} and i_{min} the agents in the network $\mathcal{S}$ with the maximum and minimum positions. If $r_{\text{ctr}} < \frac{1}{4} r_{\text{cmm}}$ and both $\max_{j \in \mathcal{N}(i_{\text{max}})} \{p^{[i_{\text{max}}]}(0) - p^{[j]}(0)\} \geq 2r_{\text{ctr}}$ and $\max_{j \in \mathcal{N}(i_{\text{min}})} \{p^{[j]}(0) - p^{[i_{\text{min}}]}(0)\} \geq 2r_{\text{ctr}}$, then the control and communication law $\mathcal{CC}_{\text{MEB}}$ solves the task $\mathcal{T}_{\text{rndzvs}}$ in minimum time which is exactly the time of the centralized solution, that is, $T^* = \left\lceil \frac{\text{diam}(p^{[1]}, \ldots, p^{[n]})}{r_{\text{ctr}}} \right\rceil$. □

Proof: Consider the input sequence of the centralized solution for the agent i_{min} (and equivalently for i_{max}). It is $u^{[i_{\text{min}}]}(t) = r_{\text{ctr}}$ for all $t < T^* - 1$ and $u^{[i_{\text{min}}]}(T^* - 1) = \text{MBC}(p^{[1]}(0), \ldots, p^{[n]}(0)) - p^{[i_{\text{min}}]}(T^* - 1)$. Since the rendezvous time is bounded by the time that i_{max} and i_{min} take to reach $\text{MBC}(p^{[1]}(0), \ldots, p^{[n]}(0))$, we need to prove that, as long as the consensus on the minimal enclosing ball is not reached, then $u^{[i_{\text{min}}]}(t) = -u^{[i_{\text{max}}]}(t) = r_{\text{ctr}}$. Due to the symmetry of the problem we will give the proof only for i_{min}. It can be easily shown that for all $t \geq 1$ such that $p_{max}^{[i_{\text{min}}]}(t) \neq p^{[i_{\text{max}}]}(0)$ (consensus is not reached), the following holds:

$$p_{max}^{[i_{\text{min}}]}(t+1) > p_{max}^{[i_{\text{min}}]}(t-1) + r_{\text{cmm}}.$$

It follows:

$$\text{MBC}^{[i_{\text{min}}]}(t+1) = \frac{1}{2}(p_{max}^{[i_{\text{min}}]}(t+1) + p^{[i_{\text{min}}]}(0)),$$

then

$$\begin{aligned} \text{MBC}^{[i_{\text{min}}]}(t+1) &> \frac{1}{2}(p_{max}^{[i_{\text{min}}]}(t-1) + r_{\text{cmm}} + p^{[i_{\text{min}}]}(0)) \\ &= \text{MBC}^{[i_{\text{min}}]}(t-1) + \frac{1}{2} r_{\text{cmm}} \\ &\geq \text{MBC}^{[i_{\text{min}}]}(t) - r_{\text{ctr}} + \frac{1}{2} r_{\text{cmm}}. \end{aligned}$$

This leads to

$$-r_{\text{ctr}} + \frac{1}{2} r_{\text{cmm}} > r_{\text{ctr}}, \quad \text{and} \quad r_{\text{ctr}} < \frac{1}{4} r_{\text{cmm}}.$$

The other two assumptions ensure the condition for $t = 0$. ■

C. Simulations

In order to illustrate the performance of our rendezvous algorithms, we implemented the move-toward-MBC algorithm, based on the *FloodMEB* consensus algorithm. We implemented it in the plane, $d = 2$, over the disk graph. The simulation run is illustrated in Figure 1. The 32 agents have a bound on the control inputs $r_{\text{ctr}} = 0.1$, and a communication radius $r_{\text{cmm}} = 3$. The initial positions of the agents were randomly generated over the rectangle $[-6, 6] \times [-3, 3]$.

The *FloodMEB* law converges in five steps, while the rendezvous is achieved at $T = 58$. As it clearly appears in the figure, once the consensus on the minimal enclosing ball is reached, all the agents move toward the center.

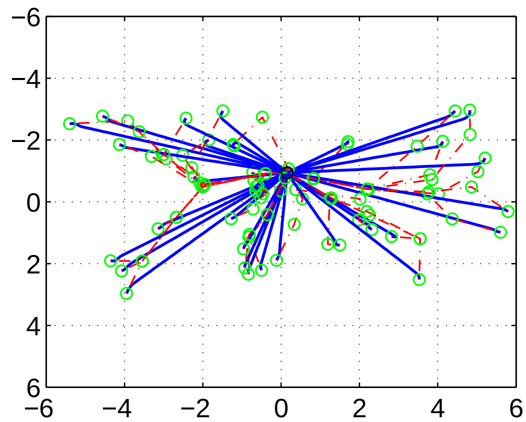

Fig. 1. Evolution of the network (in filled blue) according to $\mathcal{CC}_{\text{MEB}}$ with evolution of *FloodMEB* (green circles connected by dashed red line)

VI. Conclusions

We have presented some simple algorithms on how to compute optimal enclosing shapes for pointsets via distributed computation. These algorithms are then used to provide efficient solutions to distributed rendezvous problems for synchronous robotic networks. For future work we envision characterizing the time complexity of the *FloodMEB* algorithm and, in turn, of the move-toward-MBC control and communication law.

References

[1] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers, 1997.

[2] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, “Distributed memoryless point convergence algorithm for mobile robots with limited visibility,” *IEEE Trans. Robotics & Automation*, vol. 15, no. 5, pp. 818–828, 1999.

[3] J. Lin, A. S. Morse, and B. D. O. Anderson, “The multi-agent rendezvous problem,” in *Proc CDC*, Maui, HI, Dec. 2003, pp. 1508–1513.

[4] Z. Lin, M. Broucke, and B. Francis, “Local control strategies for groups of mobile autonomous agents,” *IEEE Trans. Automatic Ctrl*, vol. 49, no. 4, pp. 622–629, 2004.

[5] J. Cortés, S. Martínez, and F. Bullo, “Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions,” *IEEE Trans. Automatic Ctrl*, vol. 51, no. 8, pp. 1289–1298, 2006.

[6] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, “On synchronous robotic networks – Part I: Models, tasks and complexity notions. & Part II: Time complexity of rendezvous and deployment algorithms,” *IEEE Trans. Automatic Ctrl*, Apr. 2005, submitted.

[7] W. Ren, R. W. Beard, and T. W. McLain, “Coordination variables and consensus building in multiple vehicle systems,” in *Cooperative Control*, V. Kumar, N. E. Leonard, and A. S. Morse, Eds. Springer Verlag, 2004, vol. 309, pp. 171–188.

[8] K. Fischer and B. Gärtner, “The smallest enclosing ball of balls: Combinatorial structure and algorithms,” *Int. J. Computational Geom. & App.*, vol. 14, no. 4-5, pp. 341–378, 2004.