

Optimal workload sharing for mobile robotic networks in dynamic environments

Marco Pavone Ketan Savla Emilio Frazzoli

I. INTRODUCTION

Modern technological advances make the deployment of large groups of autonomous mobile agents with on-board computing and communication capabilities increasingly feasible and attractive. In the near future, large groups of such autonomous agents will be used to perform complex tasks in dynamic environments including transportation and distribution, logistics, surveillance, search and rescue operations, humanitarian demining, environmental monitoring, and planetary exploration.

The potential advantages of multiple mobile agents are, in fact, numerous. For instance, the availability of real-time data collected *in situ* would dramatically impact the modeling and study of several critical, episodic, rapidly evolving, and localized environmental phenomena, such as hurricanes, tsunamis, oil spills, and forest fires. Similar considerations can be made for the study of currents, winds, and accurate local weather forecasting. Collection of such data is currently limited by the fact that most available sensors are static (e.g., fixed monitoring stations), or remote (e.g., satellites). Moreover, the intrinsic parallelism associated with a multi-agent system provides robustness to failures of single agents, and in many cases can guarantee better time efficiency. It is also possible to reduce the total implementation and operation cost, increase reactivity and system reliability, and add flexibility and modularity to monolithic approaches. As a consequence, the interest of the systems and control, robotics, computer science, and networking communities for systems comprising multiple mobile agents has increased rapidly in the last several years.

With such a powerful technology at hand, the major scientific challenge is to design efficient coordination policies between these agents. In essence, the agents can be interpreted as *resources* to be allocated to *customers*. In surveillance and exploration missions, customers are points of interest to be visited; in transportation and distribution applications, customers are people demanding some service (e.g., utility repair) or goods; in logistics tasks, customers could be troops in the battlefield. Finally, consider a possible architecture for networks of autonomous agents performing distributed sensing: a set of n cheap sensing devices (sensing nodes), distributed in the environment, provides sensor measurements, while m sophisticated agents (cluster heads) collect information from the sensing nodes and transmit it (possibly after some computation) to the outside world. In this case, the sensing nodes represent customers, while the agents, acting as cluster heads, represent resources to be allocated.

Marco Pavone, Ketan Savla and Emilio Frazzoli are with the Laboratory for Information and Decision Systems, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, {pavone, ksavla, frazzoli}@mit.edu.

The underlying mathematical problem in many of these applications can be studied within the framework of spatial queues with the tasks as customers and the mobile agents as servers. The solution has to typically address two key challenges: task allocation among the agents and service schedule for each agent. In general, these two features are highly coupled and, therefore, devising an optimal or, at least, a provably efficient policy is an extremely difficult problem. Considering motion constraints for the agents, as should be done for unmanned aerial vehicles (UAVs), complicates things further.

A natural way to reduce the complexity is to partition the workspace among the agents and then let each agent follow a certain set of rules in its own region. To which extent does this decoupling strategy affect optimality? The objective of this article is to illustrate specific scenarios and recently developed partitioning schemes whereby one can retain optimality, or at least some degree of optimality, even under this decomposition.

Specifically, the contributions of this article are as follows. First, we describe the characteristics of a family of dynamic vehicle routing problems that indeed capture the main features of a large number of dynamic task-based motion coordination problems, appearing in a variety of foreseen application domains. Second, we are interested in identifying specific scenarios in which one can decompose, through workspace partitioning, task allocation and service scheduling, while maintaining optimality for the original problem. For these scenarios, we discuss which types of partitions should indeed be used: in some cases, the “optimal” partitions coincide with well-known tessellation of the plane from computational geometry (e.g., Voronoi diagrams), while, in other cases, new types of partitions should be used. Third and final, we are interested in designing efficient algorithms that achieve such spatial partitions.

II. A FAMILY OF DYNAMIC VEHICLE ROUTING PROBLEMS

Consider a geographical region \mathcal{Q} , in which a certain dynamic process generates spatially localized “service requests”.¹ Service requests can represent, for example, pick-up/delivery points, or events of interest that require close investigation or measurement in an environmental monitoring scenario; in a biological setting, a service request might represent the availability of a food item, or other scarce resource. The process generating service request is usually modeled as a spatio-temporal Poisson² process with temporal

¹Henceforth, if not otherwise stated, we assume that \mathcal{Q} is a compact, convex subset of \mathbb{R}^2 .

²Indeed, most of the results presented in this article are also valid when the arrival process is a general renewal process.

intensity $\lambda > 0$ and spatial distribution $\varphi(q)$ over \mathcal{Q} (i.e., upon arrival, the locations of service requests are identically and independently distributed according to $\varphi(q)$), and can be known to various degrees of accuracy. A total of m mobile agents provide service in \mathcal{Q} . In their simplest form, service requests are fulfilled when one of the agents moves to the request location—possibly within a given tolerance, determined by the characteristics of the sensory equipment on board the agents. We assume that agents are able to service an unlimited numbers of requests. Service requests can either stay until satisfied, or they might expire after a certain time. The time service requests remain active can be itself a random variable, describing customer impatience. There could be multiple classes of service requests (i.e., *heterogeneous* service requests), some of them requiring higher priority service.

Furthermore, the agents can be subject to several differential constraints due to their dynamic capabilities; in particular, most vehicles of practical interest are subject to first- or second-order non-holonomic constraints (e.g., wheeled vehicles, aircraft, and ships). The information available to the agents can also be limited in several ways. For example, agents might be able to communicate *directly* only with other agents (or static nodes) that lie within a certain radius, or might not have any communication capability (e.g., when extremely cheap mobile sensors are sought).

Finally, it is possible that no central authority is available, which is aware of the state of all agents, and can process all the information available to the system. Thus, the design of cooperative policies is also affected by the mode of its implementation (centralized or decentralized).

It is desired to maximize the Quality of Service delivered by the mobile agents, for example, in terms of the average or worst-case time delay between the issuance of a service request and the time it is fulfilled. When service requests have a finite lifetime (i.e., they can expire), another parameter of interest is the fraction of service requests that is fulfilled before expiration. In general, the focus is on the quality of service as perceived by the “end user,” rather than, for example, fuel economies achieved by the mobile agents.

The basic version of the problem in which the agents do not have any differential constraint, service requests do not expire and do not have any priority, a central dispatcher is available, and the objective is to minimize the average waiting time of service requests is known in the literature as the Dynamic Traveling Repairman Problem (DTRP). The main reference for DTRP is currently [1].

III. SPATIAL TESSELLATIONS FOR WORKLOAD SHARING

In this section, we identify specific cases, within the previous family of dynamic vehicle routing problems, in which one can decompose, by using *partitioning policies*, task allocation and service scheduling, while maintaining optimality for the original problem. For each case, we discuss which types of spatial tessellations (partitions) should indeed be used.

Specifically, a *partitioning policy* is an algorithm that, as a function of the number m of agents and, possibly, of their position and other information, partitions a bounded workspace \mathcal{Q} into sub-regions Q_i , for $i \in \{1, \dots, m\}$. Then, each agent i is assigned to sub-region Q_i , and each customer in Q_i receives service from the agent assigned to Q_i .

We start by discussing the most intuitive partitioning policy, namely partitioning \mathcal{Q} into sub-regions of equal area.

Equitable partitions

The most widely applied resource allocation strategy is to equalize the total workload assigned to each resource, i.e., in our context, to each agent. Indeed, assume that the traffic load is *large* (i.e., the fraction of time the service vehicles spend performing on-site service approaches one). Then, in [1], it is argued that an *optimal* m -agent policy for the Dynamic Vehicle Routing Problem can be constructed from an optimal single-agent policy for the DTRP by using *equitable* partitioning policies, where all subregions have *equal measure* with respect to $\varphi(q)$; a couple of such possible equiv-partitions are depicted in Figure 1. This remarkable, and somewhat surprising, result implies that, as the traffic load increases, the average service delay time (waiting time, for short) tends to depend only on the measure of each sub-region; in other words, as the traffic load increases, the shape of the subregions tends to become insignificant. This is primarily because agents, when the traffic load is large, have to spend very little time traveling between locations of various service requests (since \mathcal{Q} is densely filled with service requests).

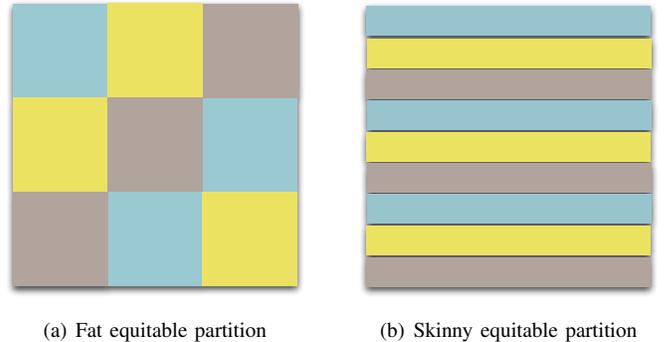


Fig. 1. Different kinds of equipartitions for uniform spatial density $\varphi(q)$

Recently, similar results have been proven for some generalizations of the DTRP problem. In particular, in [2], we studied a generalization of the DTRP problem in which service requests might expire after a certain deterministic time T (called time window). Time windows constraints are indeed common in many applications, including bank deliveries, postal deliveries, grocery distribution, dial-a-ride service, bus routing, and repairmen scheduling. The objective, then, is to minimize the number of agents needed to ensure that a certain fraction ε of service requests is fulfilled before expiration. We proved that in *heavy load* (i.e., when the traffic load is large), and when $\varepsilon \rightarrow 1^-$ (i.e., almost all service requests have to be fulfilled before expiration), an equitable partitioning policy in which each agent services outstanding service requests inside its own sub-region by repeatedly forming optimal (i.e., of minimum length) tours provides a constant-factor approximation³, with a constant equal to 3.8. In particular, a number of

³Given a minimization problem and an objective function J , we say that a policy π provides a constant-factor approximation if $J_\pi \leq cJ^*$, where J^* is the optimal value and c is a positive constant. Constant-factor approximations are, often, the best results one can hope for in complicated minimization problems.

agents sufficient to guarantee, in heavy load, that almost all service requests are serviced before expiration, is

$$m = \left\lceil \sqrt{\lambda} \left(\beta \sqrt{\frac{2}{T}} \right) \right\rceil,$$

where $\beta = 0.712$.

Another generalization of the DTRP in which equitable partitioning policies provide constant-factor approximations is the problem of dynamic vehicle routing with heterogeneous demands, introduced in [3]. The set-up is the same as for the DTRP problem, but we assume that there are n classes of service requests; given coefficients $c_1, \dots, c_n > 0$, $\sum_{i=1}^n c_i = 1$, the goal is to find the vehicle routing policy that minimizes the convex combination

$$c_1 D_1 + \dots + c_n D_n,$$

where D_i is the expected delay for service requests of class i . By increasing the coefficients for certain classes, a higher priority level can be given to their service requests. This problem has important applications in areas such as UAV surveillance, where targets are given different priority levels based on their urgency or potential importance. In [4], we showed that, in heavy load, an equitable partitioning policy in which each agent services, by forming optimal tours, groups of class i service requests with probability $p_i = c_i$ provides a constant factor approximation, with a constant equal to $2n^2$ (that, undesirably, depends on the number of classes).

All previous results hold for first-order holonomic agents and in heavy load. In [5], we study a generalization of the DTRP in which agents have non-holonomic motion constraints (as it is the case when agents are fixed-wing UAVs). In particular, we modeled the UAVs as a variant of the well-studied Dubins model⁴, where the agents are constrained to move with constant forward speed along paths whose radius of curvature is uniformly upper bounded by the parameter $1/\rho$. For the heavy load case of this problem, we proved that equitable partitioning policies still provide a constant factor approximation, but the shape of subregions becomes relevant. In particular, if \mathcal{Q} is a rectangular region, the workspace should be partitioned into strips formed by placing equidistant dividers parallel to the width (i.e., the largest of the two dimensions of the rectangular region). For a general convex region, one can optimize the performance by tuning the distance between the dividers to the shape of \mathcal{Q} in the following way. Let W and H be the width and height of a minimum rectangle enclosing \mathcal{Q} . Form strips by placing dividers parallel to the width of the enclosing rectangle in the following way. Let $|Q_1|, \dots, |Q_m|$ be the areas of the individual strips and let H_1, \dots, H_m be the heights of the resultant strips. Note that, $\sum_{i=1}^m |Q_i| = |\mathcal{Q}|$ and $\sum_{i=1}^m H_i = H$ and that $|Q_i|$ are functions of H_i and \mathcal{Q} . Then the inter-strip distances H_i are selected so that

$$|Q_i| + \frac{7}{3} \pi \rho H_i = \frac{1}{m} (|\mathcal{Q}| + \frac{7}{3} \pi \rho H) \quad \forall i \in \{1, \dots, m\} \quad (1)$$

Note that this rule for space partitioning could be regarded as a general rule for the heavy load case, in the sense that, as

⁴Henceforth, in this paper, we shall implicitly use this model for fixed-wing UAVs

$\rho \rightarrow 0^+$ (which corresponds to first order holonomic vehicles), the corresponding strips would correspond to regular equitable partitions, which we have already stated to be optimal for the first order agents.

All the previous results hold under the heavy load assumption; when the load is only moderate, the shape of subregions can have a significant effect. In moderate traffic, and with holonomic first-order agents, a solution that turns out to be effective (although there is no certificate of optimality) is to adopt equitable partitioning policies in which the subregions are “fat” (i.e., with a small diameter for a given area), rather than long and thin.

We next investigate which partitioning policies should be used, instead, in light load (i.e., when the fraction of time the service vehicles spend performing on-site service is close to zero).

Median Voronoi Tessellation

We first introduce the concept of Voronoi tessellations (or Voronoi diagrams). The use of Voronoi tessellations is ubiquitous in many fields of science, ranging from operations research, animal ethology (territorial behaviour of animals), computer science (design of algorithms), to numerical analysis (construction of adaptive grids for PDEs and general quadrature rules), and algebraic geometry (moduli spaces of abelian varieties). We now give its formal definition. A detailed exposition of Voronoi Tessellations is given in [6].

Define $G \doteq (g_1, \dots, g_m) \in \mathcal{Q}^m$. The *Voronoi Diagram* $\mathcal{V}(G) = (V_1(G), \dots, V_m(G))$ of \mathcal{Q} generated by points G is defined by

$$V_i(G) = \{q \in \mathcal{Q} \mid \|q - g_i\| \leq \|q - g_j\|, \forall j \neq i, j \in I_m\},$$

where $I_m \doteq \{1, 2, \dots, m\}$. We refer to G as the set of *generators* of $\mathcal{V}(G)$, and to $V_i(G)$ as the Voronoi cell of the i -th generator. A Voronoi diagram $\mathcal{V}(G) = \{V_1(G), \dots, V_m(G)\}$ of \mathcal{Q} is called a *Median Voronoi Tessellation* of \mathcal{Q} with respect to the density function φ if the ordered set of generators G is equal to the ordered set of generalized medians of the sets in $\mathcal{V}(G)$ with respect to φ , i.e., if

$$g_i = \arg \min_{g \in \mathbb{R}^2} \int_{V_i(G)} \|g - q\| \varphi(q) dq, \quad \forall i \in \{1, \dots, m\}.$$

It is possible to show that a Median Voronoi Tessellation always exists for any domain \mathcal{Q} and density φ . It is interesting to note that, as the number of generators increases, the Median Voronoi Tessellation for a given region starts assuming the hexagonal honeycomb structure [7].

A Median Voronoi Tessellation is proven [1] to be *almost* optimal for the scenario involving an ensemble of first order agents responding to service requests which are generated very rarely (i.e., in light load). The median locations in this case can be understood to be loitering locations for the agents, i.e., locations at which the agents wait when there are no outstanding service request. The *almost* optimality in the light load case is to be understood as follows. If one defines the average waiting time for the service requests to be a function of the loitering locations of the agents, then the generalized median locations that give rise to the Median Voronoi Tessellation correspond to the local minima or the

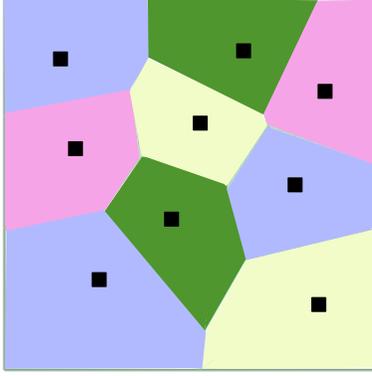


Fig. 2. A Voronoi diagram with 9 generators (generators are represented by black squares).

saddle point of this function. Additionally, even for networks of UAVs (Dubins vehicle) or tanks (differential drive robots) or rotary wing aircrafts (double integrator robots), the Median Voronoi Tessellation is proven [5], [8] to be almost optimal (in the same sense, as above) when the robotic network is sparse, i.e., the density of agents is low. For fixed-wing UAVs, the median locations in that case correspond to center of loitering circles, as illustrated in Figure 3. The primary reason for optimality of Median Voronoi Tessellations for sparse robotic networks in the light load case is that, since the agents are well-separated from each other, the distance from the loitering location of any agent to the location of a service request can be well approximated by the Euclidean distance between those locations. However, this approximation is very weak when the robotic network starts getting dense, in which case the optimal sharing of workload necessitates a *dynamic* partitioning of space, i.e., space partitions that change as a function of time.

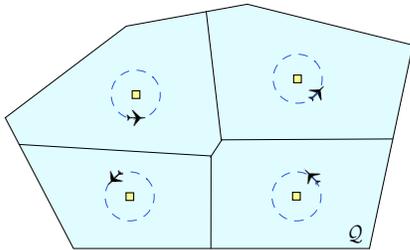


Fig. 3. Fixed-wing UAVs loitering around median locations.

Dynamic partitions

The scope of partitioning policies can be easily extended to include dynamic partitions. In [5], it was shown that for a dense group of fixed-wing UAVs, a dynamic space partition gives a constant factor approximation to the optimal quality of service, in light load. The details of the partition are as follows. Bound the environment \mathcal{Q} with a rectangle of minimum *height*, where we use height to denote the smallest of the two side lengths of a rectangle. Let W and H be the *width* and *height* of this bounding rectangle respectively. Divide \mathcal{Q} into strips

of width w where

$$w = \min \left\{ \left(\frac{4}{3\sqrt{\rho}} \frac{WH + 10.38\rho H}{m} \right)^{2/3}, 2\rho \right\}.$$

Orient the strips along the side of length W . Construct a closed *loitering* path which runs along the longitudinal bisector of each strip, as shown in Figure 4. The m UAVs loiter on this path, equally spaced, in terms of path length. If ℓ is the inter-UAV distance, then the sub-region of \mathcal{Q} allocated to every agent is a rectangle of length ℓ and width w (intersected with \mathcal{Q}) and offset by a distance $\sqrt{\rho w - w^2/4}$ ahead of it, as shown in Figure 4. The offset between the UAV's position and its region of responsibility is to make sure that the UAV can travel to any location in that region in a short time. Note that the dynamic partition associated with a particular agent is in fact fixed in the reference frame of that agent and that in the global frame these partitions could be regarded as a dynamic version of equitable partition of \mathcal{Q} , modulo the parts of regions of responsibility that lie outside \mathcal{Q} .

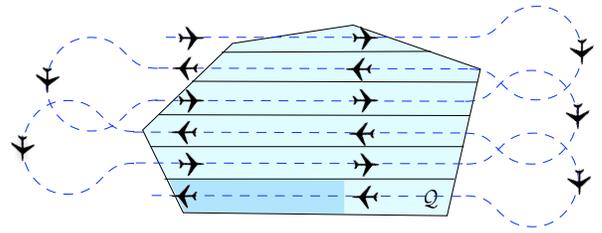


Fig. 4. Loitering pattern for a dense network of UAVs, with the illustration of region of responsibility for one of them

The same kind of dynamic partitions are proven to give constant factor optimal performance for a group of double integrators for a similar scenario in [8]. The reason that such dynamic partitions help to give good performance is because they help to keep the *region of responsibility* for an agent directly in front of it, thereby minimizing the time taken to travel to the location of a service request.

Centroidal Voronoi Tessellations

A Voronoi Diagram $\mathcal{V}(G) = \{V_1(G), \dots, V_m(G)\}$ of a set \mathcal{Q} is called a *Centroidal Voronoi Tessellation* of \mathcal{Q} with respect to the density function φ if the ordered set of generators G is equal to the ordered set of *centroids* of the sets in $\mathcal{V}(p)$ with respect to φ , i.e., if

$$g_i = \arg \min_{g \in \mathbb{R}^2} \int_{V_i(G)} \|g - q\|^2 \varphi(q) dq, \quad \forall i \in \{1, \dots, m\}.$$

It is possible to show that a Centroidal Voronoi Tessellation always exists for any domain \mathcal{Q} and density φ .

Centroidal Voronoi Tessellations have been proven to be (locally) optimal for the *sensor coverage problem*, where it is desired to solve a facility location problem to place mobile sensors to provide optimal coverage in terms of sensing capabilities [9]. Centroidal Voronoi Tessellations are usually partitions where subregions are approximately hexagonal (i.e., fat). Therefore, in our context of dynamic vehicle routing problems, centroidal *and* equitable Voronoi tessellations can be useful in the DTRP with moderate load and holonomic first-order agents, according to the discussion in the previous section.

Phase transition between tessellations

We have illustrated the use of many well-known spatial tessellations for optimal or almost optimal distribution of workload for dynamic task-allocation based scenarios. Mathematically, many of these scenarios could be considered to be variants of the same basic DTRP, in some cases representing extreme regions of the parameter space for that problem. For example, we considered the case when the traffic load is *large*, i.e., the *fraction* of time the service vehicles spend performing on-site service approaches one, and the case when the traffic load is *small*, i.e., the *fraction* of time the service vehicles spend performing on-site service is close to zero. We have seen that different spatial tessellations give optimal performance for these two extremes, suggesting the existence of a phase transition between tessellations driven by *exogenous* factors, such as λ , in such applications. Additionally, we have seen drastically different partitions giving optimal performance, as a network of robots with motion constraints gets denser. This suggests phase transition in tessellations also being driven by *endogenous* factors like density of the robotic network. In [5], we studied such a phase transition between a median Voronoi tessellation (Figure 5) and a dynamic partition (Figure 6) with respect to the density of the robotic network in an unbounded domain. We identified a novel dimensionless parameter, the *non-holonomic density* $:= \rho^2 m / |\mathcal{Q}|$ whose critical value (≈ 0.06) characterizes the corresponding phase transition. Similar *endogenous* phase transitions for other models of robots were investigated in [8]. A formal study of the factors driving such phase transitions would help the system planner to select an appropriate tessellation for a given set of problem parameters. In addition, a fundamental understanding of such phase transitions could help to give a better insight into similar phenomena in naturally occurring systems, e.g., transition from solitary to gregarious behavior in desert locusts, as reported in [10].

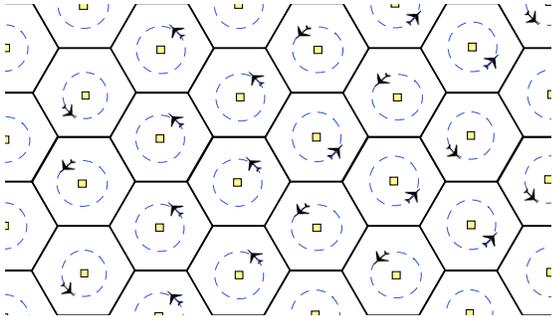


Fig. 5. The median Voronoi tessellation for an unbounded domain with UAVs loitering about median locations.

IV. ALGORITHMS FOR PARTITIONING POLICIES

In this section, we present several algorithms to implement the different partitioning policies presented in Section III.

Algorithms for equitable partitions

If we model the *workload* for sub-region $S \subseteq \mathcal{Q}$ as $\lambda_S \doteq \lambda \int_S \varphi(q) dq$, then the workload for agent i is λ_{Q_i} . Then, equitable partitioning calls for equalizing the workload λ_{Q_i} in the m subregions.

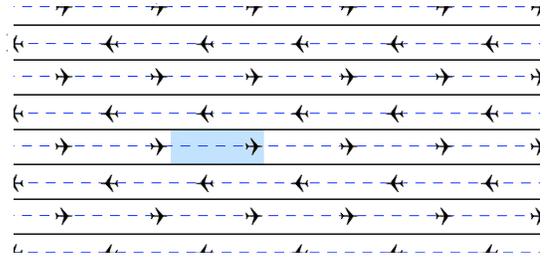


Fig. 6. UAVs loitering in an unbounded domain. The dynamic partition associated with an agent is also illustrated.

We first assume that a central dispatcher is available. In such case, many simple algorithms provide equitable partitions. For example, one could “sweep” \mathcal{Q} from a point in the interior of \mathcal{Q} using an arbitrary starting ray until $\lambda \int_{Q_1} \varphi(q) dq = \lambda/m$, continuing the sweep until $\lambda \int_{Q_2} \varphi(q) dq = \lambda/m$, etc. Sweeping techniques can also be devised for dividing \mathcal{Q} into strips of equal area or width or according to the rule described in Equation (1). If an equitable partition with fat subregions is sought, then an effective solution is to generate equitable and centroidal Voronoi partitions, by simulating in the processor of the central dispatcher the distributed algorithm presented later in this section.

Indeed, to the best of our knowledge, all equitable partitioning policies inherently assume a *central* dispatcher that computes a tessellation of the workspace. This fact is in sharp contrast with the desire of a fully distributed architecture for a multi-agent system. The lack of a fully distributed architecture limits the applicability of equitable partitioning policies to limited-size multi-agent systems operating in a known static environment. If a central dispatcher is not available, then a possible solution is to run a distributed leader election algorithm, let the leader execute one of the centralized algorithms discussed earlier, and finally let the leader broadcast subregion’s assignments to all other agents. Such conceptually simple solution, however, can be impractical in scenarios where the density $\varphi(q)$ changes over time, or agents can fail, since at every parameter’s change a new time-consuming leader election is needed. In [11], then, we introduced a radically different distributed algorithm for equitable partitioning, that does not require any leader election. Such algorithm is described next.

1) *Power Diagrams and Virtual Generators*: In the solution proposed in [11], Power Diagrams are the key geometrical concept. Define $G_W \doteq ((g_1, w_1), \dots, (g_m, w_m)) \in (\mathcal{Q} \times \mathbb{R})^m$. We refer to the pair (g_i, w_i) as a *power point*. In some sense, w_i measures the capability of g_i to influence its neighborhood. The *Power Diagram* $\mathcal{V}(G_W) = (V_1(G_W), \dots, V_m(G_W))$ generated by power points G_W is defined by

$$V_i(G_W) = \{q \in \mathcal{Q} \mid \|q - g_i\|^2 - w_i \leq \|q - g_j\|^2 - w_j, \forall j \neq i, j \in I_m\}.$$

We refer to G_W as the set of *power generators* of $\mathcal{V}(G_W)$, and to $V_i(G_W)$ as the power cell of the i -th power generator. Notice that, when all weights are the same, the Power Diagram coincides with the Voronoi Diagram. Indeed, Power Diagrams are the generalized Voronoi Diagrams that have the strongest

similarities to the original diagrams. From the definition, it is easy to see that each cell $V_i(G_W)$ is a convex set.

The key advantage of Power Diagrams is that an equitable Power Diagram always exists for any density $\varphi(q)$. More precisely, given a measure $\varphi(q)$ with support on \mathcal{Q} and $m \geq 1$ distinct points (g_1, \dots, g_m) in \mathcal{Q} , there exist weights w_i , $i \in \{1, \dots, m\}$, such that the power points $((g_1, w_1), \dots, (g_m, w_m))$ generate a Power Diagram that is equitable with respect to $\varphi(q)$.

Then, the first step in [11] for the design of decentralized equitable partitioning policies is to associate to each agent i a *virtual power generator* (virtual generator for short) (g_i, w_i) . We define the region of dominance for agent i as the Power cell $V_i = V_i(G_W)$ (see Fig. 7). We refer to the partition into regions of dominance induced by the set of virtual generators G_W as $\mathcal{V}(G_W)$. A virtual generator (g_i, w_i) is simply an artificial variable locally controlled by the i -th agent; in particular, g_i is a virtual point and w_i is its weight.

Virtual generators allow to decouple the problem of achieving an equitable partition into regions of dominance from that of positioning an agent inside its own region of dominance. We shall assume that each vehicle has sufficient information available to determine: (1) its Power cell, and (2) the locations of all outstanding events in its Power cell. A control policy that relies on information (1) and (2), is *distributed* in the sense that the behavior of each vehicle depends only on the location of the other agents with contiguous Power cells. A spatially distributed algorithm for the local computation and maintenance of Power cells can be designed following the ideas in [12].

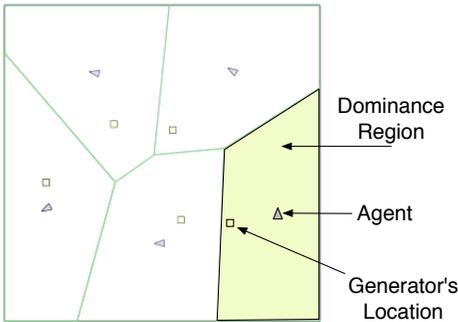


Fig. 7. Agents, virtual generators and regions of dominance.

2) *Locational Optimization*: In light of the previous result on equitable Power Diagrams, the basic idea is to enable the weights of the virtual generators to follow a (distributed) gradient descent law (while maintaining the positions of the generators *fixed*) such that an equitable partition is reached.

Assume, henceforth, that the positions of the virtual generators are *distinct*, i.e. $g_i \neq g_j$ for $i \neq j$, and belong to \mathcal{Q} . Define the set

$$S \doteq \left\{ (w_1, \dots, w_m) \in (\mathbb{R})^m \mid \lambda_{V_i} > 0 \quad \forall i \in I_m \right\}. \quad (2)$$

Set S contains all possible vectors of weights such that no region of dominance has measure *zero*.

We introduce the locational optimization function H_V :

$S \rightarrow \mathbb{R}_{>0}$:

$$H_V(W) \doteq \sum_{i=1}^m \left(\lambda \int_{V_i} \varphi(q) dq \right)^{-1} = \sum_{i=1}^m \lambda_{V_i}^{-1}.$$

Assume the generators' weight obey a first order dynamical behavior described by $\dot{w}_i = u_i$ and let $W = \{w_1, \dots, w_m\}$. Then, we set up the following control law defined over the set S

$$u_i = -\frac{\partial H_V}{\partial w_i}(W), \quad (3)$$

where we assume that the Power Diagram $\mathcal{V}(W) = \{V_1, \dots, V_m\}$ is continuously updated. One can prove that generators' weights starting at $t = 0$ at $W(0) \in S$ (this is the case, for example, if all weights are initialized to zero), and evolving under (3) remain in S and converge asymptotically to a vector of weights that yields an equitable Power diagram.

Since the gradient in Eq. (3) can be computed in a distributed way (it is possible to show that depends only on the location of the other agents with contiguous Power cells) the overall equitable partitioning policy is indeed decentralized.

3) *Optimizing the Isoperimetric Quotient*: The previous gradient descent law, although effective in providing a convex equitable partition, can yield long and “skinny” subregions. Indeed, it is possible to extend the previous algorithm to obtain convex equitable partitions while optimizing a secondary objective function. The key idea is that, to obtain an equitable partition, changing the weights, while maintaining the generators *fixed*, is sufficient. Thus, we can use the degrees of freedom given by the locations of the generators to optimize secondary cost functionals. We considered as secondary objective the maximization of the isoperimetric quotient (that measures the *regularity* of cells); such maximization is achieved by driving the power diagram toward a centroidal and equitable Voronoi diagram (see Fig. 8 for simulation results).

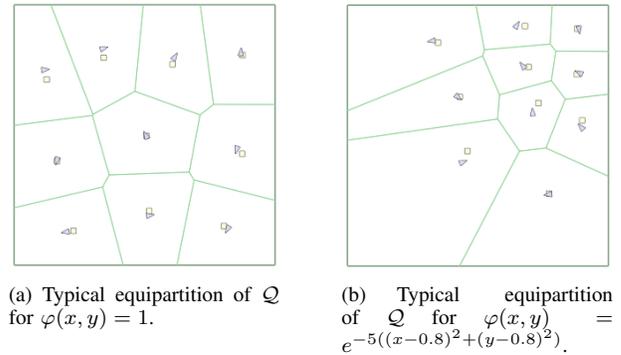


Fig. 8. Typical equipartitions achieved by using the decentralized equitable partitioning policies in [11]. Yellow dots are the virtual generators, while the gray triangles are the centroids. Notice how each bisector intersects the line segment joining the two corresponding power neighbors almost at the midpoint; hence both partitions are very close to Voronoi partitions.

Algorithm for dynamic partitions

Dynamic partitions (that are almost equitable) which have been shown to be useful for dense networks of fixed-wing UAVs in light load scenarios can be obtained using a central dispatcher. As noted before, the dynamic partition associated with an agent is fixed in its own reference frame. Therefore,

it can be completely specified by its dimensions and its offset with respect to the agent's position. The central dispatcher can compute these quantities along with the closed loitering path as shown in Figure 4. The specification of the initial location of the *loitering locations* or *virtual generators* then completes the specification of the dynamic partitioning policy.

Algorithms for Median Voronoi Tessellations

For the case of equitable partitions, we presented a decentralized equitable partitioning policy. The strategy was scalable in the sense that every virtual generator needs to talk only to a few others in order to determine its action. The success of that strategy depends on the fidelity of communication channels between various virtual generators. In fact, a common theme in cooperative control is the investigation of the effects of different communication and information sharing protocols on the system performance. Clearly, the ability to access more information at each single agent can not decrease the performance level; hence, it is commonly believed that by providing better communication among agents will improve the system's performance. In this section, we show that Median Voronoi Tessellations can be attained without any explicit communication between agents or their virtual generators; in other words, the no-communication constraint in such cases is not binding, and does not limit the steady-state performance.

In [13], we proposed a no communication based median Voronoi partitioning policy. Moreover, this policy does not assume any prior knowledge of the spatial density function $\varphi(q)$. This policy is closely related to an algorithm due to MacQueen [14], originally designed as a simple online adaptive algorithm to solve clustering problems, and later used as the method of choice in several vector quantization problems where little information about the underlying geometric structure is available. MacQueen's algorithm has the advantage to be a learning adaptive algorithm, not requiring the a priori knowledge of the distribution of the objects, but rather allowing the online generation of samples.

The behavior of the policy in the light load, i.e., for small values of λ can be summarized as follows:

- (i) At the initial time the m agents are assumed to be deployed at general positions in \mathcal{Q} , and there are no outstanding service requests.
- (ii) The agents do not move until the first service request appears. On the appearance of a service request, every agent move towards the service request location.
- (iii) As soon as one agent reaches the target, all agents start moving towards their current reference point, which is the point minimizing the average distance to targets *serviced in the past* by each agent (if there is no unique minimizer then move to the nearest one) and the process continues.

We have proven in [13] that, in the light load, almost surely, the set of reference points, $\{p_1^*(t_j), \dots, p_m^*(t_j)\}$, converges to set of generalized m -median locations of \mathcal{Q} as $j \rightarrow \infty$.

We present in Figure 9 results of numerical experiments with a non-uniform distribution, namely an isotropic normal distribution centered at $(0.25, 0.25)$, with standard deviation equal to 0.25.

It is interesting to compare the performance of this no-communication policy with a *sensor-based* policy which

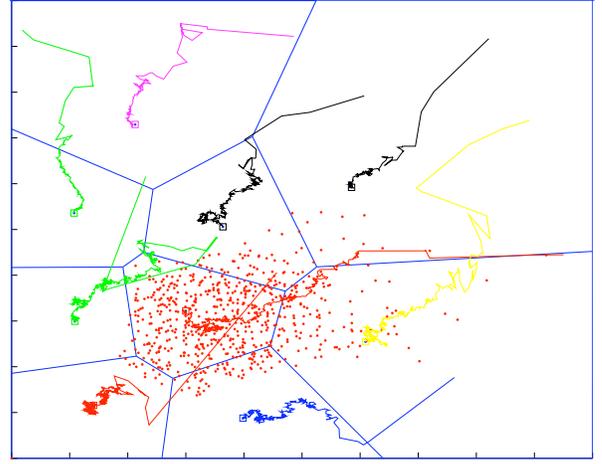


Fig. 9. Numerical simulation in the light-load case, for a normal spatial distribution. Paths followed by the reference points up to $t = 10^4$ (corresponding to approximately 5,000 targets), using the no-communication policy. The locations of all targets visited by one of the agents are also shown.

allows agents to have access to each other's positions at all times. The outcome of this additional capability in the light load case is that an agent will move towards a service request location only if it is the closest among all the agents. The sensor-based policy is more efficient than the no-communication policy in terms of the length of the path traveled by each agent, since there is no duplication of effort as several agents pursue the same target (unless a target is on the boundary of two or more Voronoi regions). However, in terms of "quality of service," we will show that there is no difference between the two policies, for low target generation rates. Numerical results show that the sensor-based policy is more efficient in a broader range of target generation rates, and in fact provides almost optimal performance both in light and heavy load conditions. However, as λ increases, the performance of the no-communication policy degrades significantly, almost approaching the performance of a single-vehicle system over an intermediate range of values of λ . Our intuition in this phenomenon is the following. As agents do not return to their own reference points between visiting successive targets, their efficiency decreases since they are no longer able to effectively separate regions of responsibility. In practice—unless they communicate and concentrate on their Voronoi region, agents are likely to duplicate efforts as they pursue the same target, and effectively behave as a single-vehicle system. Interestingly, this efficiency loss seems to decrease for large λ , and the numerical results suggest that the no-communication policy recovers a similar performance as the sensor-based policy in the heavy load limit. Unfortunately, we are not able at this time to provide a rigorous analysis of the proposed policies for general values of the target generation rate.

Another attractive feature of the no-communication based policy is that the distribution $\varphi(q)$ need not be constant: Indeed, the algorithm will provide a good approximation to a local optimum for the cost function as long as the characteristic time it takes for the target generation process to vary significantly is much greater than the relaxation time of

the algorithm. In summary, the no-communication can be seen as a learning mechanism in which the agents learn the target generation process, and the ensuing target spatial distribution, and adapt their own behavior to it.

Game-theoretic perspective for the no-communication policy

Interestingly, the no-communication based policy can be regarded as a learning algorithm in the context of the following game [13]. The service requests are considered as resources and the agents as selfish entities. The resources offer rewards in a continuous fashion and the agents can collect these rewards by traveling to the resource locations. Every resource offers reward at a rate, which depends on the number of agents present at its location: the reward rate is unity when there is one agent and it is zero when there are more than one vehicles. Moreover, the life of the resource ends as soon as more than one agents are present at its location. This setup can be understood to be an extreme form of congestion game, where the resource cannot be shared between agents and that the resource is cut off at the first attempt to share it. The total reward for agent i from a particular resource is the time difference between its arrival and the arrival of the next vehicle, if i is the first vehicle to reach the location of the resource, and zero otherwise. The utility function of agent i is then defined to be expected value of reward, where the expectation is taken over the location of the next resource. Hence, that the goal of every vehicle is to select their reference location to maximize the expected value of the reward from the next resource. In [13], we prove that the median locations, as a choice for reference positions, are an efficient pure Nash equilibrium for this game. Moreover, we prove that by maximizing their own utility function, the agents also maximize the *common global utility function*, which is the negative of the average wait time for service requests.

Algorithms for Centroidal Voronoi Tessellations

As discussed in Section III, agents performing distributed sensing tasks, with a sensing degradation that increases quadratically with the distance from the sensor, should arrange themselves so as to generate a centroidal Voronoi tessellation. If a central dispatcher is available, the main centralized algorithm, to date, to compute a centroidal Voronoi tessellation is the well-known Lloyd algorithm.

If, instead, a central dispatcher is not available, there are basically two alternatives: (i) combining distributed leader election with the Lloyd algorithm, or (ii) using the distributed version of the Lloyd algorithm presented in [9].

The algorithm [9] is conceptually similar to the algorithm in Eq. (3) for distributed equitable partitioning. If a centroidal Voronoi Tessellation is sought, then the agents should use the following update laws for generators' locations (indeed, in this case, the position of the generators should coincide with the position of the agents) and weights:

$$\begin{cases} \dot{w}_i = 0, & w_i(0) = 0, \\ \dot{g}_i = -k(g_i - C_{V_i}), & g_i(0) \in \mathcal{Q}, \end{cases}$$

where k is a positive constant, $C_{V_i} = \frac{\int_{V_i} q\varphi(q) dq}{\int_{V_i} \varphi(q) dq}$ is the *centroid* of cell V_i , and the partition $\mathcal{V}(G_W)$ is updated continuously according to the distributed control law [12].

V. CONCLUSION

In this article, we discussed the use of various spatial tessellations to determine, in the framework of partitioning policies, optimal workload share in a mobile robotic network. We also proposed efficient and distributed algorithms for achieving some of these tessellations with minimum or no communication between the agents. It is interesting to note that these tessellations appear while considering different variations of the same basic problem (DTRP). It is then natural to investigate the existence of a single objective function, whose optima correspond to the various tessellations under these different conditions. The game theory approach seems to be a promising one. We have already shown the equilibrium status of the median locations in the context of an appropriate game.

Another interesting field of study in the context of spatial tessellations is the emergent group behavior, i.e., territorial versus gregarious behavior arising out of foraging and hunting strategies of animals. The utility function approach again is a promising tool, since it provides a natural framework to study the behavior of animals as selfish agents. This line of research would be in stark contrast to the discipline of bio-robotics, where instead of taking inspiration from nature, we would try to divulge possible motives for various organisms by observing their emergent behavior.

REFERENCES

- [1] D. J. Bertsimas and G. J. van Ryzin, "Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles," *Operations Research*, vol. 41, no. 1, pp. 60–76, 1993.
- [2] M. Pavone, N. Bisnik, E. Frazzoli, and V. Isler, "A stochastic and dynamic vehicle routing problem with time windows and customer impatience," *ACM Mobile Networks and Applications*, 2008. To appear.
- [3] S. L. Smith, M. Pavone, F. Bullo, and E. Frazzoli, "Dynamic vehicle routing with heterogeneous demands," in *IEEE Conference on Decision and Control*, (Cancun, Mexico), Dec 2008.
- [4] M. Pavone, S. Smith, F. Bullo, and E. Frazzoli, "Dynamic multi-vehicle routing with multiple classes of demands," in *American Control Conference*, 2009. Submitted.
- [5] J. Enright, K. Savla, E. Frazzoli, and F. Bullo, "Dynamic routing of uav teams," tech. rep., MIT, 2008. Available for download at <http://web.mit.edu/ksavla/www/publications.html>.
- [6] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial tessellations: Concepts and Applications of Voronoi diagrams*. Wiley Series in Probability and Statistics, Chichester, UK: John Wiley & Sons, second ed., 2000.
- [7] E. Zemel, "Probabilistic analysis of geometric location problems," *Annals of Operations Research*, vol. 1, October 1984.
- [8] K. Savla and E. Frazzoli, "On endogenous reconfiguration for mobile robotic networks," in *Workshop on Algorithmic Foundations of Robotics (WAFR)*, (Guanajuato, Mexico), December 2008. To appear.
- [9] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [10] J. Buhl, D. J. Sumpter, I. D. Couzin, J. Hale, E. Despland, E. Miller, and S. J. Simpson, "From disorder to order in marching locusts," *Science*, vol. 312, pp. 1402–1406, 2006.
- [11] M. Pavone, E. Frazzoli, and F. Bullo, "Distributed policies for equitable partitioning: Theory and applications," in *IEEE Conference on Decision and Control*, (Cancun, Mexico), Dec 2008.
- [12] M. Cao and C. N. Hadjicostis, "Distributed algorithms for Voronoi diagrams and applications in ad-hoc networks," Tech. Rep. UILU-ENG-03-2222, UIUC Coordinated Science Laboratory, 2003.
- [13] A. Arsie, K. Savla, and E. Frazzoli, "Efficient routing of multiple vehicles with no communications," *IEEE Trans. on Automatic Control*, 2008. To appear.
- [14] J. Mac Queen, "Some methods for the classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Math. Stat. and Prob.* (L. M. LeCam and J. Neyman, eds.), pp. 281–297, University of California Press, 1967.