

Time Complexity of Sensor-Based Vehicle Routing

Vikrant Sharma*, Michael Savchenko*, Emilio Frazzoli[†] and Petros G. Voulgaris*

*Aerospace Engineering and Coordinated Science Laboratory

University of Illinois at Urbana-Champaign, Urbana, Illinois 61801-2935

Email: {vsharma, savchenk, vougari}@uiuc.edu

[†]Mechanical and Aerospace Engineering, University of California at Los Angeles

Los Angeles, CA 90095, USA

Email: frazzoli@ucla.edu

Abstract—In this paper, we study the following motion coordination problem: given n vehicles and n origin-destination pairs in the plane, what is the minimum time needed to transfer each vehicle from its origin to its destination, avoiding conflicts with other vehicles? The environment is free of obstacles and a conflict occurs when distance between any two vehicles is smaller than a velocity-dependent safety distance. In the case where the origin and destination points can be chosen arbitrarily, we show that the transfer takes $\Theta(\sqrt{n}\bar{L})$ time to complete, where \bar{L} is the average distance between the origin and destination points. We also analyze the case in which origin and destination points are generated randomly according to a uniform distribution, and present an algorithm providing a constructive upper bound on the time needed to transfer vehicles from origins to their corresponding destination, proving that the transfer takes $\Theta(\sqrt{n})$ time for this case.

I. INTRODUCTION

Problems involving the safe coordinated motion of several mobile agents in a shared environment are ubiquitous, appearing in many safety-critical application domains, such as surface transportation, air traffic control, and factory floors. Ground and air traffic involve ever growing numbers of individual agents; the ability of human controllers to reduce traffic congestion in major metropolitan areas and to ensure the efficient and safe operation of the national air space is approaching its limits. Moreover, as mobile robot technology progresses, unmanned and autonomously controlled ground, air and underwater vehicles will increasingly share a common environment with their human-controlled counterparts. Eventually, the design and operation of large-scale “swarms” of autonomous robots will become increasingly realistic and appealing for a variety of applications, ranging from environmental monitoring, to manufacturing, and national security. As a consequence, autonomous decision making is playing an increasing role in the development of large networks of mobile agents, and the design of algorithms for the safe and effective coordination of possibly large numbers of vehicles has attracted a great level of interest in the recent past.

While many multiple-vehicle coordination algorithms have been proposed by researchers from robotics, computer science, systems and control, and optimization, the field suffers from a lack of common language and framework. Many advances have been made using various models and assumptions, but the fundamental limits in terms of achievable performance

of the class of coordination algorithms remains unclear. As a consequence, it is difficult to characterize the true effectiveness of the algorithms available in the literature, especially when the problem involves a very large input size for which an optimal solution cannot be practically computed.

The aim of this research is to make a contribution in this direction; namely, to provide (i) a characterization of the minimum time needed to solve certain classes of motion coordination problems, and (ii) algorithms for conflict-free motion coordination, with an asymptotic analysis (as $n \rightarrow \infty$, where n is the number of vehicles) of their time complexity. We note that the expression *time complexity* is used here to denote the time needed to complete a given task by all n vehicles, and not necessarily the *algorithmic* complexity of the presented algorithm.

This work draws largely upon results from two different information technology (IT) fields: those of wireless communications and distributed computing. Perhaps surprisingly, the insight gained through the analysis of wireless networks and mesh routing, and many of the results, can be used to yield novel findings in terms of the time complexity of a class of motion coordination problem. In that sense, this research also takes a step in the direction of bridging the gap between IT and mechanical science.

The need to circumvent the intractability of the problem of determining an optimal, feasible motion plan for a multiple-robot system as the number of robots increases has necessitated compromises in the generality of the problem. Most often it is optimality that is sacrificed, such as in environments where the motion of robots is restricted to fixed paths, or roadmaps ([1], [2]), or the dynamics of the vehicle is otherwise quantized ([3]). In other work, it is the feasibility of the solution that is not guaranteed, while optimality may be reached ([4], [5]).

In this work, we focus on algorithm performance in terms of the order of the time complexity as a function of the number of agents in the system, disregarding additive and multiplicative constants. In that sense, the work is closest to that of [6], where the author develops a characterization of the *communication* complexity involved in multi-agent coordination. Our work is inspired by the research of [7] and [8] on the limits of wireless communication networks.

The paper is organized as follows. In Section II we introduce the problem of sensor-based vehicle routing, formulate the

notion of its time complexity, introduce some notation, provide some preliminary results and motivate the problem we wish to address. In Section III we derive the bounds on time complexity of the routing problem for arbitrary distribution of origin-destination pairs, showing the time complexity of the problem is $\Theta(\sqrt{n}\bar{L})$, where \bar{L} is the average distance between the origin and destination points. In Section IV, we provide upper and lower bounds on the time complexity of the vehicle routing problem in the case in which origin-destination pairs are chosen randomly from a uniform distribution, proving that the problem has time complexity $\Theta(\sqrt{n})$ with high probability. Finally, in Section V we draw some conclusions and discuss future directions.

II. PROBLEM FORMULATION

Consider a square environment \mathcal{Q} of area A . Consider n pairs of points in \mathcal{Q} , to which we will refer to as origin-destination (O, D) pairs, with $(O, D)_i \in \mathcal{Q} \times \mathcal{Q}$, $i \in \{1, \dots, n\}$. The i^{th} (O, D) pair is assigned to a mobile agent \mathcal{A}_i , $i \in \{1, \dots, n\}$. Initially each agent is *inactive*, i.e., is not considered to be in the environment, and cannot be involved in a conflict. (In other words, origin and destination points are considered as “safe havens,” in which agents are protected from traffic: these may correspond to airports and parking garages in practical applications.) Let $t_{0,i} \geq 0$ be the time at which the i^{th} agent is activated and enters the environment at location O_i ; the agent reverts to the inactive state upon arrival to its destination D_i , at time $t_{0,i} + T_i$. While active, the i^{th} agent moves within \mathcal{Q} along a continuous, time-parameterized path $\gamma_i : [0, T_i] \rightarrow \mathcal{Q}$. The position of an agent as a function of time is given by the function $x_i : t \rightarrow x_i(t) = \gamma_i(t - t_{0,i})$; by convention, we will set $x_i(t) = \gamma_i(0)$ for $t < t_{0,i}$, and $x_i(t) = \gamma_i(T_i)$ for all $t > T_i$. Finally, let $v_i(t)$ be the velocity of agent i at time t , and assume its magnitude is bounded by $v_{\max} > 0$. Assume $v_{\max} < \sqrt{A}/2$. For each active agent, we define an *exclusion zone* C , modeled as a disk centered at the agent’s position, and with radius depending affinely on the agent’s velocity, i.e.,

$$C_i(t) = \{z \in \mathbb{R}^2 : \|z - x_i(t)\| \leq r_0 + k \|v_i(t)\|\}, \quad (1)$$

for given constants $r_0 \geq 0, k > 0$. r_0 can be understood as the radius of the smallest circle enclosing each vehicle. In the paper, we will sometime identify an agent with its exclusion zone $C_i(t)$ at time t . We say that a *conflict* occurs between agents \mathcal{A}_i and \mathcal{A}_j if there exists a time t_c such that:

- Both \mathcal{A}_i and \mathcal{A}_j are active at time t_c and
- $C_i(t_c) \cap C_j(t_c) \neq \emptyset$.

A *routing policy* is a map $\pi : (O, D) \mapsto (t_0, T, \gamma)$ that, given a set of (O, D) pairs, assigns to each agent an activation schedule, and a time-parameterized path. A routing policy π is *safe* if it generates no conflicts. We define the Sensor-Based Vehicle Routing Problem (SBVRP) as the problem of finding a safe routing policy, given the (O, D) pairs. Let us indicate with $T_\pi(O, D)$ the time at which the last agent is deactivated according to policy π ; we will define the time complexity of SBVRP for the (O, D) pairs as the infimum of this time over

all possible safe routing policies π , i.e.,

$$T^*(O, D) = \inf_{\pi \text{ safe}} T_\pi(O, D).$$

We write $f(n) = O(g(n))$ if there exists a positive constant c_1 such that $f(n) \leq c_1 g(n)$ for all n large enough. Similarly, we write $f(n) = \Omega(g(n))$ if there exists a positive constant c_2 such that $f(n) \geq c_2 g(n)$ for all n large enough. We say that $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

A. Preliminary results

We have the following trivial bound, showing that our problem formulation is such that a solution always exists (hence making the feasibility decision problem trivial, unlike the generalized movers’ and warehouseman’s problems), and providing a conservative upper bound on the time complexity.

Proposition 2.1: For any set of n origin-destination pairs, the SBVRP is feasible, and its time complexity is $O(n)$.

Proof: Assume agents are labeled by their activation time, i.e., $t_{0,i} \leq t_{0,i+1}$. If agents are activated sequentially, i.e., agent \mathcal{A}_{i+1} is activated upon deactivation of agent \mathcal{A}_i , no conflicts can arise. Each agent moves with the maximum velocity along the line joining its origin and destination and hence the time needed for the i -th agent to reach its destination can be bounded as $T_i \leq \text{diam } \mathcal{Q}/v_{\max}$. Hence, the time at which the last agent arrives at its destination is $T_{\text{seq}}(O, D) = \sum_{i=1}^n T_i \leq n \frac{\text{diam } \mathcal{Q}}{v_{\max}}$, which proves the claim. \square

Let us consider now the case in which $r_0 > 0$, i.e., the area of the exclusion regions is bounded below by a positive constant. We have the following.

Theorem 2.1: If $r_0 > 0$ in (1), for any set of n origin-destination pairs, such that $\min_i |O_i - D_i| \geq l > 0$, the time complexity of the SBVRP is $\Theta(n)$.

Proof. Each agent has to stay inside the environment and since the disk of radius r_0 is the minimum radius disk covering an agent, the center of this disk also lies inside the environment at any time. Since the environment \mathcal{Q} is chosen as a square, each active agent claims exclusive ownership of a region in \mathcal{Q} of area at least $\pi r_0^2/4$, if $r_0 \leq \sqrt{A}/2$; in such a case, at most $n_a = \lfloor 4A/(\pi r_0^2) \rfloor$ agents can be active at the same time. If $r_0 > \sqrt{A}/2$, then at most one agent can be active at any given time, i.e., $n_a = 1$. At most n_a new agents can be activated no sooner than every l/v_{\max} time units, i.e., the minimum time needed for at least one agent to reach its destination. Hence the time needed to activate and transfer all agents is at least $\lceil n/n_a \rceil \cdot l/v_{\max}$. Therefore, the total time needed to complete the transfer of all agents can be bounded as

$$T_{r_0 > 0} \geq n \frac{\pi r_0^2 l}{4A v_{\max}},$$

if $r_0 \leq \sqrt{A}/2$, and $T_{r_0 > 0} \geq nl/v_{\max}$ otherwise. The lower bounds, together with the upper bound in Proposition 2.1, prove the claim. \square

This result condemns sensor-based vehicle routing problems with agents of non-zero size to linear time complexity, that is, no better than a sequential agent activation, when n becomes large. In the remainder of this work, we will study the case in

which $r_0 = 0$ in the definition of the no-conflict constraints, i.e., the case in which the radius of the exclusion region is directly proportional to the agent's velocity and each agent occupies a point in the environment.

While this is not—strictly speaking—a physically realistic modeling assumption, one must keep in mind that, in most problems of interest, conflicts are generated when vehicles get closer than some safety distance that is much bigger than the physical dimensions of the vehicle; for example, in air traffic control, a conflict is generated whenever two aircraft get within 5 nautical miles from each other. In automotive traffic, defensive drivers typically maintain a safety distance from a leading car equivalent at least to the distance traveled in 2 seconds, which at 50 km/h (about 30 mph) amounts to about 30 meters (about 90 ft), i.e., several times the length of a standard car. Even pedestrian traffic follows a similar rule, as people generally regulate their walking speed to maintain a comfortable distance from others when moving through crowded areas.

In addition, setting $r_0 = 0$ lets us study the effect of velocity on traffic congestion, and provides insight into the nature of cooperation between agents aiming at minimizing the overall transfer time. The intuition is that as agents move faster, they need a bigger buffer to avoid collisions with others, hence reclaiming a larger portion of a shared resource (the environment), and thereby imposing severe constraints on the motion of other agents.

III. ARBITRARY (O, D) PAIRS

In this section we will compute the lower and upper bounds on the time complexity of SBVRP, where the origin-destination pairs are chosen arbitrarily. Note that “arbitrarily” here must be understood as “in such a way as to minimize time complexity.” In other words, while the lower bound by definition applies to all network configurations, the upper bound is meant to apply to transportation networks designed for efficiency (rather than an adversarial worst-case).

In talking about arbitrary networks, we note that the distance between origin and destination for each agent is an arbitrary variable that directly affects the time complexity. It can be made arbitrarily small, hence making the completion time of the algorithm likewise arbitrarily small. A relevant measure in the arbitrary case is therefore not completion time, but time per average distance traveled by the agents. This is equivalent to including the distance variable in the bound on time complexity.

A. A lower bound on the time complexity

In the arbitrary case, we have the following lower bound.

Lemma 3.1: For any set of n (O, D) pairs, such that the average distance between origin and destination points is \bar{L} , the time complexity of the SBVRP is $\Omega(\sqrt{n\bar{L}})$.

Proof. Let us assume that the motion of all the agents can be represented as a set of straight-line motions, over a common, synchronized time schedule of length h . For simplicity, let us assume that each time interval has the common duration τ .

By definition, all agents reach their destination within time $T^* = h\tau$. Let us denote by r_i^j the length of the straight-line segment along which the i -th agent moves during the j -th time interval. Obviously, we have $\sum_{j=1}^h r_i^j \geq L_i$, where $L_i = |O_i - D_i|$, and

$$\sum_{i=1}^n \sum_{j=1}^h r_i^j \geq n\bar{L}, \quad (2)$$

where $\bar{L} = \frac{1}{n} \sum_{i=1}^n L_i$.

Assuming the velocity of any agent i during a time interval j stays constant and is denoted by v_i^j , we define $\delta_i^j = kv_i^j = kr_i^j/\tau$. Hence the exclusion region area at any time during interval j for agent i is given by

$$A_i^j = \pi \left(\delta_i^j \right)^2 = \pi k^2 \left(\frac{r_i^j}{\tau} \right)^2$$

The collision avoidance constraint requires that at any time during a time interval, the sum of the areas of the exclusion regions of all agents lying inside the environment \mathcal{Q} cannot exceed the area of the environment. Since the environment chosen is a square, this means that at least one fourth of each exclusion region is within \mathcal{Q} . Hence, the sum of the areas of the exclusion regions of all agents at any time during any time interval cannot exceed four times the area of the environment, i.e.,

$$\frac{\pi k^2}{\tau^2} \sum_{i=1}^n \left(r_i^j \right)^2 = \sum_{i=1}^n A_i^j \leq 4A.$$

Summing over all intervals in the time schedule, and rearranging, we get

$$\sum_{i=1}^n \sum_{j=1}^h \left(r_i^j \right)^2 \leq \frac{4Ah\tau^2}{\pi k^2}. \quad (3)$$

Consider a convex function $f : \mathfrak{R} \rightarrow \mathfrak{R}$; Jensen's inequality states that

$$f \left(\frac{1}{P} \sum_{p=1}^P x_p \right) \leq \frac{1}{P} \sum_{p=1}^P f(x_p).$$

Since the function $x \mapsto x^2$ is convex, we can apply Jensen's inequality to (3) to obtain

$$\left(\sum_{i=1}^n \sum_{j=1}^h r_i^j \right)^2 \leq hn \sum_{i=1}^n \sum_{j=1}^h \left(r_i^j \right)^2 \leq \frac{4Ah^2\tau^2n}{\pi k^2},$$

that is,

$$\sum_{i=1}^n \sum_{j=1}^h r_i^j \leq \left(\frac{4(T^*)^2 An}{\pi k^2} \right)^{\frac{1}{2}}. \quad (4)$$

Thus, from (2) and (4) we get

$$T^* \geq \left(\frac{\pi k^2}{4A} \right)^{\frac{1}{2}} \sqrt{n\bar{L}}.$$

Since we have made no assumptions on the time schedule, this bound applies in the limit as $\tau \rightarrow 0$, i.e., for continuous schedules, which proves the result. \square

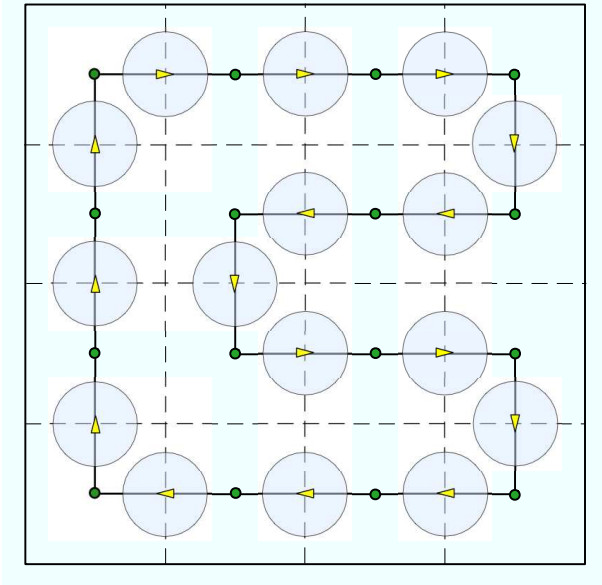


Fig. 1. Illustration of the procedure used in the proof of Lemma 3.2 to select origin-destination pairs that achieve the claimed upper bound on time complexity. In this case, $n = 16$.

B. A constructive upper bound on the time complexity

In this section, we demonstrate that there exists a choice of (O, D) pairs that achieves the same time complexity that appears in the lower bound in Lemma 3.1.

Lemma 3.2: For any $n \in \mathbb{N}$ there exist sets of n (O, D) pairs, such that the time complexity of the SBVRP is $O(\bar{L}\sqrt{n})$, where \bar{L} indicates the average distance between origins and the corresponding destinations.

Proof. We will prove the statement by presenting a procedure for the selection of (O, D) pairs that admits a safe routing policy of time complexity proportional to $\bar{L}\sqrt{n}$.

Given $n \in \mathbb{N}$, partition \mathcal{Q} into p^2 openly disjoint identical squares, where we define $p = \lceil \sqrt{n} \rceil$ if $\lceil \sqrt{n} \rceil$ is even otherwise $p = \lceil \sqrt{n} \rceil + 1$. Here $\lceil x \rceil$ is the smallest integer larger than or equal to x . The side length of each such square is then \sqrt{A}/p . Consider a minimum-length tour Γ through the centers of such squares as shown in Figure 1; the length of Γ is equal to $p\sqrt{A}$. Notice such a tour can always be constructed since p is even.

Pick an arbitrary direction on Γ , and assign to each agent a unique square center as its origin, and the next square center on Γ as its destination. The distance between all origin and destination points is the same, and equal to $\bar{L} = \sqrt{A}/p$.

All agents can be activated at the same time, and can travel along Γ at speed equal to

$$v = \frac{\eta_1}{k} \frac{\sqrt{A}}{2\sqrt{2}p},$$

where $\eta_1 < 1$ can be chosen arbitrarily close to one. This ensures that no two agents collide during their journey. The total time taken for transfer of all agents is then given by

$$T^* = \bar{L}/v = \frac{2\sqrt{2}k}{\eta_1\sqrt{A}} p\bar{L} \leq \frac{2\sqrt{2}k}{\eta_1\sqrt{A}} (\sqrt{n} + 2)\bar{L}. \quad \square$$

Let $\mathcal{OD}(n)$ be the set of all n (O, D) pairs lying in \mathcal{Q} and $\bar{L}_{(O,D)}$ be the average distance between the origins and their corresponding destinations for a given (O, D) pair. The combination of Lemmas 3.1 and 3.2 proves our first main result:

Theorem 3.1: For any $(O, D) \in \mathcal{OD}(n)$, the time complexity of the SBVRP is $\Omega(\sqrt{n}\bar{L}_{(O,D)})$ and there exists $(O, D) \in \mathcal{OD}(n)$ such that the time complexity of the SBVRP for this pair is $O(\sqrt{n}\bar{L}_{(O,D)})$.

Hence the time-average agent velocity toward goal is of order $\Theta(1/\sqrt{n})$. Multiplying by n , we see that the agent-meters per second throughput metric of the network as a whole is therefore $\Theta(\sqrt{n})$, which corresponds to analogous bounds of wireless network capacity in arbitrary networks, as in [7].

IV. RANDOM (O, D) PAIRS

In this section, we will consider the case in which (O, D) pairs are generated randomly, i.e., origin and destination points are sampled from a uniform distribution in \mathcal{Q} , identically and independently. Throughout the paper, we will use the phrase “with high probability” abbreviated as *whp* to stand for “with probability approaching 1 as $n \rightarrow \infty$.”

A. A lower bound on the time complexity

Lemma 4.1: The time complexity of the SBVRP for a set of n origin-destination pairs, randomly sampled from a uniform distribution in \mathcal{Q} , is $\Omega(\sqrt{n})$, *whp*.

Proof. Denote by L_i the distance between O_i and D_i , $i = 1, \dots, n$. Since origin and destination points are independently and identically distributed (i.i.d.), the distances L_i are i.i.d. random variables. The expected distance between two points sampled from a compact set is bounded. In particular, the expected distance between two uniformly distributed random points in a square of area A can be computed as

$$\mathbb{E}[L_i] = \frac{2 + \sqrt{2} + 5 \ln(1 + \sqrt{2})}{15} \sqrt{A} \approx 0.521\sqrt{A} \quad (5)$$

Similar arguments can be made to show that the variance of the distance is bounded. The weak law of large numbers [9] ensures that the average distance \bar{L} between origin and destination points converges *whp* to the common mean—given in (5) in the case of a square environment—as n increases. In other words for any $\epsilon > 0$,

$$\lim_{n \rightarrow \infty} \text{Prob} \left[\left| \frac{1}{n} \sum_{i=1}^n L_i - \mathbb{E}[L_i] \right| \geq \epsilon \right] = 0.$$

Using Lemma 3.1, we can conclude that

$$T^* \geq \frac{k}{2} \mathbb{E}[L_i] \sqrt{\frac{\pi n}{A}}, \quad \text{whp}$$

In other words, $T^* \geq 0.462k\sqrt{n}$, *whp*. \square

B. An upper bound on the time complexity

In this subsection, we present an algorithm to compute a safe routing policy for the case of random (O, D) pairs, such that it takes $O(\sqrt{n})$ time, *whp*, to route all agents from their origins to their destinations. The policy can be split into three stages of vehicle routing, namely: i) Initialization phase, ii) Main phase and iii) Termination phase, in this order. We show that the initialization and final phases can be executed in $O((\log n)^{3/2})$ time *whp*, and the main phase can be finished in $O(\sqrt{n})$ *whp*. Hence the overall algorithm terminates in $O(\sqrt{n})$ *whp*.

Our algorithm relies on the definition of the following square tilings of \mathcal{Q} :

- A *coarse* tiling $\mathcal{P}_c(n)$, obtained by partitioning the environment into $\left\lceil \sqrt{\frac{n}{K \log n}} \right\rceil^2$ squares of equal area (coarse cells), where $K > \frac{1}{\log(4/e)}$ (e is the base of natural logarithms). Let

$$l_c(n) = \frac{\sqrt{A}}{\left\lceil \sqrt{\frac{n}{K \log n}} \right\rceil}$$

be the side length of such squares. Let $\mu^* \in (0, 1)$ be the sole root of the equation

$$-\mu^* + (1 + \mu^*) \log(1 + \mu^*) = \frac{1}{K}$$

- A *medium* tiling $\mathcal{P}_m(n)$, obtained by partitioning each cell in \mathcal{P}_c into $\left\lceil \sqrt{2(\mu + 1)K \log n} \right\rceil^2$ squares of equal area (medium cells), with $\mu > \mu^*$. Let

$$l_m(n) = \frac{\sqrt{A}}{\left\lceil \sqrt{\frac{n}{K \log n}} \right\rceil \left\lceil \sqrt{2(\mu + 1)K \log n} \right\rceil}$$

be the side length of such squares.

- A *fine* tiling $\mathcal{P}_f(n)$, obtained by partitioning each cell in \mathcal{P}_m into nine squares of equal area (fine cells). Let $l_f(n) = l_m(n)/3$ be the side length of such squares.

As explained in the following, the coarse tiling will be used in the initialization phase, to identify regions in the environment in which agents can be activated and moved independently. The medium tiling will provide the main structure for the actual routing of agents, through the solution of a specialized version of the permutation routing problem in 2-D meshes. The fine tiling will ensure the existence of a buffer to safely accommodate agents temporarily sharing a medium cell.

1) *Initialization and Termination phase*: In the initialization phase, all agents are activated, and moved in such a way that over-crowding is avoided; more specifically, at the end of the initialization phase, agents will be placed at rest at the center of cells in the medium tiling, with at most one agent occupying each cell. In the termination phase, the opposite process is implemented, in the sense that agents are moved from the centers of cells in the medium tiling to their destination, and deactivated. In order to minimize the execution time of the

initialization and termination phases, a large number of agents are activated at the same time; the absence of conflicts in these phases is achieved by concurrently activating agents that are sufficiently far from one another.

The following result stated in Lemma 3.1 in [10] is critical in proving some of the results in this section.

Lemma 4.2: Each coarse cell in \mathcal{P}_c contains no more than $(\mu + 1)K \log n$ origins (destinations), *whp*.

Since each coarse cell $p_c \in \mathcal{P}_c$ contains at least $2(\mu + 1)K \log n$ medium cells, an immediate consequence of Lemma 4.2 is that in each coarse cell there are at least $(\mu + 1)K \log n$ medium cells that do not contain origin points, *whp*. (The same can be said about destination points.) Therefore, it is possible to associate to each agent \mathcal{A}_i , $i = 1, \dots, n$, two intermediate way-points, O'_i and D'_i , with the following properties:

- P1)** If O_i (resp., D_i) is in the coarse cell p_c , then O'_i (resp., D'_i) is in the same coarse cell.
- P2)** Each cell in the medium tiling contains at most one of the points in $\{O'_i : i \in 1, \dots, n\}$, and at most one of the points in $\{D'_i : i \in 1, \dots, n\}$.
- P3)** If a medium cell contains a point in $\{O'_i : i \in 1, \dots, n\}$ (resp., $\{D'_i : i \in 1, \dots, n\}$), then it does not contain any point in $\{O_i : i \in 1, \dots, n\}$ (resp., $\{D_i : i \in 1, \dots, n\}$).
- P4)** All intermediate way-points $\{O'_i, D'_i : i = 1, \dots, n\}$ are at the center of a cell in the medium tiling. Note that since each medium cell contains nine fine cells, the center of a medium cell also coincides with the center of a cell in the fine partition.

At the end of the initialization phase, all agents will be active, and at rest at the respective first intermediate way-point O' . Similarly, at the start of the termination phase, all agents will be at rest at their second intermediate way-point D' . Next we wish to show that it is possible to move all agents in such a configuration in a safe manner, within time $O((\log n)^{3/2})$.

Lemma 4.3: Let O be a set of n origin points sampled at random from a uniform distribution, and O' a set of intermediate way-points, satisfying properties P1-4. Then, n agents can be activated and moved from points in O to points in O' within time $O((\log n)^{3/2})$, *whp*.

Proof. We will prove the result by providing an algorithm that solves the problem with a cost satisfying the stated bound. The algorithm that we propose for the initialization phase, called SPREADOUT, requires the solution of a coloring problem on the tiling \mathcal{P}_c , in such a way that no neighboring cells have the same color (including cells touching at one point only, i.e., diagonally). It is straightforward to see that four colors would suffice to color such a square tiling on a plane in this manner (e.g., see [11] to see standard coloring problems).

Given the points O_i and O'_i , and a coloring scheme, the SPREADOUT algorithm works as follows (see Figure 2 for a pictorial representation). Choose a color c and, concurrently in all coarse cells with the same color, activate agents in turn, moving them to their first intermediate way-point. (If an origin point lies at the boundary between two or more coarse cells, assign the corresponding agent to one of the cells arbitrarily.)

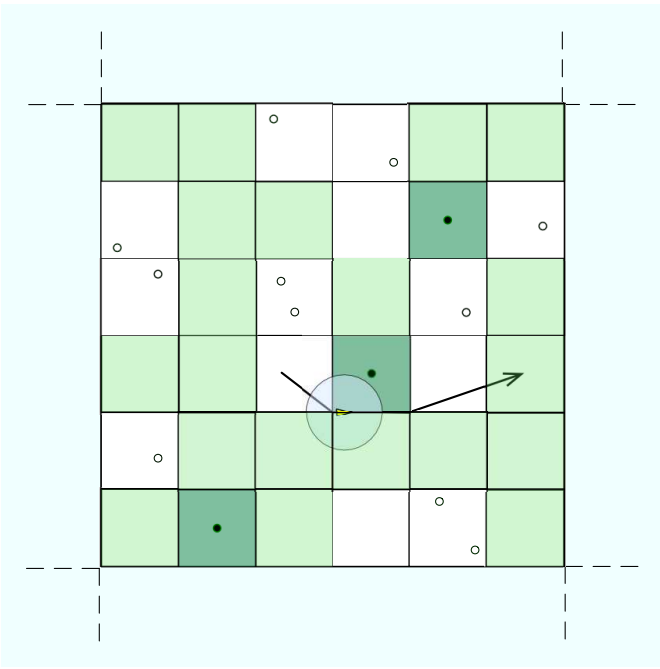


Fig. 2. Zoomed-in sketch of the algorithm used in the initialization phase. The figure shows a typical agent configuration within a coarse cell that is divided into medium cells. Empty dots represent the location (origin) of inactive agents. Filled dots represent the location of active agents, at rest at their first intermediate way-point which is the center of an exclusive medium cell. Lightly-shaded cells do not contain origin points, whereas darkly shaded cells contain an active agent at their centers. The thick curve represents the path γ_i^{init} of the agent \mathcal{A}_i , that has been recently activated. The point where the curve starts is O_i and the point where the arrow ends is O'_i . Location of the agent and its exclusion region are also shown at a time instant during its transfer from O_i to O'_i .

The path γ_i^{init} followed by agent \mathcal{A}_i is chosen as the shortest continuous curve joining O_i to O'_i , without intersecting the interior of medium cells already containing an active agent. Agents travel along their paths at speed $v^{\text{init}} = \eta_2 l_m / 2k$, where $\eta_2 < 1$ can be chosen arbitrarily close to one. Once all agents with origin point in cells with color c are active and at rest at their first intermediate way-point, choose a different color, and repeat until all agents have been activated.

Because of property P1, the length of the paths γ^{init} is bounded by $2l_c$: as a consequence, each agent reaches its first intermediate way-point within time $\frac{4k}{\eta_2} \left[\sqrt{2(\mu+1)K \log n} \right]$ after it is activated. Since there are at most $(\mu+1)K \log n$ agents per coarse cell *whp* by Lemma 4.2, activation and transfer of all agents in a coarse cell will be completed within time $\frac{4k}{\eta_2} (\mu+1)K \log n \left[\sqrt{2(\mu+1)K \log n} \right]$, *whp*. Since all coarse cells with the same colors are processed concurrently, and there are four colors, the execution time of SPREADOUT is bounded by $\frac{16\sqrt{2}k}{\eta_2} ((\mu+1)K \log n)^{3/2} + \frac{16k}{\eta_2} (\mu+1)K \log n$, *whp*.

Now we need to show that the SPREADOUT algorithm is safe, i.e., it does not generate conflicts. The exclusion region of agents moving according to the SPREADOUT algorithm has radius $\delta^{\text{init}} = kv^{\text{init}} = \eta_2 l_m / 2$. The SPREADOUT algorithm

moves at most one agent at a time in each coarse cell with a given color. Since neighboring coarse cells have different colors, active and moving agents are at least a distance $l_c > 2\delta^{\text{init}} = \eta_2 l_m$ away from each other. As a consequence, no conflicts can arise between agents in motion. Finally, active agents are either moving or at rest at the center of a medium cell (because of property P4); by construction, the paths γ^{init} are greater than δ^{init} distance away from the centers of cells occupied by active agents at rest, thus avoiding conflicts between agents in motion and agents at rest. \square

The termination phase starts with all agents active and at rest at their second intermediate way-point D' . During this phase, all agents are moved to their destination and deactivated, without conflicts. It can be recognized that a minor modification to the SPREADOUT algorithm (wherein the γ^{term} paths join the D' and D points) will achieve this goal, with the same cost of the initialization phase:

Lemma 4.4: Let D be a set of n destination points sampled at random from a uniform distribution, and D' a set of intermediate way-points, satisfying properties P1-4. Then, n agents can be moved from points in D' to points in D and deactivated within time $O((\log n)^{3/2})$, *whp*.

2) *Main Phase:* As a prelude to describing the algorithm for the main phase, we review a related problem studied in the parallel and distributed computing research community. Consider a square of m^2 processing units (PUs) with m PUs in each row and column. Each PU is connected to its vertical and horizontal neighbors with a communication link. Each processing unit can send and receive one packet along each communication link in a time slot. (Thus every PU can receive and transmit at most four packets simultaneously in a time slot.) In addition, each PU is able to store in a buffer a queue of packets waiting to be transmitted.

Suppose each PU is the source and destination of exactly one packet. The problem of routing the total m^2 packets to their destinations is a well studied problem in the parallel and distributed computing literature under the name of *permutation routing*. The following result characterizes the performance of permutation routing algorithms with minimal queue length requirements at the PUs.

Theorem 4.1: [12] Permutation routing in a $m \times m$ mesh can be performed deterministically in $N_{\text{PR}} = 2m - 1$ time steps and with maximum queue size equal to 2.

An algorithm achieving this performance characteristics is given in [12]. We will refer to such an algorithm as Permutation Routing (PERMROUTING) algorithm. In our proposed solution to SBVRP problems, at the start of the main phase, all agents are active and at rest at the intermediate way-points $\{O'_i : i = 1, \dots, n\}$; at the start of the termination phase, all agents must be active and at rest at the intermediate way-points $\{D'_i : i = 1, \dots, n\}$. Consider the medium partition $\mathcal{P}_m(n)$. Because of property P2, each cell in this partition contains at most one of the points in O' and at most one of the points in D' . Hence, the problem of routing agents from the respective first intermediate way-point to the second one can be cast as a permutation routing problem, in which medium cells play the

role of processing units. In mesh routing, the PERMRouting algorithm associates to each packet and time step a processing unit; in our case, the PERMRouting algorithm outputs a map $S : \{1, \dots, n\} \times \{1, \dots, N_{PR} + 1\} \rightarrow \mathcal{P}_m$, associating to each agent and time step a cell in the medium partition. We will use the shorthand $S_i^q = S(i, q)$ to indicate the cell assigned to the i -th agent at the beginning of time step q . Naturally, $O_i' \in S_i^1$, and $D_i' \in S_i^{N_{PR}+1}$, for all $i = 1, \dots, n$. If $S_i^q = S_i^{q+1}$ the i -th agent is held in a queue at time step q , i.e., it must wait before being transferred to the next cell on its path.

A difference between our routing problem and the one considered in [12] is that in the case of vehicle routing “communication links” are not full duplex, i.e., it is not possible to travel at the same time in both directions across a single link. We obviate this problem by splitting each time slot into four rounds. In each round, only cell transfers in a specified direction (e.g., North, South, East, or West) are allowed, in turn. Let $\text{Neigh} : \mathcal{P}_m \times \{\text{North, South, East, West}\} \rightarrow \mathcal{P}_m \cup \text{null}$ be a map that associates to each medium cell its neighbor in a given direction (if it exists, null otherwise).

At most four agents will be transferred out of a medium cell during each time slot, one per round. The following result will be useful in the analysis of the application of the PERMRouting algorithm to Sensor-Based Vehicle Routing:

Proposition 4.1: The maximum number of agents in any medium cell $p_m \in \mathcal{P}_m$ at the start of a routing time slot is 6, i.e.,

$$\max_{p_m \in \mathcal{P}_m} \left\{ \max_{q \in \{1, \dots, N_{PR} + 1\}} \text{card}(\{i \in \{1, \dots, n\} : S_i^q = p_m\}) \right\} \leq 6$$

Proof. We prove the claim by contradiction. Assume there exists an integer q and a cell $p_m \in \mathcal{P}_m$ such that $\text{card}(\{i \in \{1, \dots, n\} | S_i^q = p_m\}) > 6$. Since during the time slot q at most 4 agents can leave the cell p_m , this implies that cell p_m will have to buffer more than two agents during the q -th time slot, which is a contradiction by Theorem 4.1. \square

We are now ready to explain our algorithm for the main phase, to which we will refer as MAINROUTING. The first step consists of solving the permutation routing problem with the PERMRouting algorithm. For each time slot q in the routing schedule, perform the following. In each round of the time slot, pick one direction $dir \in \{\text{North, South, East, West}\}$. Concurrently, for each cells $p_m \in \mathcal{P}_m$, pick (if it exists) the unique agent \mathcal{A}_i such that $S_i^q = p_m$, and $S_i^{q+1} = \text{Neigh}(S_i^q, dir)$. If $D_i' \notin S_i^{q+1}$, then move the agent to the center of an empty fine cell inside S_i^{q+1} , other than the middle one, and put it to rest. Proposition 4.1 and the fact that at most one agent enters a medium cell every round and at most one leaves every round ensure that at most 7 agents are inside a medium cell at the start of a round if there is an agent to be transferred to this cell during the round (notice there are only 4 rounds). This ensures that an empty fine cell is always available for the agent to be transferred to. Otherwise, if $D_i' \in S_i^{q+1}$, move the i -th agent to its second intermediate way-point D_i' , which is the center of the middle fine cell in

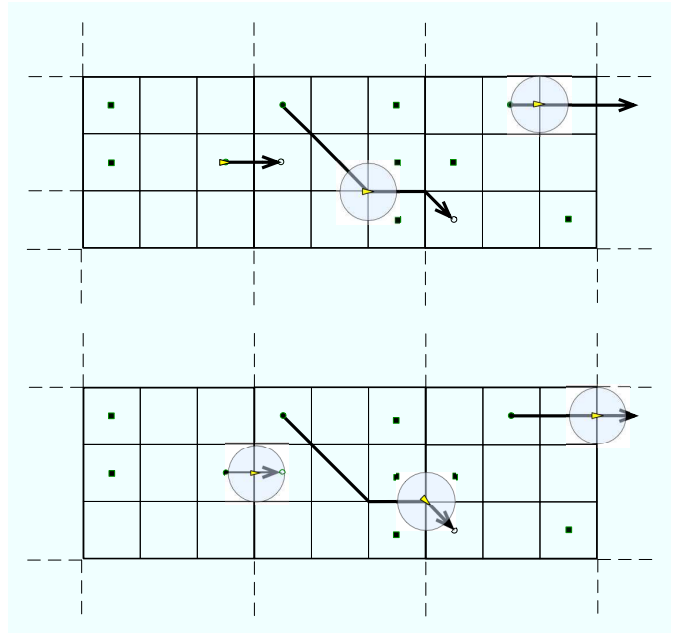


Fig. 3. The figure at the top shows the transfer of vehicles in a round in three consecutive cells in a row of the medium partition. The figure at the bottom shows the vehicle positions halfway through the round. The dots are the vehicles at rest at the center of an exclusive fine cell and the moving vehicles are depicted by small arrows with the circular exclusion zones around them.

S_i^{q+1} , and put it to rest. Repeat the above for all four directions, one per round.

Next we characterize the time needed to execute one single round of the MAINROUTING algorithm safely:

Lemma 4.5: Each round of the MAINROUTING algorithm can be performed with no conflicts in time $T_{\text{round}} \leq 24k/\eta_3$, where $\eta_3 < 1$ is an arbitrary constant chosen close to one.

Proof. Let q be the current routing time slot; because of the symmetry of the problem, we can restrict our analysis to the case in which $dir = \text{East}$. Let I_{East} be the set of indices of agents moving East in the current round. For all $i \in I_{\text{East}}$, choose an empty destination fine cell $p_f \in S_i^{q+1}$ and define $\gamma_i^q : [0, T_{\text{round}}] \rightarrow \mathcal{Q}$ as the shortest continuous time-parameterized trajectory satisfying the following properties:

- Q1) $\gamma_i^q(0)$ is the location of the agent at the beginning of the q -th routing time slot.
- Q2) $\gamma_i^q(T_{\text{round}}/2)$ lies on the boundary between S_i^q and S_i^{q+1} .
- Q3) $\gamma_i^q(T_{\text{round}})$ is the center of the destination fine cell for the i -th agent at the q -th routing time step.
- Q4) All points in γ_i^q are at least at a distance $l_f/2$ from the boundary of $S_i^q \cup S_i^{q+1}$.
- Q5) No point in γ_i^q lies in the interior of a fine cell containing any other agent at rest at its center.

Such a trajectory can always be designed, see Figure 3 for an example. The length of the the segment of γ_i^q within S_i^q (and S_i^{q+1}) is bounded by $2l_m$; agents can move along these paths at speed $\eta_3 l_f/2k$, where $\eta_3 < 1$ can be chosen arbitrarily close to one. Hence, the exclusion zones of each moving agent has radius $< l_f/2$ throughout the motion. This and property Q5

ensure moving agents do not collide with agents at rest. Since it would take at most $12k/\eta_3$ time for any agent to reach the boundary it is crossing and at most the same time from there to the center of its destination cell, choosing $T_{\text{round}} = 24k/\eta_3$ ensures that property Q2 and Q3 are satisfied. Property Q2 ensures that the agent maintains separation between agents being transferred from the left and right neighboring medium cells and does not collide with them. Property Q4 ensures that the agent maintains separation and does not collide with the moving agents in neighboring medium cells in the above and below row of medium cells (recall that $l_m = 3l_f$). \square

Based on this intermediate result, we can conclude that:

Lemma 4.6: Consider n pairs of intermediate way-points (O', D') satisfying properties P1-4. Then, all agents at rest at their respective intermediate first way-point in O' can be routed to their corresponding second way-point in D' in $O(\sqrt{n})$ time.

Proof. According to theorem 4.1, permutation routing can be performed in $2m - 1$ steps, where in our case m is the number of cells in a row in the medium partition \mathcal{P}_m , i.e., $m = \sqrt{A}/l_m = \left\lceil \sqrt{\frac{n}{K \log n}} \right\rceil \left\lceil \sqrt{2(\mu + 1)K \log n} \right\rceil$. Since each routing step is composed of four rounds, one for each direction of motion, and each round takes at most $24k/\eta_3$ time, the total time taken for routing agents from their respective first way-points to their respective second way-points is at most $\frac{96k}{\eta_3} (2(\sqrt{\frac{n}{K \log n}} + 1)(\sqrt{2(\mu + 1)K \log n} + 1) - 1)$. This proves the result. \square

Lemmas 4.3, 4.4 and 4.6 together give the following second main result of this paper:

Theorem 4.2: For any set of n (O, D) pairs randomly chosen from a uniform distribution in \mathcal{Q} , the time complexity of the SBVRP is $\Theta(\sqrt{n})$ whp

V. CONCLUSION

In this paper, we have studied the time complexity of sensor-based vehicle routing problem where conflict is defined by the intersection of velocity-dependent exclusion regions. We first showed that if the area of the exclusion region is bounded away from zero, the time complexity of the routing problem is $\Theta(n)$, i.e., is no better than the trivial worst-case bound. We then focused on the case in which the exclusion region can be made arbitrarily small by reducing the agent's velocity, and showed that for the case in which origin and destination pairs can be chosen arbitrarily, the time complexity of the vehicle routing problem is $\Theta(\sqrt{n\bar{L}})$, where \bar{L} is the average distance between the origin and destination points. In the case of random origin-destination pairs, we showed that the time complexity is $\Theta(\sqrt{n})$. In the future, we intend to present a worst case vehicle routing situation and characterize

its time complexity. We also intend to introduce formation mobility model and show how this can be used to improve the time complexity of the vehicle routing problem. Another direction of research is to tie this work back to its origins in the wireless communications literature. There has been recent work ([13],[14]) demonstrating that node mobility improves the theoretical capacity of wireless networks, established in [7]. By coupling the capacity problem with a realistic physical model of nodes as mobile agents, we can determine the true fundamental limits of communication. This new model would reflect the constraints imposed by both wireless medium access and the shared resource of physical space.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under ITR grant 0325716, NSF grant 0133869 and AFOSR grant F49620-02-1-0325. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the supporting organizations.

REFERENCES

- [1] S. M. Lavelle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, 1998.
- [2] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," in *Proceedings of Fifth International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Nice, France, 2002.
- [3] E. Frazzoli, Z. H. Mao, J. H. Oh, and E. Feron, "Aircraft conflict resolution via semi-definite programming," *AIAA Journal of Guidance, Control and Dynamics*, vol. 24, no. 1, pp. 79–86, 2001.
- [4] G. Inalhan, D. M. Stipanovic, and C. J. Tomlin, "Decentralized optimization, with application to multiple aircraft coordination," in *Proceedings of IEEE Conference on Decision and Control*, 2002.
- [5] B. P. Gerkey and M. J. Mataric, "Sold!: Auction methods for multi-robot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.
- [6] E. Klavins, "Communication complexity of multi-robot systems," in *Proceedings of Fifth International Workshop on the Algorithmic Foundations of Robotics*, Nice, France, 2002.
- [7] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 388–404, 2000.
- [8] S. R. Kulkarni and P. Vishwanath, "A deterministic approach to throughput scaling in wireless networks," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1041–1049, June 2004.
- [9] R. Ash, *Basic Probability Theory*. John Wiley and Sons Inc, 1970.
- [10] F. Xue and P. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, March 2004.
- [11] D. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, 2000.
- [12] B. S. Chlebus and J. F. Sibeyn, "Routing on meshes in optimum time and with really small queues," in *Proceedings of IEEE International Symposium on Parallel and Distributed Processing*, April 2003.
- [13] M. Grossglauser and D. Tse, "Mobility increases the capacity of wireless adhoc networks," in *Proceedings of IEEE INFOCOM '01*, April 2001.
- [14] M. D. Neely and E. Modiano, "Improving delay in adhoc mobile networks via redundant packet transfers," in *Proceedings of Conference on Information Sciences and Systems*, March 2003.