

Pareto optimal multi-robot motion planning

Guoxiang Zhao, Minghui Zhu

Abstract—This paper studies a class of multi-robot coordination problems where a team of robots aim to reach their goal regions with minimum time and avoid collisions with obstacles and other robots. A novel numerical algorithm is proposed to identify the Pareto optimal solutions where no robot can unilaterally reduce its traveling time without extending others'. The consistent approximation of the algorithm in the epigraphical profile sense is guaranteed using set-valued numerical analysis. Simulations show the anytime property and increasing optimality of the proposed algorithm.

Index Terms—robotic motion planning, multi-robot coordination, Pareto optimality

I. INTRODUCTION

Robotic motion planning is a fundamental problem where a control sequence is found to steer a mobile robot from an initial state to a goal set, while enforcing dynamic constraints and environmental rules. It is well-known that the problem is computationally challenging. For example, the piano-mover problem is shown to be PSPACE-hard in general [1]. Sampling-based algorithms are demonstrated to be efficient in addressing robotic motion planning in high-dimension spaces. The Rapidly-exploring Random Tree (RRT) algorithm [2] and its variants are able to find feasible paths quickly. However, the optimality of returned paths is probably lost. In fact, computing optimal motion planners is much more computationally challenging than finding feasible motion planners [3]. It is shown that computing the shortest path in $\mathbb{R}^3$ populated with obstacles is NP-hard in the number of obstacles [3]. Recently, RRT* [4] and its variants are shown to be both computationally efficient and asymptotically optimal.

Multi-robot optimal motion planning is even more computationally challenging, because the worst-case computational complexity exponentially grows as the robot number. Current multi-robot motion planning mainly falls into three categories: centralized planning [5] [6], decoupled planning [7] [8] and priority planning [9] [10]. Noticeably, none of these multi-robot motion planners is able to guarantee the optimality of returned solutions. Recent papers [11] and [12] employ game theory to synthesize open-loop planners and closed-loop controllers to coordinate multiple robots, respectively. It is shown that the proposed algorithms converge to Nash equilibrium [13] where no robot can benefit from unilateral deviations. As RRTs, the algorithms in [11] and [12] leverage incremental sampling and steering functions. Steering functions require to solve two-point boundary value

problems. There are only a very limited number of dynamic systems, whose steering functions have known analytical solutions, including single integrators, double integrators and Dubins cars [14]. Heuristic methods are needed to compute steering functions when dynamic systems are complicated.

In the control community, distributed coordination of multi-robot systems has been extensively studied in last decades [15] [16] [17]. A large number of algorithms have been proposed to accomplish a variety of missions; e.g., rendezvous [18], formation control [15], vehicle routing [19] and sensor deployment [20] [21]. This set of work is mainly focused on the design and analysis of algorithms, which are scalable with respect to network expansion. To achieve scalability, most algorithms adopt gradient descent methodologies, which are easy to implement. Their long-term behavior; e.g., asymptotic convergence, can be ensured but usually there is no guarantee on transient performance; e.g., aggregate costs, due to the myopic nature of the algorithms. Another set of more relevant papers is about (distributed) receding-horizon control or model predictive control for multi-robot coordination. Representative works include [22] [23] on formation stabilization, [24] [25] on vehicle platooning and [26] on trajectory optimization. Receding-horizon control has a unique capability to handle constraints on states and inputs. However, it is inherently suboptimal because finite computing horizons are used as approximations. In contrast, multi-robot motion planning aims to find controllers which can optimize certain cost functionals over entire missions; e.g., finding collision-free paths with shortest distances or minimum fuel consumption.

Differential games extend optimal control from single players to multiple players. Linear-quadratic differential games are the most basic, and their solutions can be formulated as coupled Riccati equations [27]. For nonlinear systems with state and input constraints, there are a very limited number of differential games whose closed-form solutions are known, and some examples include the homicidal-chauffeur and the lady-in-the-lake games [27] [28]. Otherwise, numerical algorithms are desired. Existing numerical algorithms are mainly based on partial differential equations [29] [30] [31] and viability theory [32] [33] [34]. Noticeably, this set of papers only considers zero-sum two-player scenarios.

Contribution statement: This paper investigates a class of multi-robot closed-loop motion planning problems where multiple robots aim to reach their own goal regions as soon as possible. The robots are restricted to complex dynamic constraints and need to avoid the collisions with static obstacles and other robots. Pareto optimality is used as the

This work was partially supported by the grant NSF ECCS-1710859.

G. Zhao and M. Zhu are with School of Electrical Engineering and Computer Science, Pennsylvania State University, University Park, PA 16802. (email: {gxzhao, muz16}@psu.edu)

solution notion where no robot can reduce its own travelling time without extending others'. A numerical algorithm is proposed to identify the Pareto optimal solutions. It is shown that, under mild regularity conditions, the algorithm can consistently approximate the epigraph of the minimal arrival time function. The proofs are based on set-valued numerical analysis [32] [33] [34]. Simulations on unicycle robots are conducted to demonstrate the anytime property and increasing optimality of our algorithm; i.e., quickly generating a feasible controller to safely steer the robots to their goal regions and steadily reducing traveling times as more computation time is given.

II. PROBLEM FORMULATION

Consider a team of mobile robots, labeled by $\mathcal{V} \triangleq \{1, \dots, N\}$. The dynamic of robot i is governed by:

$$\dot{x}_i(s) = f_i(x_i(s), u_i(s)), \quad \forall i \in \mathcal{V}, \quad (1)$$

where $x_i(s) \in X_i$ is the state of robot i and $u_i : [0, +\infty) \rightarrow U_i$ is the control of robot i . Here, the state space and the set of all possible control values for robot i are denoted by $X_i \subseteq \mathbb{R}^{d_i}$ and $U_i \subseteq \mathbb{R}^{m_i}$ respectively. The obstacle region and goal region for robot $i \in \mathcal{V}$ are denoted by $X_i^O \subseteq X_i$ and $X_i^G \subseteq X_i \setminus X_i^O$ respectively. Denote the minimum safety distance between any two robots as $\sigma > 0$. The free region for robot i is denoted by $X_i^F \triangleq \{x_i \in X_i \setminus X_i^O \mid \|x_i - x_j\| \geq \sigma, x_j \in X_j^G, i \neq j\}$. Let $\mathbf{X} \triangleq \prod_{i \in \mathcal{V}} X_i$, $\mathbf{X}^G \triangleq \prod_{i \in \mathcal{V}} X_i^G$ and $\mathbf{X}^F \triangleq \prod_{i \in \mathcal{V}} X_i^F$. Assume $\|x_i - x_j\| \geq \sigma, \forall x \in \mathbf{X}^G, i \neq j$. Define the safety region as $\mathbf{S} \triangleq \{x \in \mathbf{X}^F \mid \|x_i - x_j\| \geq \sigma, i \neq j\}$. Here $\|\cdot\|$ denotes the 2-norm.

The sets of state feedback control policies for robot i and the whole robot team are defined as $\varpi_i \triangleq \{\pi_i(\cdot) : \mathbf{X} \rightarrow U_i\}$ and $\varpi \triangleq \{\prod_{i \in \mathcal{V}} \pi_i(\cdot) \mid \pi_i(\cdot) \in \varpi_i\}$ respectively. Consider the scenario where the robot team starts from $x \in \mathbf{X}$ and executes policy $\pi \in \varpi$. The induced minimal arrival time vector is characterized as $\vartheta(x, \pi) \triangleq \inf\{t \in \mathbb{R}_{\geq 0}^N \mid \forall i \in \mathcal{V}, x_i(0) = x_i, \dot{x}_i(s) = f_i(x_i(s), \pi_i(x(s))), x(s) \in \mathbf{S}, x_i(t_i) \in X_i^G, 0 \leq s \leq \max_{i \in \mathcal{V}} t_i\}$, where the infimum uses the partial order in footnote 1. The i -th element of $\vartheta(x, \pi)$ represents the first time robot i reaches its goal region without collisions when the robots start from initial state x and execute policy π . In our multi-robot motion planning problem, the set of Pareto optimal solutions is defined as $\mathcal{U}^*(x) \triangleq \{\pi^* \in \varpi \mid \nexists \pi \in \varpi \text{ s.t. } \vartheta(x, \pi) \neq \vartheta(x, \pi^*) \text{ and } \vartheta(x, \pi) \preceq \vartheta(x, \pi^*)\}$. The interpretation of Pareto optimal solutions is that no robot can unilaterally reach its goal region earlier without extending other robots' travelling times. Denote the minimal arrival time function by $\Theta^*(x) \triangleq \{\vartheta(x, \pi^*) \mid \pi^* \in \mathcal{U}^*(x)\}$. Note that the elements of $\vartheta(x, \pi^*)$ could be infinite, indicating that some robots cannot safely reach the goal regions. Infinite time may cause numerical issues. To tackle this, transformed minimal arrival time function is defined as $v^*(x) \triangleq \Psi(\Theta^*(x))$, where Kruzhkov transform $\Psi(t) \triangleq \text{col}(1 - e^{-t_i})$ for $t \in \mathbb{R}_{\geq 0}^N$ normalizes $[0, +\infty)$ to $[0, 1]$. Notice that Kruzhkov transform is bijective and monotonically

increasing.

The objective of this paper is to identify optimal control policies in $\mathcal{U}^*(x)$ and corresponding minimal arrival time function $\Theta^*(x)$ (or equivalently $v^*(x)$).

III. ASSUMPTIONS AND NOTATIONS

This section summarizes the assumptions, notions and notations used throughout the paper. Most notions and notations on sets and set-valued maps follow the presentation of [35].

The multi-robot system (1) can be written in the differential inclusion form: $\dot{x}_i(s) \in F_i(x_i(s)), \forall s \geq 0$, where the set-valued map $F_i : X_i \rightrightarrows \mathbb{R}^{d_i}$ is defined as $F_i(x_i) \triangleq \{f_i(x_i, u_i) \mid u_i \in U_i\}$. Let $F(x) \triangleq \prod_{i \in \mathcal{V}} F_i(x_i)$. The following assumptions are imposed.

Assumption III.1. *The following properties hold for $i \in \mathcal{V}$:*

- (A1) X_i and U_i are non-empty and compact;
- (A2) $f_i(x_i, u_i)$ is continuous over both variables;
- (A3) $f_i(x_i, u_i)$ is linear growth; i.e., $\exists c_i \geq 0$ s.t. $\forall x_i \in X_i$ and $\forall u_i \in U_i$, $\|f_i(x_i, u_i)\| \leq c_i(\|x_i\| + \|u_i\| + 1)$;
- (A4) For each $x_i \in X_i$, $F_i(x_i)$ is convex;
- (A5) $F_i(x_i)$ is Lipschitz with Lipschitz constant l_i .

Assumptions (A1) and (A2) imply that, for each $i \in \mathcal{V}$, $\exists M_i > 0$ s.t. $\sup_{x_i \in X_i, u_i \in U_i} \|f_i(x_i, u_i)\| \leq M_i$. Define $M^+ \triangleq \sqrt{\sum_{i \in \mathcal{V}} M_i^2}$ and $l^+ \triangleq \sqrt{\sum_{i \in \mathcal{V}} l_i^2}$. Then F is bounded by M^+ and is l^+ -Lipschitz.

Remark III.1. *One sufficient condition of Assumption (A4) is $f_i(x_i, u_i)$ is linear with respect to u_i and U_i is convex. One sufficient condition of Assumption (A5) is $f_i(x_i, u_i)$ is Lipschitz continuous with respect to both variables on $X_i \times U_i$. ■*

Define the distance from a point $x \in \mathcal{X}$ to a set $A \subseteq \mathcal{X}$ as $d(x, A) \triangleq \inf\{\|x - a\| \mid a \in A\}$. A closed unit ball around $x \in \mathcal{X}$ in space $\mathcal{X}$ is denoted as $x + \mathcal{B}_{\mathcal{X}} \triangleq \{y \in \mathcal{X} \mid \|y - x\| \leq 1\}$. Similarly, the unit expansion of a set $A \subseteq \mathcal{X}$ is defined as $A + \mathcal{B}_{\mathcal{X}} \triangleq \{x \in \mathcal{X} \mid d(x, A) \leq 1\}$. Specifically, we denote $x + \mathcal{B}_N \triangleq \{y \in \mathbb{R}^N \mid \|y - x\| \leq 1\}$ if $x \in \mathbb{R}^N$. Similar notation applies to a set A . The subscript of closed unit ball may be omitted when there is no ambiguity. The Hausdorff distance that measures the distance of two sets A and B is defined by $d_H(A, B) \triangleq \inf\{\delta \geq 0 \mid A \subseteq B + \delta \mathcal{B}, B \subseteq A + \delta \mathcal{B}\}$. Kuratowski lower limit and Kuratowski upper limit of sets $\{A_n\} \subseteq \mathcal{X}$ are denoted by $\text{Liminf}_{n \rightarrow +\infty} A_n = \{x \in \mathcal{X} \mid \lim_{n \rightarrow +\infty} d(x, A_n) = 0\}$ and $\text{Limsup}_{n \rightarrow +\infty} A_n = \{x \in \mathcal{X} \mid \liminf_{n \rightarrow +\infty} d(x, A_n) = 0\}$ respectively. If $\text{Liminf}_{n \rightarrow +\infty} A_n = \text{Limsup}_{n \rightarrow +\infty} A_n$, the common limit is defined as Kuratowski limit $\text{Lim}_{n \rightarrow +\infty} A_n$.

The Pareto frontier of a nonempty set $A \subseteq \mathcal{X}$ is denoted as $\mathcal{E}(A) \triangleq \{t \in A \mid \nexists t' \in A \text{ s.t. } t' \neq t, t' \preceq t\}$. Let $A + B \triangleq \{a + b \mid a \in A, b \in B\}$ be the sum of two sets A and B . Denote the n -fold Cartesian product of a set A by A^n .

¹Throughout this paper, product order is imposed; i.e. two vectors $a, b \in \mathbb{R}^N$ are said " a is less than b in the Pareto sense", denoted by $a \preceq b$, if and only if $a_i \leq b_i, \forall i \in \{1, \dots, N\}$. Similarly, strict inequality can be defined by $a \prec b \iff a_i < b_i, \forall i \in \{1, \dots, N\}$.

Specifically, when A is an interval; e.g., $A = [a, b]$, its n -fold product is denoted by $[a, b]^n$. When A is a singleton; e.g., $A = \{a\}$, its n -fold product is written as $\{a\}^n$. Let $A \times \{b\} \triangleq \{(a, b) | a \in A\}$ be the Cartesian product of a set A and a point b . Define Hadamard product for two vectors $a, b \in \mathbb{R}^N$ as $a \circ b \triangleq [a_1 b_1 \ \cdots \ a_N b_N]^T$. Define $a \circ B \triangleq \{a \circ b | b \in B\}$. Denote N -dimensional zero vector and all-ones vector by $\mathbf{0}_N$ and $\mathbf{1}_N$ respectively. The subscript may be omitted when there is no ambiguity. The cardinality of a set is denoted as $|\cdot|$.

Define the distance on two set-valued maps $g, \bar{g} : \mathbf{X} \rightrightarrows [0, 1]^N$ by $d_{\mathbf{X}}(g, \bar{g}) \triangleq \sup_{x \in \mathbf{X}} d_H(g(x), \bar{g}(x))$.

Definition III.1 (Epigraph). *The epigraph of Θ is defined by $\mathcal{Epi}(\Theta) \triangleq \{(x, t) \in \mathcal{X} \times \mathbb{R}^N | \exists t' \in \Theta(x) \text{ s.t. } t \succeq t'\}$.*

Definition III.2 (Epigraphical Profile). *The epigraphical profile of Θ is defined by $E_{\Theta}(x) \triangleq \Theta(x) + \mathbb{R}_{\geq 0}^N$.*

Remark III.2. *For a Kruzhkov transformed function v , we define its epigraphical profile by $E_v(x) \triangleq (v(x) + \mathbb{R}_{\geq 0}^N) \cap [0, 1]^N$.* ■

IV. ALGORITHM STATEMENT AND PERFORMANCE GUARANTEE

In this section, we present our algorithmic solution and summarize its convergence in Theorem IV.1.

A. Algorithm Statement

The proposed algorithm, Algorithm 1, is informally stated as follows. The state space of each robot is discretized by a sequence of finite grids $\{X_i^p\} \subseteq X_i$ s.t. $X_i^p \subseteq X_i^{p+1}, \forall p \geq 1$, where p is the grid index and by convention $X_i^0 = \emptyset$. The state space for the robot team is discretized by $\{\mathbf{X}^p\} \subseteq \mathbf{X}$ with decreasing spatial resolutions $h_p \rightarrow 0$, where $\mathbf{X}^p \triangleq \prod_{i \in \mathcal{V}} X_i^p$. On each grid $\mathbf{X}^p$, Algorithm 1 chooses temporal resolution $\epsilon_p > 2h_p$ and discretizes $\mathbf{S}$ into $\mathbf{S}^p \triangleq (\mathbf{S} + h_p \mathcal{B}) \cap \mathbf{X}^p$. Denote $\mathbb{R}_{\geq 0}^p$ as an integer lattice on $\mathbb{R}_{\geq 0}$ consisting of segments of length h_p , and $(\mathbb{R}_{\geq 0}^N)^p$ as a lattice on $\mathbb{R}_{\geq 0}^N$.

With these spatial and temporal discretization, Algorithm 1 partially solves a multi-robot optimal control problem on grid $\mathbf{X}^p$ via value iterations. Denote the last estimate of minimal arrival time function on $\mathbf{X}^p$ by $v_{\bar{n}_p}^p$, where $\bar{n}_p$ denotes the total number of value iterations executed on $\mathbf{X}^p$ when they terminate. When proceeding to grid $\mathbf{X}^p$, Algorithm 1 first uses $v_{\bar{n}_{p-1}}^{p-1}$ to generate interpolated value function $\tilde{v}^p$ as line 5 to line 16 and then initializes value function v_0^p as line 17 to line 22.

When some robots are considered in the goal regions, they are not supposed to move and affect other robots' motions. Define the set of equivalent nodes $X_E^p(x)$ of $x \in \mathbf{X}^p$ by

$$X_E^p(x) \triangleq \{x' \in \mathbf{X}^p | x_i = x'_i, \forall i \in \mathcal{V} \setminus \mathcal{V}_p^G(x), \\ d(x'_i, X_i^G) \leq M_i \epsilon_p + h_p, \forall i \in \mathcal{V}_p^G(x)\}. \quad (2)$$

Here, $\mathcal{V}_p^G(x) \triangleq \{i \in \mathcal{V} | d(x_i, X_i^G) \leq M_i \epsilon_p + h_p\}$ denotes the set of robots which are close to or already in the goal regions. Then the values of the nodes on the new grid are

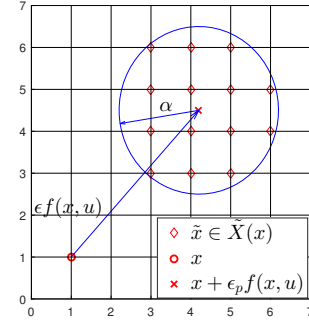


Fig. 1. Set-valued discretization of robot dynamics

initialized by $v_0^p(x) = \bigcup_{\tilde{x} \in X_E^p(x)} \tilde{v}^{p-1}(\tilde{x}), \forall x \in \mathbf{X}^p \setminus \mathbf{X}^{p-1}$, i.e., line 21 in Algorithm 1.

After initialization, as line 24 to 29, Algorithm 1 calls Algorithm 2 to apply value iterations to the nodes in $\mathbf{X}^p$, which are close to or in the goal region. The value iterations are executed for n_p times or until the fixed point is reached. After that, Algorithm 1 refines the grid and begins a new cycle of updates. Notice the total number of value iterations $\bar{n}_p$ may be less than n_p . In particular, Algorithm 2 constructs the following set-valued dynamics to approximate system (1) from lines 2 to 8:

$$\tilde{X}_i^p(x_i) = \begin{cases} x_i + \epsilon_p F_i(x_i) + \alpha_p \mathcal{B}_{X_i}, \\ \quad \text{if } d(x_i, X_i^G) > M_i \epsilon_p + h_p; \\ x_i, \quad \text{otherwise,} \end{cases} \quad (3)$$

and time dynamic $\dot{t} = 1$ is approximated by:

$$\tilde{T}_i^p(x_i) = \begin{cases} \epsilon_p + 2h_p \mathcal{B}_1, & \text{if } d(x_i, X_i^G) > M_i \epsilon_p + h_p; \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where $\alpha_p \triangleq 2h_p + \epsilon_p h_p l^+ + \epsilon_p^{2l^+} M^+$. Let $\tilde{X}^p(x) \triangleq \prod_{i \in \mathcal{V}} \tilde{X}_i^p(x_i) \cap \mathbf{S}^p$ and $\tilde{T}^p(x) \triangleq \prod_{i \in \mathcal{V}} \tilde{T}_i^p(x_i) \cap (\mathbb{R}_{\geq 0}^N)^p$ as line 9 in Algorithm 2. The balls $\alpha_p \mathcal{B}_{X_i}$ in (3) and $2h_p \mathcal{B}_1$ in (4) represent perturbations on the dynamics. The perturbations ensure that the image set of any x is non-empty and the set-valued dynamic is well-defined. See Figure 1 for an illustration of set-valued dynamics (3). Let $\epsilon_p \rightarrow 0$ and $\frac{h_p}{\epsilon_p} \rightarrow 0$; i.e., the spatial resolutions diminish faster than the temporal resolutions. It indicates that the set-valued dynamics well approximate discrete-time systems on $\mathbf{X}$ when ϵ_p is very small. For robots that $d(x_i, X_i^G) \leq M_i \epsilon_p + h_p$, they are treated to be in the goal region, and hence they could stay and their clocks stop counting time.

Given the above set-valued dynamics, Algorithm 2 searches for Pareto optimal solutions of minimal arrival time vectors and stores values in v_n^p and the last controls in $\mathcal{U}^p$. The Bellman operator in the Pareto sense is defined by

$$(\mathbb{T}\Theta_n^p)(x) \triangleq \mathcal{E}(\{\tilde{t} + t | \tilde{t} \in \tilde{T}^p(x), \tilde{x} \in \tilde{X}^p(x), t \in \Theta_n^p(\tilde{x})\}), \quad (5)$$

where $\mathcal{E}$ functions as Pareto minimization and $\Theta_n^p : \mathbf{X}^p \rightrightarrows \mathbb{R}_{\geq 0}^N$ is the estimate of Θ^* on grid $\mathbf{X}^p$ after n value iterations.

Since $\mathcal{E}(\tilde{T}^p(x))$ is a singleton, $\tilde{t} = \mathcal{E}(\tilde{T}^p(x))$. Fix $x \in \mathbf{X}$, apply Kruzhkov transform on both sides of (5) and replace Θ_n^p with $\Psi^{-1}v_n^p$. Then the transformed Bellman operator in the Pareto sense is represented by:

$$(\mathbb{G}v_n^p)(x) = \mathcal{E}(\{\tilde{\tau} + \tau - \tilde{\tau} \circ \tau | \tilde{\tau} = \Psi(\mathcal{E}(\tilde{T}^p(x))), \tilde{x} \in \tilde{X}^p(x), \tau \in v_n^p(\tilde{x})\}), \quad (6)$$

where $\mathbb{G} \triangleq \Psi\mathbb{T}\Psi^{-1}$ is used in line 10 of Algorithm 2. Let $\mathcal{U}^p(x)$ be the set of controls which solve the last value iteration $v_n^p(x) = (\mathbb{G}v_{n-1}^p)(x)$ on grid $\mathbf{X}^p$. It corresponds to line 11 of Algorithm 2.

Algorithm 1 Pareto-based anytime computation algorithm

```

1: Input: System dynamics  $f$ , state space  $\mathbf{X}$ , discretization
   grids  $\{\mathbf{X}^p\}_{p=1}^P$ , the associated resolutions  $h_p$ ,  $\epsilon_p$  and the
   number of updates to be executed  $n_p$ .
2: for  $1 \leq p \leq P$  do
   Grid refinement
3:  $\alpha_p = 2h_p + \epsilon_p h_p l^+ + \epsilon_p^2 l^+ M^+$ ;
4:  $\mathbf{S}^p = (\mathbf{S} + h_p \mathcal{B}) \cap \mathbf{X}^p$ ;
   Value function interpolation
5: for  $x \in \mathbf{X}^{p-1}$  do
6:  $\tilde{v}^{p-1}(x) = v_{n_{p-1}}^{p-1}(x)$ ;
7: end for
8: for  $x \in \mathbf{X}^p \setminus \mathbf{X}^{p-1}$  do
9:   for  $i \in \mathcal{V}$  do
10:    if  $d(x_i, X_i^G) \leq M_i \epsilon_p + h_p$  then
11:       $\tilde{v}_i^{p-1}(x) = 0$ ;
12:    else
13:       $\tilde{v}_i^{p-1}(x) = 1$ ;
14:    end if
15:  end for
16: end for
   Value function initialization
17: for  $x \in \mathbf{X}^{p-1}$  do
18:  $v_0^p(x) = \tilde{v}^{p-1}(x)$ ;
19: end for
20: for  $x \in \mathbf{X}^p \setminus \mathbf{X}^{p-1}$  do
21:  $v_0^p(x) = \bigcup_{\tilde{x} \in X_E^p(x)} \tilde{v}^{p-1}(\tilde{x})$ ;
22: end for
   Value function update
23:  $n = 0$ ;
24: while  $n \leq n_p$  and  $v_n^p \neq v_{n-1}^p$  do
25:    $n = n + 1$ ;
26:   for  $x \in \mathbf{X}^p \setminus (\mathbf{X}^G + (M^+ \epsilon_p + h_p) \mathcal{B})$  do
27:      $(v_n^p(x), \mathcal{U}^p(x)) = \text{VI}(x, \mathbf{S}^p, v_{n-1}^p)$ ;
28:   end for
29: end while
30:  $\bar{n}_p = n$ ;
31: for  $x \in \mathbf{X}^p \cap (\mathbf{X}^G + (M^+ \epsilon_p + h_p) \mathcal{B})$  do
32:    $v_{\bar{n}_p}^p(x) = \tilde{v}^{p-1}(x)$ ;
33: end for
34: end for
35: Output:  $v_{\bar{n}_p}^p, \mathcal{U}^p$ .

```

Algorithm 2 Pareto-based value iteration (VI)

```

1: Input:  $x, \mathbf{S}^p, v_{n-1}^p$ 
2: for  $i \in \mathcal{V}$  do
3:   if  $d(x_i, X_i^G) > M_i \epsilon_p + h_p$  then
4:      $\tilde{T}_i = \epsilon_p + 2h_p \mathcal{B}_1$ ;  $\tilde{X}_i = x_i + \epsilon_p F_i(x_i) + \alpha_p \mathcal{B}_{X_i}$ ;
5:   else
6:      $\tilde{T}_i = \{0\}$ ;  $\tilde{X}_i = \{x_i\}$ ;
7:   end if
8: end for
9:  $\tilde{T} = \prod_{i \in \mathcal{V}} \tilde{T}_i \cap (\mathbb{R}_{\geq 0}^N)^p$ ;  $\tilde{X} = (\prod_{i \in \mathcal{V}} \tilde{X}_i) \cap \mathbf{S}^p$ ;
10:  $v_n^p(x) = \mathcal{E}(\{\tau + \tilde{\tau} - \tau \circ \tilde{\tau} | \tilde{\tau} = \mathcal{E}(\Psi(\tilde{T})), \tilde{x} \in \tilde{X}, \tau \in v_{n-1}^p(\tilde{x})\})$ 
11:  $\mathcal{U}^p(x) = \{\text{the solutions to } u \text{ in the above step}\}$ 
12: Output:  $v_n^p, \mathcal{U}^p$ 

```

B. Performance Guarantee

Recall that n_p at line 24 of Algorithm 1 is the number of value iterations to be executed on grid $\mathbf{X}^p$. The choice of n_p needs to satisfy the following assumption to ensure the convergence of Algorithm 1.

Assumption IV.1. *There is a subsequence $\{D_k\}$ of the grid index sequence $\{p\}$ with $D_0 = 0$ s.t. $D_k - D_{k-1} \leq \bar{D}$ for some constant $\bar{D}$ and all $k \geq 0$ and $\exp(-\sum_{p=D_{k-1}+1}^{D_k} n_p \kappa_p) \leq \gamma < 1$ for every $k \geq 0$, where $\kappa_p \triangleq (\lceil \frac{\epsilon_p}{h_p} \rceil - 2)h_p$ is the minimum running cost.*

Assumption IV.1 implies that the distance between the estimate and the fixed point on the D_k -th grid reduces at least by $\gamma \in [0, 1)$ over the update window length $\{D_{k-1} + 1, \dots, D_k\}$.

The choice of ϵ_p and h_p should satisfy the following.

Assumption IV.2. *The following hold for the sequences of $\{\epsilon_p\}$ and $\{h_p\}$:*

- (A6) $\epsilon_p > 2h_p, \forall p \geq 1$;
- (A7) $\epsilon_p \rightarrow 0$ and $\frac{h_p}{\epsilon_p} \rightarrow 0$ monotonically as $p \rightarrow +\infty$;
- (A8) $2h_p + \epsilon_p h_p l^+ + \epsilon_p^2 l^+ M^+ \geq h_{p-1}, \forall p \geq 1$;
- (A9) $[X_i^G + (\sigma + M_i \epsilon_1 + h_1) \mathcal{B}] \cap X_j^F = \emptyset, \forall i \neq j$.

The consistent approximation of v^* via Algorithm 1 in the epigraphical profile sense is summarized in Theorem IV.1.

Theorem IV.1. *Suppose Assumption III.1, IV.1 and IV.2 hold, then the sequence $\{v_{\bar{n}_p}^p\}$ in Algorithm 1 converges to v^* in the epigraphical profile sense; i.e., for any $x \in \mathbf{X}$,*

$$E_{v^*}(x) = \lim_{p \rightarrow +\infty} \bigcup_{\tilde{x} \in (x + h_p \mathcal{B}) \cap \mathbf{X}^p} E_{v_{\bar{n}_p}^p}(\tilde{x}).$$

C. Discussion

For single robot scenario, if $\bar{D} = 1$ and $\gamma = 0$ with only Assumptions III.1, (A6) and (A7) are imposed, Algorithm 1 and Theorem IV.1 become Algorithm 3.2.4 and Corollary 3.7 in [32] correspondingly.

The progress towards v^* slows down or even stops as more value iterations are performed. A γ close to one ensures that excessive value iterations are postponed to finer grids, and

a longer update interval reduces each grid's efforts to reach the discount factor.

V. SIMULATION

This section presents the simulations conducted to assess the performance of Algorithm 1. The environment of the simulations is a four-way intersection with no signs or signals. Each road consists of two lanes with opposite directions. The width of each lane is 0.5. Each robot is a disc of radius $r = 0.2$. In particular, each robot is a unicycle and its dynamic is given by $\dot{p}_i^x = v_i \cos \theta_i$, $\dot{p}_i^y = v_i \sin \theta_i$, where $x_i = (p_i^x, p_i^y)$ denotes the i -th robot's position and $u_i = (\theta_i, v_i) \in U_i = U_i^\theta \times U_i^v$ is its control including heading angle θ_i and speed v_i . The goal for each robot is to pass the crossroads and arrive at its goal region without colliding with curbs or any other robot. The robots stop as long as they have reached their goal regions.

In practice, the allowable computation times for the robots are varying and uncertain. So it is desired to compute control policies, which can safely steer the robots to their goal regions, within a short time. This is referred to anytime property. The first simulation examines the anytime property of Algorithm 1 for multiple robots. We choose $\epsilon_p = \sqrt{h_p}$. The constraint sets of controls are given as: $U_i^v = [0, 0.25]$, $U_1^\theta = [-\pi, -\pi/2]$, $U_2^\theta = [-\pi/2, \pi/2]$ and $U_3^\theta = [0, \pi]$. The dimensions of state space is 6. For the purpose of collision avoidance, we set the inter-robot safety distance as 0.6 and ignore perturbations added to $\mathbf{S}$ in line 4; i.e., we choose $\mathbf{S}^p = \mathbf{S} \cap \mathbf{X}^p$. Since Algorithm 1 only returns control policies on discrete grids, we need to interpolate the control policies into the continuous state space. In particular, to select one control from $\mathcal{U}^p(x)$ for $x \in \mathbf{S}^p$, uniform sampling is used. For state $x \in \mathbf{X} \setminus \mathbf{S}^p$, the control is interpolated by nearest neighbor method; i.e., we take $u = \text{sampling}(\mathcal{U}^p(\arg \min_{\hat{x} \in \mathbf{S}^p} \|\hat{x} - x\|))$, where $\text{sampling}(\cdot)$ represents uniform sampling. Figure 2 shows the trajectories of the robots when they apply the interpolated control policies which are computed in 0.61s. Figure 3 shows the inter-robot distances over time corresponding to Figure 2, indicating that no collision is caused throughout the movement of robots. Figure 4 displays the linear speeds of each robot over time. Between 0.7s and 1s, both robot 2 and robot 3 slow down such that robot 1 can first pass the crossroads. Due to insufficient value iterations, the returned control policies may not be optimal, and this is the reason why both robot 2 and robot 3 stay still from 3s to 3.5s. The results show that given short computation time; i.e., 0.61s, the algorithm can already generate a feasible policy which accomplishes the planning task without violating any hard constraint.

The second set of simulations examines the increasing optimality of Algorithm 1. The parameters are identical to the previous simulation with the difference that robot 3 is excluded and safety distance is 0.4. The operating region of the robot team is discretized by the sequence of uniform square grids $\{\mathbf{X}^p\}_{p=1}^4$ with decreasing resolutions $h_p \in \{0.2, 0.1, 0.05, 0.025\}$, each of which contains

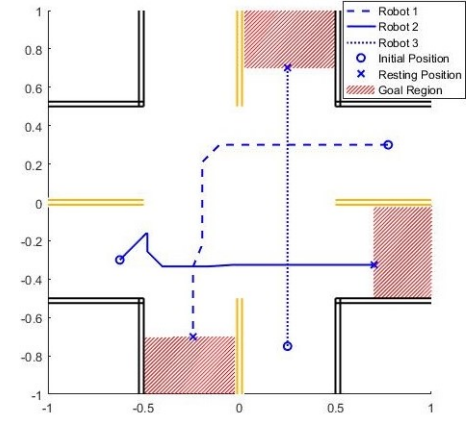


Fig. 2. The trajectories of three robots when the computation time is 0.61s.

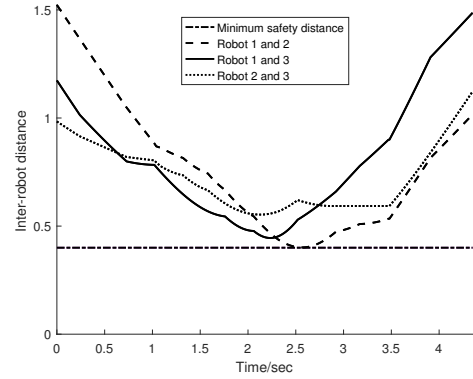


Fig. 3. Inter-robot distances over time.

145, 3403, 34344, 416689 nodes respectively. All the grids are within the same update window. The benchmark v^* is the estimate of minimal arrival time function computed on the finest grid $\mathbf{X}^4$ with resolution $h_p = 0.025$. To measure approximation errors, we use nearest neighbor method to interpolate each estimate of minimal arrival time function v^p into $\hat{v}^p$ so that both $\hat{v}^p$ and v^* share the finest grid as their domains. Note that $\hat{v}^p(x) \triangleq v^p(\arg \min_{\hat{x} \in \mathbf{X}^p} \|\hat{x} - x\|)$ for every $x \in \mathbf{X}^4$, where $\mathbf{X}^p$ is the domain of v^p . Then approximation error of $\hat{v}^p$ is measured by $\|\hat{v}^p - v^*\|_{\mathbf{X}^4}$. Figure 5 shows the approximation errors over time. The peaks at 1s, 20s and 600s are caused by grid refinements, where a number of new nodes are added. These new nodes introduce large approximation errors. Other than these, the approximation errors are monotonically decreasing over time.

VI. CONCLUSION

In this paper, a numerical algorithm is proposed to find the Pareto optimal solution of a class of multi-robot motion planning problems. The consistent approximation of the algorithm is guaranteed using set-valued analysis. A set of simulations are conducted to assess the anytime property and increasing optimality. There are a couple of interesting problems to solve in the future. First, the proposed algorithm is centralized. It is of interest to study distributed implemen-

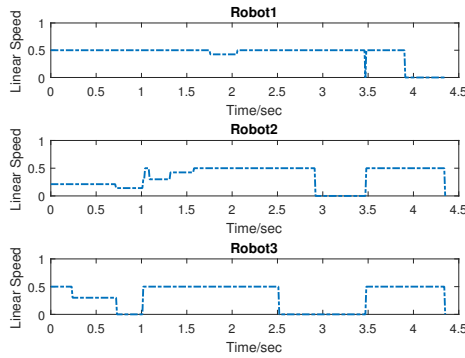


Fig. 4. Robot linear speeds over time.

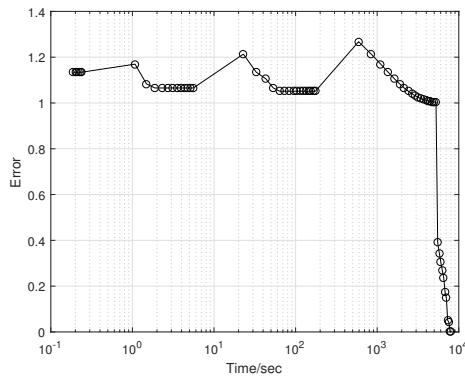


Fig. 5. Approximation errors over time.

tation. Second, the simulations show that the approximation error increases when the grid is refined. It is of interest to study new grid refinement schemes to reduce the amplitudes of error rebound.

REFERENCES

- [1] J. H. Reif, "Complexity of the mover's problem and generalizations," in *IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 421–427, 1979.
- [2] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [3] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.
- [4] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," in *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [5] G. Sánchez and J.-C. Latombe, "On delaying collision checking in PRM planning: Application to multi-robot coordination," in *The International Journal of Robotics Research*, vol. 21, no. 1, pp. 5–26, 2002.
- [6] E. K. Xidias and N. A. Aspragathos, "Motion planning for multiple non-holonomic robots: A geometric approach," in *Robotica*, vol. 26, no. 04, pp. 525–536, 2008.
- [7] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," in *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.
- [8] T. Siméon, S. Leroy, and J.-P. Laumond, "Path coordination for multiple mobile robots: A resolution-complete algorithm," in *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.
- [9] S. J. Buckley, "Fast motion planning for multiple moving robots," in *Proceedings of IEEE International Conference on Robotics and Automation*. 1989, pp.322-326.

- [10] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," in *Algorithmica*, vol. 2, no. 1-4, pp. 477–521, 1987.
- [11] M. Zhu, M. Otte, P. Chaudhari, and E. Frazzoli, "Game theoretic controller synthesis for multi-robot motion planning Part I: Trajectory based algorithms," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 1646–1651.
- [12] D. K. Jha, M. Zhu, and A. Ray, "Game theoretic controller synthesis for multi-robot motion planning Part II: Policy-based algorithms," in *The 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, vol. 48, pp. 168–173, 2015.
- [13] J. Nash, "Non-cooperative games," in *Annals of Mathematics*, pp. 286–295, 1951.
- [14] S. Karaman and E. Frazzoli, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5041–5047.
- [15] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.
- [16] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.
- [17] M. Zhu and S. Martinez, *Distributed optimization-based control of multi-agent networks in complex environments*. Springer, 2015.
- [18] M. Cao, A. S. Morse, and B. D. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," in *SIAM Journal on Control and Optimization*, vol. 47, no. 2, pp. 575–600, 2008.
- [19] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *43rd IEEE Conference on Decision and Control*, 2004, pp. 3357–3363.
- [20] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [21] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.
- [22] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization" in *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.
- [23] M. Zhu and S. Martínez, "On distributed constrained formation control in operator-vehicle adversarial networks" in *Automatica*, vol. 49, no. 12, pp. 3571–3582, 2013.
- [24] W. B. Dunbar and D. S. Caveney, "Distributed receding horizon control of vehicle platoons: Stability and string stability" in *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 620–633, 2012.
- [25] H. Li, Y. Shi and W. Yan, "Distributed receding horizon control of constrained nonlinear vehicle formations with guaranteed γ -gain stability" in *Automatica*, vol. 68, no. 68, pp. 148–154, 2016.
- [26] Y. Kuwata, A. Richards, T. Schouwenaars and J. P. How, "Distributed robust receding horizon control for multivehicle guidance" in *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 627–641, 2007.
- [27] T. Basar, G. J. Olsder, "Dynamic noncooperative game theory" in *SIAM Classics in Applied Mathematics*, 1999.
- [28] R. Issacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*, Courier Corporation, 1999.
- [29] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Springer Science & Business Media, 2008.
- [30] M. Bardi, M. Falcone and P. Soravia, "Numerical methods for pursuit-evasion games via viscosity solutions" in *Annals of the International Society of Dynamic Games*, vol.4, pp.105–179, 1999.
- [31] P. E. Souganidis, "Two-player, zero-sum differential games and viscosity solutions" in *Annals of the International Society of Dynamic Games*, vol.4, pp.69–104, 1999.
- [32] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre, "Set-valued numerical analysis for optimal control and differential games," in *Stochastic and differential games*. Springer, 1999, pp. 177–247.
- [33] J.-P. Aubin, *Viability theory*. Springer Science & Business Media, 2009.
- [34] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre, *Viability theory: New directions*. Springer Science & Business Media, 2011.
- [35] J.-. Aubin and H. Frankowska, *Set-valued analysis*. Springer Science & Business Media, 2009.