Multi-Vehicle Path Coordination in Support of Communication

Pramod Abichandani, Hande Y. Benson, and Moshe Kam

Abstract— **This paper presents a framework for generating time-optimal velocity profiles for a group of path-constrained vehicle robots that have fixed and known initial and goal locations and are required to maintain communication connectivity. Each robot must follow a fixed and known path, arrive at its goal as quickly as possible (or at least not increase the time for the last robot to arrive at its goal) and stay in communication with other robots in the arena throughout its journey. The main contribution of this paper is the formulation of the problem as a discrete time nonlinear programming problem (NLP) with constraints on robot kinematics, dynamics, collision avoidance, and communication connectivity. We develop** ***Partition Elimination*** **constraints that assist in ensuring that the communication network is fully connected (no network partitions). These constraints are enforced only when network partitions would otherwise occur, an approach which significantly reduces the problem size and the required computational effort.**

In addition, we introduce path-constrained jammer robots with known paths and velocity profiles into the scenario. These jammer robots have an effective jamming range and disrupt all communications within this range. Except for the jammers, all robots must remain outside this jamming range at all times. We investigate the scalability of the proposed approach by testing scenarios involving up to fifty (50) robots. Solutions demonstrate (i) the trade off between the arrival time and the communication connectivity requirements in scenarios with and without jamming; and (ii) the dependence of computation time on the number of robots.

I. INTRODUCTION

The coordination of the motion of a number of robots (say n) in a shared workspace so that they avoid collisions is known as the *multiple robot path coordination problem* [1], [2]. In previous work [3] we addressed this problem under fixed communication connectivity constraints by generating time optimal velocity profiles. The robots were confined to fixed paths and sought to arrive from a set of initial points to specified final destinations.

Pramod Abichandani is a Ph.D. candidate at the Data Fusion Laboratory, Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA 19104, USA pva23@drexel.edu

Hande Y. Benson is an Assistant Professor at the Department of Decision Sciences, Drexel University, Philadelphia, PA 19104, USA hvb22@drexel.edu

Moshe Kam is the Department Head and Robert Quinn Professor of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA kam@minerva.ece.drexel.edu

The one-hop communication constraint in [3] ensured that each robot remains in immediate communication range with k other robots, (k = 1,...,n) and not use any intermediary robots as relays.

In this paper, we consider a more general problem, where we seek to coordinate the movement of the robots from initial locations to final destinations while using intermediate robots for communication. Moreover, we require that throughout the scenario each robot can communicate, directly or indirectly, with other other robots. Thus the network never gets partitioned. We formulate constraints called the "partition elimination constraints", which are enforced only when network partitions would otherwise occur, an approach which greatly reduces the problem size and the required computational effort.

Jammer robots are a part of the scenario, attempting to disrupt communication of the non-jamming robots. These jammers can be thought of as stationary or mobile obstacles with an effective jamming radius. Once a robot (other than a jammer robot) is within the jamming radius, it loses communication with all other robots. Thus the robots must try to stay out of the jamming ranges at all times.

The problem is formulated as a nonlinear programming (NLP) problem. The fixed paths of the robots are represented by piecewise cubic spline curves. The feasibility criteria for trajectories require that the robots' kinematic and dynamic constraints be satisfied, along with avoiding collisions and obeying the communication constraints (including the avoidance of partitions). The spline paths are generated using Matlab, which is interfaced with the modeling environment AMPL [4]. We use the software package LOQO [5] with AMPL to solve the NLP.

In this paper, the following assumptions are made

- All calculations are made by a central coordinator.
- Since this study deals with higher level velocity planning, we assume that for each vehicle, there exists a perfect lower level control to achieve the planned position and heading angle without any time delay.
- There is no slipping in the motion of the robots. This is a valid assumption since we enforce kinematic (non-holonomic constraints, continuous first

and second derivatives of the spline paths) and dynamic constraints (upper bounds on the velocities and accelerations) which disallow such slippage.
- Free space propagation between vehicle antennas is assumed. Factors such as multi path propagation, fading, time delay and crosstalk are not considered.
- The path and velocity profile of the jammers are known (in reality these will have to be estimated).

II. RELATED WORK

While a significant body of work was devoted to path planning for mobile robots [1], [6], we focus our work on the relatively untouched area of velocity planning along predetermined routes. Indeed, more often than not one does not get the liberty of planning an arbitrary path around sparse obstacles, and must rather follow a prescribed route. These prescribed routes can also be thought of as the solutions to a path planning problem.

Several approaches have been used to address the problem of path coordination of multiple robots [1], wherein multiple robots with fixed paths coordinate with each other so as to avoid collisions and reach destination points. These approaches include the use of coordination diagrams [7], constrained configuration space roadmap [8], grouping robots with shared collision zones into subgroups [2]. In [9], mixed integer linear programming (MILP) formulations were used to generate continuous velocity profiles for a group of robots that satisfy kino-dynamics constraints, avoid collisions and minimize the task completion time. In [3], we extended this body of work by adding communication constraints to the problem, incorporating the resulting nonlinearities, and using state of the art interior point methods to solve the resulting NLP [10].

III. PROBLEM FORMULATION

A. Robot Motion and Path

Consider a two-wheeled differential drive mobile robot as shown in Fig. 1. The robot moves in a global (X, Y) Cartesian co-ordinate plane and is represented by the following kinematic model with an associated non-holonomic constraint (that disallows sliding sideways).

$$\dot{x} = v\cos(\theta); \quad \dot{y} = v\sin(\theta); \quad \dot{\theta} = \omega \tag{1}$$

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0. \tag{2}$$

Here v and ω are the linear and angular velocities of the robot respectively; x, y and θ are the coordinates of the robot with respect to the global (X, Y) coordinate system. Consider a group of n such mobile robots. Each robot $i = 1, 2,...,n$ is represented by a common mathematical model (1) with associated non-holonomic constraint (2) and has a fixed path p^i of length len^i to follow, with a given start (origin) point o^i and a given end (goal) point e^i. P is the set of all the fixed paths of each robot. $p^i(x(t), y(t)) \in P, \forall i = 1, 2, ...n$. O is the set of all start (origin) points. $o^i \in O, \forall i = 1, 2, ...n$. E is the set of all end points. $e^i \in E, \forall i = 1, 2, ...n$. The Euclidean distance between two robots i and j is denoted by d^{ij}. The robots are required to maintain a minimum safe distance d_{safe} from each other in order to avoid collisions. At any given time step, the distance between the current location and the goal point for robot i is given by d^i_{goal}. s^i and ω^i denote the speed and angular velocity respectively of robot i along its fixed path at a given time.

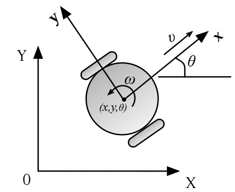

Fig. 1. Robot architecture and notations

We formulate this scenario as a discrete time problem with the parameter t representing steps in time. $T_{\max}$ is the time taken by the last robot to reach its end point. At $t = T_{\max}$ the mission is completed. If a robot reaches the goal point before $T_{\max}$, it continues to stay there till the mission is over. If required, such a robot can still communicate with other robots.

Let M be the set of n_{jam} communication jammers, where each jammer $m = 1, 2,...,n_{jam}$ is a mobile robot represented by (1) - (2) and has a fixed path with given start and end points. The effective jamming range of a jammer is d_{jam}. In this study, all the jammers are assumed to have same jamming radius.

Each robot follows a fixed path represented by a two dimensional piecewise cubic spline curve, which is obtained by combining two one dimensional piecewise cubic splines $x(u)$, and $y(u)$, where the parameter u is arc length along the curve. Let $\kappa(u)$ be the curvature along the spline curve. For each robot i,

$$\omega^i(u) = s^i(u)\kappa^i(u) \tag{3}$$

These piecewise cubic splines have continuous first derivatives (slope) and second derivatives (curvature) along the curve. This property makes the path kinematically feasible. Furthermore, upper and lower bounds on the speed, acceleration and angular speed (turning rate) are enforced, thereby taking into account the robot dynamics. The paths represented by the two dimensional piecewise cubic splines along with the constraints on the speed, accelerations and turn rates result in a kinodynamically feasible trajectory. For a detailed discussion on spline curve design and analysis, see [11] and its references.

B. Communication Model

Each robot is equipped with a wireless transceiver node. The Signal to Noise Ratio (SNR) experienced by the receiver robots is calculated to determine whether the robots are in communication range of each other. If the SNR experienced by a receiver node placed on a robot is above a predefined threshold τ, the two robots are considered to be in communication range of each other.

Consider two robots that try to communicate with each other at a given point in time. The Euclidean distance between them is denoted by d. The signal transmission power of the wireless node placed on the transmitter robot is denoted by P_{tr}. The received signal power of the wireless node placed on the receiver robot is denoted by P_r. The power experienced by the receiver robot node is calculated using Friis's equation

$$P_r = P_{tr} G_t G_r \left(\frac{\lambda}{4\pi d}\right)^{\alpha}, \qquad (4)$$

where α is the path loss exponent. The noise is assumed to be thermal ($kTBF$). λ is the wavelength and is equal to c/f, where c = 3 x 10^8 m/s and f = 2.4 x 10^9 Hz. The values of G_t and G_r (antenna gains) is assumed to be 1. The values of α range from 1.6 (indoor with line of sight) to 6 (outdoor obstructed) depending on the environment. The value α = 2 corresponds to free space.

IV. MODEL FORMULATION

We seek to minimize $T_{\max}$, the time of arrival of the last arriving robot, while each robot tries to remain in one-hop communication range with at least k other robots and network partitions are eliminated at all time steps. The optimization is performed over the speeds of the robots along the specified paths. We identify the time instances when partitions may occur and eliminate them by changing the velocity profiles of the robots at these times. Since $T_{\max}$ is not known *a priori*, we pick a sufficiently large number of time steps T ($T_{\max} \leq T$) in our model so as to get a feasible solution. For a given value of k between 0 to $n-1$, the problem described by (5)-(21) is then solved.

A. Decision Variables

In (5)-(21), the main decision variables are the speeds, $s^i(t)$, for vehicle i at time t. The values of the remaining variables are dependent on the speeds, as described in the following subsections on the problem constraints (subsections *C-I*).

$$\begin{aligned}
\text{minimize} \quad & T_{\max} + \sigma \sum_{i,t} d^i_{goal}(t) && (5)\\
& \forall i \in \{1,2,...n\}, \quad \forall t \in \{1,2,...T\}, \\
& \forall j \in \{1,2,...n\}, j \neq i \\
& \forall m \in \{1,2,...n_{jam}\} \\
\text{subject to} \quad & (x^i(0), y^i(0)) = o^i && (6)\\
& (x^i(T), y^i(T)) = e^i && (7)\\
& u^i(0) = 0 && (8)\\
& u^i(t) = u^i(t-1) + s^i(t)\Delta t && (9)\\
& (x^i(t), y^i(t)) = ps^i(u^i(t)) && (10)\\
& s_{min} \leq s^i(t) \leq s_{max} && (11)\\
& \dot{s}_{min} \leq \dot{s}^i(t) \leq \dot{s}_{max} && (12)\\
& d^{ij}(t) \geq d_{safe} && (13)\\
& d^{im}(t) \geq d_{jam} && (14)\\
& 0 \leq A^i(t) \leq 1 && (15)\\
& A^i(t) d^i_{\text{goal}}(t) = 0 && (16)\\
& \forall i, T_{\max} \geq \left(\sum_{t=0}^{T} \left(1 - A^i(t)\right) \right) && (17)\\
& 0 \leq C^{ij}(t) \leq 1 && (18)\\
& l^{ij}(t) = \text{SNR}^{ij}_r(t) - \tau && (19)\\
& C^{ij}(t) l^{ij}(t) \geq 0 && (20)\\
& \sum_{j:j\neq i} C^{ij}(t) \geq k && (21)
\end{aligned}$$

B. Objective Function

Equation (5) represents the objective function to be minimized. The first term of the objective function is $T_{\max}$ which represents the time taken by the last robot to reach its goal point. A second term with a penalty parameter σ forces the robots to reduce the distance between their current location and the goal position. This term prevents the robots from stalling.

C. Path (Kinematic) Constraint

Constraints (6)-(10) define the path of each robot. Constraints (6) and (7) form the set of boundary requirements that each robot i has to start at a designated start point o^i and finish at a designated end point e^i at the end of the planning horizon. Constraint (8) initializes the arc length travelled, u, to zero value. Constraint (9) increments the arc length at each time step based on the speed of the robot. Δt is the discrete time step. Constraint (10) ensures that the robots follow their respective paths as defined by the cubic splines. The function $ps^i(u^i(t))$ denotes the location of robot i at time step t after travelling an arc length of u along the piecewise cubic spline curves.

D. Speed and Acceleration (Dynamic) Constraint

Constraints (11)-(12) are dynamic constraints and ensure that the speed (and hence, angular velocity) and the acceleration, respectively, are bounded from above and below. These constraints are determined by the capabilities of the robot and the curvature of the paths represented by the spline curve. Here we assume that the curvature of the paths is within the achievable bounds of the angular speed and radial acceleration of the robots. Hence the angular speed required by the robots corresponding to the optimal speed is always achievable (and can be determined by equation (3)).

E. Collision Avoidance Constraint

Constraint (13) ensures that there is a sufficiently large distance between each pair of robots to avoid a collision.

F. Communication Jamming Constraint

If robot i is within jamming range of one or more jammer robots, it loses all its capabilities to communicate with the other robots. Accordingly constraint (14) is added to ensure that each robot remains outside the jamming range of the jammer robots at all times.

G. Definition of $T_{\max}$

As defined by constraints (15) and (16), $A^i(t)$ measures the number of time periods for which robot i is not at its destination. The equilibrium constraint (16) and the bounds on $A^i(t)$ (15) ensure that when $d^i_{goal}(t) > 0$, $A^i(t) = 0$. Therefore, if $A^i(t) = 1$ for all (i, t) with $d^i_{goal}(t) = 0$, the total amount of time it takes a robot i to reach its destination is

$$\sum_{t=0,\ldots,T} \left(1 - A^i(t)\right) \tag{22}$$

The equilibrium constraint (16) cannot alone guarantee that the property (22) will hold. However, constraint (17) specifies $T_{\max}$ as an upper bound for (22), and equation (5) minimizes $T_{\max}$. Therefore at the optimal solution, for the last robot(s) to reach its (their) destination(s), $A^i(t) = 1$ when robot i is at its destination at time t and (17) will hold with equality.

Note that the solution obtained by including the constraints (15)-(17) is equivalent to the one obtained by using the following mixed-integer definition:

$$\begin{aligned} A^i(t) &= 0 \text{ if } (d^i_{goal}(t) \neq 0) && (23)\\ &= 1 \text{ if } (d^i_{goal}(t) = 0) \\ T_{\max} &= \max_{i=1,\ldots,n} \left(\sum_{t=0,\ldots,T} \left(1 - A^i(t)\right) \right) && (24) \end{aligned}$$

It is, however, more advantageous for computational efficiency of the algorithm to solve an NLP instead of a mixed integer nonlinear programming problem (MINLP). With recent research in handling equilibrium constraints in NLPs, handling the resulting nonsmoothness is not a complicating factor in the solution process. For details on how the solver used in this study (LOQO) handles equilibrium constraints, see [10].

H. One-hop Communication Connectivity Constraint

Constraints (18)-(21) define the requirement that each robot must be in one-hop communication with at least k other robots at all times. If there is a need for each robot to communicate within one hop with a greater number of robots (e.g., for contingency planning), the right-hand side of constraint (21) will be increased.

Constraint (19) defines an intermediate variable, $l^{ij}(t)$, that aids in defining the communications constraint. $l^{ij}(t) \geq 0$ indicates that the two robots i and j are in one-hop communication range whereas; $l^{ij}(t) < 0$ indicates that these two robots are not in one-hop communication range of each other.

Constraint (20) then ensures that if there is no one-hop communication between robots i and j at time t, then the variable $C^{ij}(t)$ must necessarily equal 0. That is, if pairwise communication is lost, we have that $l^{ij}(t) < 0$ and since constraint (18) requires that the value of $C^{ij}(t) \geq 0$, the only way to satisfy constraint (20) is to have $C^{ij}(t) = 0$. If there is one-hop communication between the two robots at time t, then $C^{ij}(t)$ can take on any value between 0 and 1, inclusive, as allowed by constraint (18). Finally, constraint (21) ensures that for each robot i at time t, at least k of the $C^{ij}(t)$, $j \in \{1, 2, ...n\}$, $j \neq i$ are greater than zero.

I. Connectivity Partition Elimination Constraint

The one-hop communication connectivity constraints will not necessarily prevent partition of the network to non-communicating subgroups. A constraint of the form

$$\sum_{i \in \mathbb{I}, j \notin \mathbb{I}} C^{ij}(t) \geq 1 \tag{25}$$

for each subgroup $\mathbb{I}$ of size $k+1, ..., n-1$ at each time period would ensure that there exists at least one connection between each subgroup thereby resulting in connectivity throughout the network. We will refer to (25) as the *"partition elimination"* (P.E.) constraints.

The number of partition elimination constraints p to be added to the problem expressed by (5)-(21) for all possible partitions is given by

$$p = \sum_{i=k+1}^{\lfloor n/2 \rfloor} \binom{n}{i} - \frac{h}{2}\binom{n}{\frac{n}{2}} \tag{26}$$

where $h = 1$ if n is even and $n > 2k$, and $h = 0$ otherwise.

Adding partition elimination constraints for all possible partitions for each time period to the problem expressed by (5)-(21) would increase its size exponentially. Instead, we apply Algorithm 1 (also referred to as the P.E. algorithm), which solves the problem without these constraints first and then detects any partitions in the solution. The P.E. algorithm is a form of breadth first search algorithm that searches the entire communication connectivity graph to detect partitions. For each partition detected, we add one constraint of the form (25) to (5)-(21). The updated problem is re-solved and the process is continued until no more partitions are detected. This greatly reduces the size of the problem. By using the partition elimination algorithm at each iteration, we solve a relaxation of the actual problem.

V. SIMULATIONS AND RESULTS

A. Simulation Setup

The Matlab function `spline()` was used to generate piecewise cubic splines passing through randomly generated waypoints. The spline paths are parametrized by arc length u. The optimization model, defined by (5)-(21) was implemented in the modeling environment AMPL and the solver LOQO was used . The AMPL-LOQO combination was implemented on a PC running RedHat Linux 2.4.20-8 with 512MB of main memory and a 2.4GHz clock speed. In our numerical testing, we have used LOQO Version 6.07 compiled with the AMPL solver interface Version 20021031.

B. Simulations

We focus on the effect of the partition elimination constraints and the presence of jammers on velocity profile design for the group. We have tested our model for scenarios that include up to fifty (50) mobile robots and up to four (4) jammer robots, and a number of communications constraints. In the following discussion, we plot the spline curve paths of the robots with different colors indicating different robots and different robot groups. The origin and destination points of each robot is indicated by a dot marking. The triangular markings on the curves indicate the position of the robot in the (X,Y) Cartesian coordinate plane at each step in time while traveling at optimal speeds along the path. The parameters used in our simulations are listed in Table. I. For all the simulations the value of P_{tr} is in milliwatts and $\Delta t = 1$ second. In all the examples, the robots are required to maintain one hop communication connectivity with one other robot, i.e., k = 1 in (21).

Algorithm 1 Partition Elimination (P.E.)

Require: n robots with fixed paths
Ensure: Eliminate partitions
  repeat
    Solve the model.
    Let Done = FALSE
    for $t \in \{1, \ldots, T\}$ **do**
      Let $\mathbb{I} = \{1\}$.
      for $i \in \mathbb{I}, j \notin \mathbb{I}$ **do**
        if $C^{ij}(t) > 0$ **then**
          $\mathbb{I} = \mathbb{I} \cup \{j\}$
        end if
      end for
      if $\mathbb{I} = \{1, \ldots, n\}$ **then**
        Done = TRUE
      else
        add the constraint $\sum_{i \in \mathbb{I}, j \notin \mathbb{I}} C^{ij}(t) \geq 1$ to the model
      end if
    end for
  until Done = FALSE

TABLE I

PARAMETER VALUES USED FOR SIMULATIONS

d_{safe}	0.01 m	s_{min}	0	s_{max}	2 m/s
$\dot{s}_{min}$	-1 m/s^2	$\dot{s}_{max}$	0.5 m/s^2	σ	100
d_{jam}	0.45 m	Δt	1 s	$\dot{s}_{jam}$	0 m/s^2
τ	4.5 x 10^{-3}	T	10	α	2

Readers are referred to our previous work [3] to see the effect of varying k on the velocity profiles of the robots. In all plots, the triangular markings on different paths do not overlap with each other completely at any point in time. This observation indicates that the robots indeed do not collide with each other at any point in time (thus satisfying the collision avoidance constraint).

1) Effect of partition elimination constraints on the velocity profile:

- *4 robots*:
  Fig. 2 shows the trajectories of the robots in a four (4) robot scenario in presence (right side plot) and absence (left side plot) of the partition elimination constraints. In absence of the partition elimination constraints, it is observed that all robots travel at the maximum allowed speed at all times. In fact the robots form two subgroups based on the physical proximity of their paths and the one-hop communication constraint is always satisfied. Partitions are detected at time periods 2 and 3.
  After applying the partition elimination algorithm to establish communication between the two subgroups of vehicles at time periods 2 and 3, the trajectory of robot 2 changes. Robot 2 slows down in order to maintain connectivity. In both cases, the mission ends at $T_{\text{max}} = 8$.
- *10 robots*:
  Fig. 3 shows the trajectories of the robots in a ten (10) robot scenario in presence (right side plot) and absence (left side plot) of the partition elimination constraints. The velocity profiles of robots 2 and 3 change due to the partition elimination constraints, and are demonstrated in Fig. 4. The transmission power in this case is 1.3 mW and the transmission range is 1.69 meters. Before applying the partition elimination constraints, the value of $T_{\text{max}} = 7$. After the partition elimination constraints were added, the value increases to $T_{\text{max}} = 8$.
- *20 robots*:
  Fig. 5 shows the trajectories of the robots in a twenty (20) robot scenario in presence (right side plot) and absence (left side plot) of the partition elimination constraints. The velocity profile of Robot 1 changes due to the presence of the partition elimination constraints. Since the paths of the robots are very close to each other, the transmission power required for feasibility in this case is 0.2 mW which corresponds to a transmission range of 0.66 meters. Before applying the partition elimination constraints, the value of $T_{\text{max}} = 7$. After the partition elimination constraints were added, the value increases to $T_{\text{max}} = 8$.
- *50 robots*:
  Fig. 6 shows the trajectories of the robots in a fifty (50) robot scenario in presence (right side plot) and

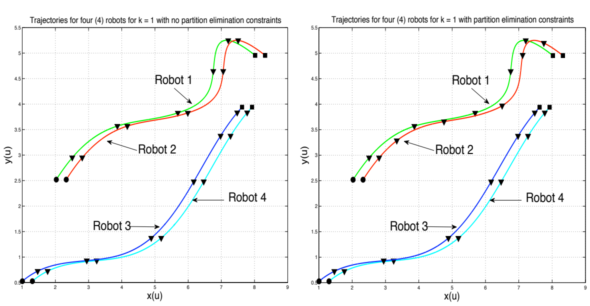

Fig. 2. A 4 robot scenario with and without the partition elimination constraints

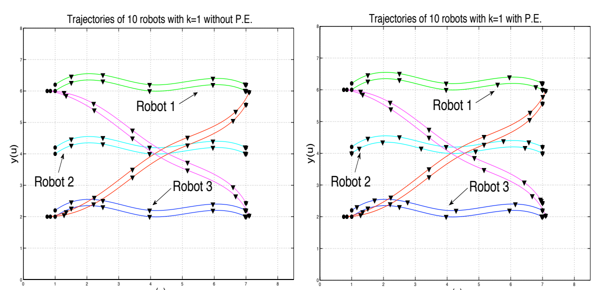

Fig. 3. A 10 robot scenario with and without the partition elimination constraints

TABLE II

EFFECT OF n ON T_{COMP} AND PROBLEM SIZE REDUCTION

n	$T_{\max}$ no PE	$T_{\max}$ with PE	T_{comp}	p no PE algorithm	p with PE algorithm
4	8	8	7.785	30	2
10	7	8	46.869s	5010	3
20	7	8	123.217s	5242670	1
50	7	9	1314.897s	5.6295 x 10^{15}	1

absence (left side plot) of the partition elimination constraints. It can be clearly observed that the robot velocity profiles change considerably in order to comply with the communication requirements. The transmission power in this case is 2.2 mW and the transmission range is 2.2 meters. The value of $T_{\max}$ increases from 7 to 9 after the partition elimination constraints are added.

The following observations are made

- The partition elimination constraints make a difference; they make the robots change their speeds in order to maintain connectivity.
- Typically, the "fast" robots whose times of arrival at their respective destinations are less than $T_{\max}$ change their velocity profiles to comply with the new communication connectivity constraint.

2) Scalability and problem size reduction:

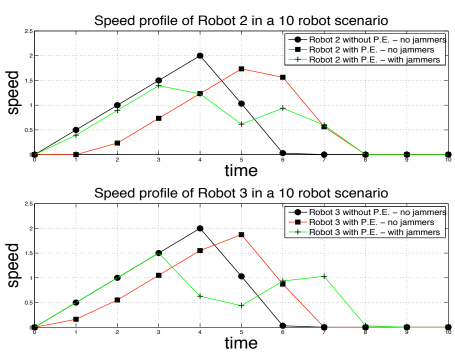

Fig. 4. Velocity profiles of Robot 2 and Robot 3 in a 10 robot scenario with and without the partition elimination constraints in absence and presence of jammers

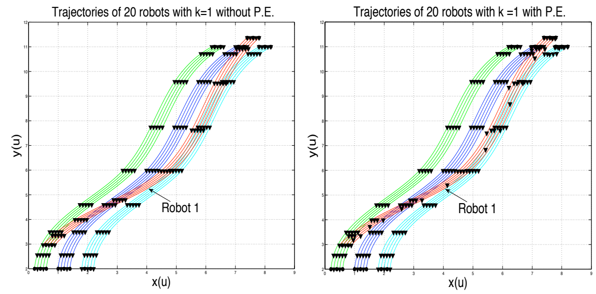

Fig. 5. A 20 robot scenario with and without the partition elimination constraints

- *Scalability with respect to the number of robots n*: We demonstrate the effect of increasing the number of robots n on the computational time. Table II summarizes the results obtained in cases of 4, 10, 20, and 50 robots. The total computational time T_{comp} is measured in seconds and is reported along with the values of $T_{\max}$ for all cases with and without the PE algorithm. It is observed that T_{comp} increases with the increase in the number of robots.
- *Problem size reduction*:
  In our computational experience it is observed that with the use of the P.E. Algorithm in the simulated scenarios, the number of partition elimination constraints p added to the problem are greatly reduced, (see Table II).
  For the earlier problem with 4 vehicles, k = 1, and T = 10, we would have had to add 30 partition elimination constraints if we included all such constraints

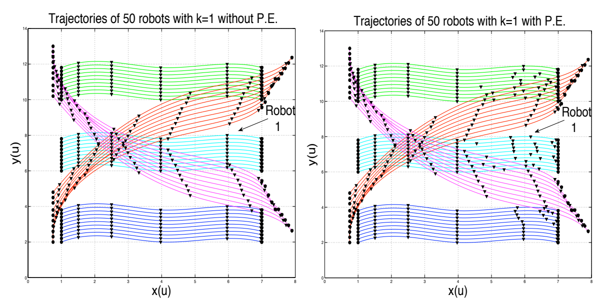

Fig. 6. A 50 robot scenario with and without the partition elimination constraints

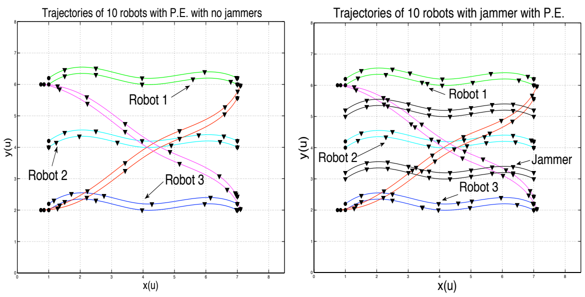

Fig. 7. A 10 vehicle scenario in absence and presence of 4 jammers traveling at constant speed. The speeds of the robots change in order to avoid being jammed.

in the problem since there would be $\frac{1}{2}\binom{4}{2}$ possible partitioned subgroups of 2 robots per subgroup at each time period. In this example, by using P.E. algorithm only 2 of these constraints are added, one at time step 2 and the other at time step 3.

3) Effect of presence of jammer robots on the velocity profile: For a ten robot scenario, Fig. 7 shows trajectories of the robots in absence and presence of a set of 4 jammers. We consider four jammer robots with (black colored) spline paths that move at a constant, predetermined speeds s_{jam}. The triangular markings show the position of the jammers at different time steps. The partition elimination algorithm is applied. When constraints of the form (14) are added to the model, the robots change their velocity profiles in order to remain outside the jamming distance of the jammer d_{jam}. The velocity profiles of Robots 2 and 3 are indicated in the Fig. 4. The velocity profiles of these robots are different when compared to the case with no jammer. The value of $T_{\max}$ is 8 in both cases. In this case, s_{jam} = 0.6 m/s.

VI. CONCLUSIONS

Using the proposed framework, we formulated the problem as a NLP and generated time optimal velocity profiles for a group of path constrained mobile robots with fixed initial and goal points that were subject to kinodynamic and collision avoidance constraints and were required to maintain connectivity through out the network by avoiding network partition. Furthermore, we generated time optimal velocity profiles for this group of robots in the presence of jammer robots. We reported on the scalability of the proposed approach by investigating scenarios involving up to 50 robots. Finally, it was observed that the P.E. algorithm developed in this paper significantly reduces the number of P.E. constraints added to the optimization problem resulting in the scenarios considered, thereby reducing the overall problem size. Future efforts will be directed towards applying this paradigm in a decentralized and distributed fashion by taking into account any stochastic disturbances affecting the system and comparing the results with the proposed centralized approach.

REFERENCES

[1] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, 1991.

[2] T. Simeon, S. Leroy, and J. Laumond, “Path coordination for multiple mobile robots: a resolution-complete algorithm,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.

[3] P. Abichandani, H. Benson, and M. Kam, “Multi-vehicle path coordination under communication constraints,” in *Proc. American Control Conference (ACC’08)*, Seattle,WA, June 2008.

[4] R. Fourer, D. Gay, and B. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. Scientific Press, 1993.

[5] R. Vanderbei and D. Shanno, “An interior-point algorithm for nonconvex nonlinear programming,” *Computational Optimization and Applications*, vol. 13, pp. 231–252, 1999.

[6] E. Todt, G. Raush, and R. Sukez, “Analysis and classification of multiple robot coordination methods,” in *Proceedings of the International Conference on Robotics and Automation, San Francisco, CA*, 2000.

[7] P. O’Donnell and T. Lozano-Perez, “Deadlock-free and collision-free coordination of two robot manipulators,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Scottsdale, AZ*, 1989, pp. 484–489.

[8] S. LaValle and S. Hutchinson, “Optimal motion planning for multiple robots having independent goals,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, 1998.

[9] J. Peng and S. Akella, “Coordinating multiple robots with kinodynamic constraints along specified paths,” *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 295–310, 2005.

[10] H. Y. Benson, A. Sen, D. Shanno, and R. J. Vanderbei, “Interior point algorithms, penalty methods and equilibrium problems,” *Computational Optimization and Applications*, vol. 34(2), pp. 155–182, June 2006.

[11] M.Lepetic, G. Klancar, I. Skrjanc, D. Matko, and B. Potocnik, “Time optimal path planning considering acceleration limits,” *Robotics and Autonomous Systems*, vol. 45, pp. 199–210, 2003.