Spatio-Temporal Multi-Robot Routing *

Smriti Chopra, Magnus Egerstedt

Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract

In this paper, we consider the problem of routing multiple robots to service spatially distributed requests at specified time instants. We show that such a routing problem can be formulated as a pure assignment problem. Additionally, we incorporate connectivity constraints into the problem by requiring that range-constrained robots ensure a connected information exchange network at all times. We discuss the feasibility aspects of such a *spatio-temporal* routing problem, and derive the minimum number of robots required to service the requests. Moreover, we explicitly construct the corresponding routes for the robots, with the total length traveled as the cost to be minimized.

Key words: assignment problems; autonomous mobile robots; connectivity constraints; optimization problems; vehicle routing.

1 Introduction

Multi-robot routing requires multiple robots to visit a set of spatially distributed locations for some purpose (e.g., delivery or acquisition) with routes that optimize certain criteria (e.g., minimization of total distance traveled, completion time, or energy consumption). In this paper, we consider such a problem of servicing spatial requests, with an added temporal constraint that each request be serviced at a specified time instant. Moreover, we consider the connectivity constrained version of the problem, where we require that the underlying information exchange network remains connected at all times.

In the robotics literature, multi-robot routing is a well studied topic (e.g. see Burgard et al. [2005], and Mosteo et al. [2008]). Many problems on combinatorial optimization are associated with multi-robot routing. For instance, the multiple traveling salesman problem (m-TSP) consists of determining a set of optimal routes for m salesmen who all start from and turn back to a home city (see Bektas [2006]). Another example is the vehicle routing problem (VRP) (see Arsie et al. [2009]), which concerns the design of optimal delivery or collection routes for a fleet of vehicles from one or many depots to a number of geographically scattered customers with known demands. The dynamic counterpart of the VRP, known as the dynamic vehicle routing problem, deals with online arrival of customer demands during the operation (see Bullo et al. [2011], and Pavone and Frazzoli [2010]).

Applications of such routing problems include surveillance, search and rescue, transportation on demand, and assembly. However, to solve these problems is computationally expensive. In fact, the VRP is proven to be NP-hard (see Karp [1972]). To overcome this complexity, one can note that many times, applications require an ordered sequence in which requests be serviced. For instance, an autonomous structure assembly system, or a car manufacturing system, may require multiple robots to service locations in a synchronized and sequenced manner, thus motivating the need for *spatio-temporal* requests in lieu of spatial requests. In this paper, we show that by adding such temporal constraints to the spatial requests, a notion of directionality appears in the otherwise NP- hard problem of routing, and thus, it can be converted to an assignment problem, solvable in polynomial time (for preliminary results in this direction, see Chopra and Egerstedt [2012b]).

An important aspect of multi-robot coordination concerns connectivity maintenance, where in order to ensure that the robots can execute a mission in a collaborative manner, the induced information exchange network must be sufficiently rich. In this paper, we require that the range-constrained network induced by the positions of the robots be connected for all times (for preliminary results, see Chopra and Egerst-

* This work was sponsored by the ONR grant MURI HUNT. We would also like to thank Leonard Tsai for his help with the hardware implementation.

Email addresses: smriti.chopra@gatech.edu (Smriti Chopra,), magnus@ece.gatech.edu (Magnus Egerstedt).

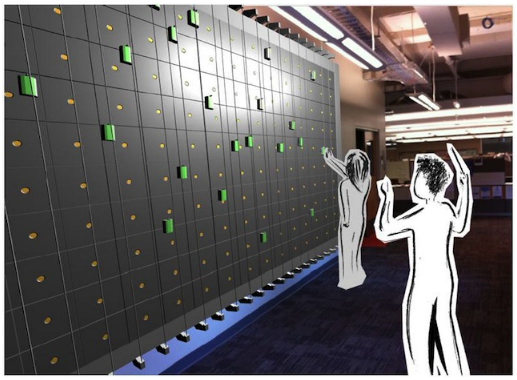


Fig. 1. A rendering of the *Robot Music Wall* concept

edt [2012a]). In general, connectivity maintenance in multi-robot networks requires techniques for ensuring connectivity of a range constrained multi-robot network during some task execution. Such techniques include using relays dedicated towards maintaining sensing or communication links (e.g. Nguyen et al. [2003], and Dixon and Frew [2009]), or using formation control strategies towards motion planning (see Kan et al. [2011]). Other methods seek connectivity at *particular* time instants only, (e.g. Ponda et al. [2011]). However, we are interested in constructing routes that maintain connectivity *for all times*, while allowing dynamic assignment between robots and *spatio-temporal* requests such that no robots exist *solely* for the task of maintaining connectivity links.

This paper is organized as follows : In Section 2, we discuss the Unconstrained Routing Problem, followed by its corresponding Connectivity Constrained version in Section 3. Finally, we demonstrate the routing problems through simulations and hardware implementations, in Section 4.

A Motivating Example - The Robot Music Wall

Consider a two-dimensional magnetic-based surface (wall) with a grid of strings in different pitches that generate sound when plucked. Distinct positions on the wall correspond to distinct sound frequencies, i.e. distinct notes of an instrument. Multiple robots with the ability to traverse the wall can reach these positions and pluck at the strings above them.

With this set-up, we can interpret any piece of music consisting of a series of notes to be played at specified time instants, as a series of corresponding *spatio-temporal* requests (timed positions) on the music wall. We call such a series a *Score*, which contains positions that must be reached at specified time instants. By routing multiple robots to service such timed positions, we can effectively “play” the piece of music associated with them on the wall.

2 The Unconstrained Routing Problem

We let $T = \{t_1, t_2, ..., t_n\}$ denote the set of n discrete time instants over which the *Score* is defined, where $t_1 < ... < t_n$. Moreover, we let P_i denote the corresponding set of planar positions that require simultaneous servicing at time t_i. Each position in this set is denoted by $P_{i,\alpha}$, where $\alpha \in \{1, ..., |P_i|\}$ (the symbol $|\cdot|$ denotes cardinality), i.e.,

$$P_i = \{P_{i,\alpha} \,|\, \alpha \in \{1, ..., |P_i|\}\}, \quad \forall i \in \{1, ..., n\} \tag{1}$$

We let $\mathcal{K}$ be the maximum number of positions that require simultaneous servicing at any time instant in T, i.e.,

$$\mathcal{K} = \max_{i \in \{1,...,n\}} |P_i| \tag{2}$$

Definition 1. Let the *Score*, denoted by Sc, be the set of all timed positions that the robots must reach. We express such timed positions as (position, time) pairs in the *Score*, i.e.,

$$Sc = \{(P_{i,\alpha}, t_i) \,|\, i \in \{1, ..., n\}, \alpha \in \{1, ..., |P_i|\}\} \tag{3}$$

Moreover, for a given set of r robots, denoted by $R = \{1, ..., r\}$, we let $P_0 = \{P_{0,\alpha} \,|\, \alpha \in \{1, ..., |P_0|\}\}$ be the set of their initial positions, defined at time instant t_0.

Notice that if we have fewer robots than the maximum number of positions requiring simultaneous servicing in the *Score*, given by $\mathcal{K}$, then all $\mathcal{K}$ positions cannot be reached simultaneously. Thus, we must have at least $\mathcal{K}$ robots, i.e. $r \geq \mathcal{K}$.

We are interested in the problem of optimally routing these robots to reach the timed positions contained in the *Score*. By optimal, we mean a routing plan that minimizes the total distance traveled by the robots. Moreover, we want our solution to act at a high enough level of abstraction so that the dynamics of the robots do not have to be explicitly accounted for. This construction must be inherently hybrid in that it connects the continuous dynamics to a discrete solution. Hence, we assume single integrator dynamics for every robot, given by $\dot{x}_p = u_p$, $p \in R$. Since for such systems, minimum distance paths are straight lines and minimum energy motions have constant velocities, we let robots move between assigned positions in straight line paths with constant velocities that ensure their timely arrival.

Note that we can interpret the path of any robot as a *series of individual assignments* between timed positions assigned to that robot, directed in increasing order of specified time instants. Hence, the information contained in the optimal paths of the robots can be encoded in a different function that explicitly describes such individual assignments. We elaborate on this in subsequent paragraphs,

Definition 2. Let the $Assignees$, denoted by As, be the set containing all timed positions in the *Score* specified before the last time instant t_n, in addition to all timed initial positions of the robots, i.e.,

$$As = \{(P_{i,\alpha}, t_i) \,|\, i \in \{0, ..., n-1\}, \alpha \in \{1, ..., |P_i|\}\} \tag{4}$$

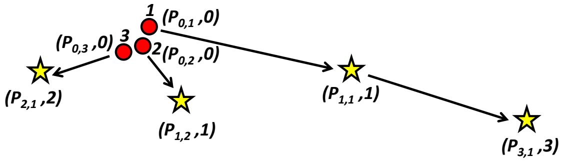


Fig. 2. An example of an *optimal assignment* between three robots (circles) and a *Score* (stars)

Note that $r \geq \mathcal{K}$ implies that $|As| \geq |Sc|$.

We let $\pi : As \rightarrow Sc$ be a function that maps between timed positions in the *Assignees* and the *Score*. If there exists some $As' \subseteq As$, such that firstly, the restricted function $\pi|_{As'} : As' \rightarrow Sc$ is a bijection, and secondly, $\pi((P_{i,\alpha}, t_i)) = (P_{j,\beta}, t_j) \in Sc \Rightarrow t_j > t_i$ for all $(P_{i,\alpha}, t_i) \in As'$, then we call this restricted function a *feasible assignment*. The first condition ensures that every timed position in the *Score* is assigned, no two timed positions in the *Assignees* map to the same timed position in the *Score*, and no two timed positions in the *Score* are assigned to the same timed position in the *Assignees*. The second condition enforces directionality within each individual assignment, i.e. it states that a position in the *Score* specified at time instant t_j must be assigned to a position in the *Assignees* specified at some time instant t_i earlier than t_j i.e. $t_i < t_j$. We call this the directionality constraint.

In addition to being *feasible*, if the total distance associated with the individual assignments in $\pi|_{As'}$ is minimum (akin to saying that the total distance traveled by all the robots is minimum), then we call it an *optimal assignment*, denoted by $\pi^\star$. Note that $\pi^\star$ is restricted to the subset $As' \in As$ because the condition $|As| \geq |Sc$ forces $(|As| - |Sc|)$ number of timed positions in As to go unassigned, in order to ensure $\pi^\star$ is indeed a bijection.

See Figure 2 for an example of an *optimal assignment* $\pi^\star : As' \rightarrow Sc$, where,

$R = \{1, 2, 3\}$
$Sc = \{(P_{1,1}, 1), (P_{1,2}, 1), (P_{2,1}, 2), (P_{3,1}, 3)\}$
$As = \{(P_{0,1}, 0), (P_{0,2}, 0), (P_{0,3}, 0), (P_{1,1}, 1), (P_{1,2}, 1), (P_{2,1}, 2)\}$
$As' = \{(P_{0,1}, 0), (P_{0,2}, 0), (P_{0,3}, 0), (P_{1,1}, 1)\}$
$\pi^\star((P_{0,1}, 0)) = (P_{1,1}, 1)$
$\pi^\star((P_{1,1}, 1)) = (P_{3,1}, 3)$
$\pi^\star((P_{0,2}, 0)) = (P_{1,2}, 1)$
$\pi^\star((P_{0,3}, 0)) = (P_{2,1}, 2)$

From the above example, we can see that by applying $\pi^\star$ repeatedly on the initial position of a robot, we can determine the corresponding optimal path for that robot. Thus, we focus on the problem of finding an *optimal assignment* $\pi^\star$ for a given triple (Sc, R, P_0).

2.1 Problem Definition

Let $\mathcal{I} \triangleq \{0, ..., n-1\}$ and $\mathcal{J} \triangleq \{1, ..., n\}$ be the index sets for i and j, representing the time instants at which positions are specified in As and Sc respectively. Also, let $\mathcal{A}_i \triangleq \{1, ..., |P_i|\}$, $i \in \{0, ..., n\}$ be the index set for α and β.

By defining a mapping $l(i, \alpha, j, \beta)$, we can formally define the routing problem as a linear program,

$$\min_l \sum_{i \in \mathcal{I}} \sum_{\alpha \in \mathcal{A}_i} \sum_{j=i+1}^{n} \sum_{\beta \in \mathcal{A}_j} ||P_{j,\beta} - P_{i,\alpha}||\, l(i, \alpha, j, \beta) \quad (5)$$

subject to:

$$l(i, \alpha, j, \beta) \in \{0, 1\} \quad (6)$$

$$\sum_{i=0}^{j-1} \sum_{\alpha \in \mathcal{A}_i} l(i, \alpha, j, \beta) = 1,\ \forall\, j \in \mathcal{J}, \beta \in \mathcal{A}_j \quad (7)$$

$$\sum_{j=i+1}^{n} \sum_{\beta \in \mathcal{A}_j} l(i, \alpha, j, \beta) \leq 1,\ \forall\, i \in \mathcal{I}, \alpha \in \mathcal{A}_i \quad (8)$$

where $l(i, \alpha, j, \beta)$ represents the individual assignment of $(P_{i,\alpha}, t_i) \in As$ to $(P_{j,\beta}, t_j) \in Sc$, and is 1 if the assignment is done, and 0 otherwise. The resulting l gives us the corresponding *optimal assignment* $\pi^\star$, where $l(i, \alpha, j, \beta) = 1 \iff \pi^\star((P_{i,\alpha}, t_i)) = (P_{j,\beta}, t_j)$. Equations (7) and (8) ensure *feasibility* of this assignment, while (5) ensures that the total distance associated with the individual assignments is minimum.

For the sake of convenience, we refer to the Unconstrained Routing Problem given by Equations (5) - (8), as the *URP*.

2.2 Assignment

The *URP* is a modified version of the classic *linear sum assignment problem* (*LSAP*) (see Martello and Toth [1987]) that concerns the following: given two equal sized sets P and Q with some non-negative cost function $\mathcal{C} : (P \times Q) \rightarrow \mathbb{R}$, the objective is to find a complete assignment, i.e. a bijection $S : P \rightarrow Q$ that minimizes the function $\sum_{a \in P} \mathcal{C}(a, S(a))$. As and Sc in the *URP* correspond to P and Q in the *LSAP*, and the *feasible assignment* π corresponds to S. Note that the *LSAP* insists on P and Q being equal sized while the *URP* insists on $|As| \geq |Sc|$. Moreover, in the *LSAP*, there exist no forbidden individual assignments between P and Q, contrary to the *URP*, where individual assignments between As and Sc that violate the directionality constraint are forbidden. However, we can apply algorithms developed for solving the *LSAP* towards solving the *URP*, by incorporating certain modifications that we discuss later in this section.

Many algorithms, both sequential and parallel, have been developed for solving the *LSAP* (e.g. Martello and Toth [1987]), ranging from primal-dual combinatorial algorithms to simplex-like methods, cost operation algorithms, forest algorithms, and relaxation approaches. Although immaterial to the underlying theory, in this paper, we choose to use the first polynomial-time primal-dual algorithm developed for solving the *LSAP*, called the *Hungarian Method* (see Kuhn [1955]). Note that the fastest version of the *Hungarian Method* involving N stages is $\mathcal{O}(N^3)$ (see implementation in Lawler [1976]).

In order to use the *Hungarian Method* towards solving the *URP*, we generate a cost matrix $C = [c_{i_\alpha, j_\beta}]$ of size $|As| \times |Sc|$, where the rows and columns in C represent timed positions in the *Assignees* and the *Score* respectively, and the element c_{i_α, j_β} equals the distance between the corresponding timed positions, i.e. $||P_{j,\beta} - P_{i,\alpha}||$, $(P_{i,\alpha}, t_i) \in As$ and $(P_{j,\beta}, t_j) \in Sc$. The *Hungarian Method* requires a square cost matrix, for which we introduce $(|As| - |Sc|)$ *dummy* [1] positions as targets (in addition to the timed positions in the *Score*). For convenience, we denote the set of such *dummy* positions by $P_{n+1} = \{P_{n+1,\beta} \,|\, \beta \in \mathcal{A}_{n+1}\}$ where $\mathcal{A}_{n+1} \triangleq \{1, 2, ..., (|As| - |Sc|)\}$. Moreover, we let the cost associated with reaching these *dummy* positions be zero. We define Sc' to be the set containing the *Score* in addition to the *dummy* positions, i.e., $Sc' = Sc \cup \{(P_{n+1,\beta}, t_{n+1}) \,|\, \beta \in \mathcal{A}_{n+1}\}$.

The *Hungarian Method* operates on such a square cost matrix to search for a one-to-one correspondence (assignment) between its row and column elements (assignees and targets respectively), such that the assignment has a minimum cost. In the case of the *LSAP*, it always terminates with a complete assignment, i.e. a bijective function $H : P \to Q$ with minimum cost, denoted by $cost(H)$. More importantly, there exist no targets in Q that go unassigned. However, in the case of the *URP*, there exist forbidden individual assignments between As and Sc' that need to be taken into account. The way these are typically dealt with within the framework of the *Hungarian Method*, is to associate a prohibitively large cost M with each of them (see e.g. Burkard et al. [2009]). We denote this modified cost matrix by $\hat{C} = [\hat{c}_{i_\alpha, j_\beta}]$, where,

$$\hat{c}_{i_\alpha, j_\beta} = \begin{cases} ||P_{j,\beta} - P_{i,\alpha}||, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i, \\ & j \in \{i+1, ..., n\}, \beta \in \mathcal{A}_j \\ M, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i, \\ & j \in \{1, ..., i\}, \beta \in \mathcal{A}_j \\ 0, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i, \\ & j = n+1, \beta \in \mathcal{A}_j \end{cases} \tag{9}$$

The symbol M represents forbidden assignments. If M is large enough, then the *Hungarian Method* finds a complete assignment between As and Sc' that avoids forbidden individual assignments, if such an assignment exists. We denote the assignment by $H_r : As \to Sc'$, where r refers to the number of robots.

2.3 *Existence of Solutions*

Since the cost associated with reaching a *dummy* position in Sc' is zero, H_r always contains $(|As| - |Sc|)$ number of individual assignments between timed positions in the *Assignees* and *dummy* positions. Thus, all timed positions in the *Score* are reached, i.e. $Sc \subseteq range(H_r)$, *if and only if* H_r is a complete assignment that avoids forbidden individual assignments. Moreover, given such a complete assignment, we can construct an *optimal assignment* $H_r|_{As'} : As' \to Sc$, where $As' \subseteq As$ is the set of timed positions in the *Assignees* that are *not* assigned to *dummy* positions. We denote such an assignment by $H_r^\star$ ($H_r|_{As'}$ is a bijection that satisfies the directionality constraint, with minimum associated cost).

Theorem 1 *Given the URP ((5) - (8)), the Hungarian Method operating on the cost matrix $\hat{C}$, given by (9), produces an optimal assignment $H_r^\star$.*

Proof : We only provide the outline of the proof, due to space considerations (for the complete proof, see Chopra and Egerstedt [2012b]). In particular, we consider a bijective function $\omega_r : As \to Sc'$ such that ω_r contains individual assignments between all timed positions in As and Sc' that have a corresponding diagonal cost entry in the cost matrix $\hat{C}$. By showing that no diagonal element in $\hat{C}$ equals M, we prove that ω_r is a complete assignment that avoids forbidden individual assignments. The existence of such an assignment further proves that the *Hungarian Method* always finds a complete assignment that avoids forbidden individual assignments, i.e. a bijection $H_r : As \to Sc'$ which is, at the very least, equal to ω_r. Thus, we can construct an *optimal assignment* $H_r|_{As'} : As' \to Sc$, denoted by $H_r^\star$, where $As' \subseteq As$ is the set of timed positions in the *Assignees* that are *not* assigned to *dummy* positions. ■

Theorem 2 *The minimum number of robots, given by $r^\star$, required to ensure that a solution exists to the URP, equals the maximum number of positions that require simultaneous servicing in the Score ($\mathcal{K}$), i.e.,*

$$r^\star = \min_r \{Sc \subseteq range(H_r)\} = \mathcal{K} \tag{10}$$

Proof : If $r < \mathcal{K}$, then there are not enough robots to ensure that all $\mathcal{K}$ positions (specified at some time instant t_j, $j \in \mathcal{J}$), are reached simultaneously. Moreover, from Theorem 1, we know that for $r \geq \mathcal{K}$, there exists a solution to the *URP* that can be found through the *Hungarian Method*. Hence, $r^\star = \mathcal{K}$. ■

3 The Connectivity Constrained Routing Problem

In this section, we incorporate connectivity constraints in the *URP* discussed previously in this paper, i.e., we require that the range-constrained network induced by the positions of the robots be connected at all times.

[1] The term *dummy* is standard in the assignment literature, e.g. see Martello and Toth [1987].

Recall that for the set of robots R, we represent each robot as a planar point particle with single integrator dynamics $\dot{x}_p(t) = u_p(t)$, $p \in R$, $t \in [t_0, t_n]$. We assume that u_p is continuous almost everywhere, and we use the notation $x_p \in \hat{C}_2[t_0, t_n]$ to denote this fact (x_p denotes the differentiable *almost everywhere* trajectory of robot p over the interval $[t_0, t_n]$). Moreover, we let $X(t)$ denote the set of positions of all robots at time t, i.e., $X(t) = \{x_p(t) \,|\, p \in R\}$. By defining $\mathcal{C}_r$ as the following space,

$$\mathcal{C}_r = \underbrace{\hat{C}_2[t_0, t_n] \times ... \times \hat{C}_2[t_0, t_n]}_{r\,\text{copies}}$$

$X \in \mathcal{C}_r$ as such, denotes a collection of differentiable *almost everywhere* trajectories of the robots over the time interval $[t_0, t_n]$. Additionally, we let $d_{p,q}(t)$ denote the Euclidean distance between robots p and q, i.e.,

$$d_{p,q}(t) = ||x_p(t) - x_q(t)|| \tag{11}$$

Each robot has a fixed sensing range $\Delta \in \mathbb{R}$. In other words, at a given time t, robot p can sense (or "see") all robots that lie within a circle of radius Δ centered at $x_p(t)$. Since all robots possess the same range Δ, sensing links between pairs of robots are bidirectional, i.e. if robot p can sense robot q, then robot q can sense robot p as well. The positions of the robots and the resulting sensing links induce a $\Delta-$ disk proximity graph $G(X(t), \Delta)$, where the vertex set of G is given by the set R, and distinct vertices p and q share an edge in G *if and only if* the distance between them $(d_{p,q})$ is at most equal to Δ, i.e.,

$$(p, q) \in E(G) \iff \Delta - d_{p,q} \geq 0 \tag{12}$$

where $E(G)$ denotes the edge set of G.

3.1 Feasibility

In this section, we discuss the feasibility aspects of the Connectivity Constrained version of the *URP*.

Definition 3. Given (Sc, R, P_0, Δ), $X \in \mathcal{C}_r$ is *feasible* if it satisfies the following conditions,

(a) $P_i \subseteq X(t_i) \;\forall\; i \in \{0, ..., n\}$
(b) $G(X(t), \Delta)$ is connected $\forall\; t \in [t_0, t_n]$

The first condition ensures that every timed position in the *Score* is reached by a robot. The second condition ensures that the $\Delta-$ disk proximity graph induced by the positions of the robots is connected for all time over the interval $[t_0, t_n]$.

We let $\mathcal{F}_r \subseteq \mathcal{C}_r$ denote the set of all *feasible* sets of trajectories,

$$\mathcal{F}_r = \{X \,|\, X \in \mathcal{C}_r \, is\, feasible\}, \tag{13}$$

which allows us to state the Feasibility Problem :

Given (Sc, Δ), the objective is to find the minimum number of robots, $r^\star$ such that $\mathcal{F}_{r^\star} \neq \emptyset$ for the corresponding $(Sc, R^\star, P_0, \Delta)$ quadruple, where $R^\star = \{1, ..., r^\star\}$ is the set of robots and P_0 is the set of their initial positions.

3.1.1 Establishing Feasibility

In this section, we present results on the existence of a *feasible* set of trajectories.

Theorem 3 *Given (Sc, R, P_0, Δ), there exists $X \in \mathcal{C}_r$ such that,*

(a) $P_i \subseteq X(t_i) \;\forall\; i \in \{0, ..., n\}$
(b) $G(X(t_i), \Delta)$ is connected $\forall\; i \in \{0, ..., n\}$

if and only if there exists a set of trajectories X' such that $X' \in \mathcal{F}_r$ and $X'(t_i) = X(t_i) \;\forall\; i \in \{0, ..., n\}$.

Proof : Assume that there exists $X \in \mathcal{C}_r$ that satisfies the above conditions (a) and (b). Notice that both these conditions constrain X at only *particular* time instants, i.e. the initial time instant (t_0) and the subsequent *Score* time instants $(t_1, ..., t_n)$. In other words, the conditions constrain sets of robot positions $X(t_i)$, $i \in \{0, ..., n\}$.

Consider a pair of such sets of robot positions, specified at successive time instants, denoted by say $X(t_{i-1})$ and $X(t_i)$. From condition (b), we see that the corresponding induced graphs $G(X(t_{i-1}), \Delta)$ and $G(X(t_i), \Delta)$ are connected. For such a pair of connected graphs, it was shown in Spanos and Murray [2005] that there exist *connectivity preserving* motions from one configuration to another. In other words, there exists $X' \in \mathcal{C}_r$ such that $X'(t_{i-1}) = X(t_{i-1})$ and $X'(t_i) = X(t_i)$, and $G(X'(t), \Delta)$ is connected over the interval (t_{i-1}, t_i). Moreover, one can see that such a X' exists between *every* pair of successive sets of positions, thereby proving the existence of a set of piecewise robot trajectories $X' \in \mathcal{F}_r$, where $X'(t_i) = X(t_i) \;\forall\; i \in \{0, ..., n\}$.

Conversely, if we assume that there exists $X'(t) \in \mathcal{F}_r$ such that $X'(t_i) = X(t_i) \;\forall\; i \in \{0, ..., n\}$, then by definition, $X(t) \in \mathcal{C}_r$ satisfies conditions (a) and (b). ■

Theorem 3 states that the positions of the robots at *particular* time instants are sufficient to determine the existence of a *feasible* set of trajectories. However, in order to ensure that the positions satisfy conditions (a) and (b), we need to first ensure that we have enough robots. The following equations establish such a requirement for a minimum number of robots,

$$r < \mathcal{K} \Rightarrow \mathcal{F}_r = \emptyset \tag{14}$$
$$r \geq \mathcal{K} \not\Rightarrow \mathcal{F}_r \neq \emptyset \tag{15}$$

Equation (14) states that if the number of robots is *less than* the maximum number of positions requiring simultaneous servicing ($\mathcal{K}$) in the *Score*, then there are not enough robots to ensure that all those positions are occupied simultaneously. In other words, condition (a) would never be satisfied, and consequently, there would exist no *feasible* set of trajectories. Equation (15), on the other hand, states that having a minimum of $\mathcal{K}$ number of robots *still* does not guarantee the existence of a *feasible* set of trajectories. For instance, it is entirely possible that, for a given range Δ, the positions requiring simultaneous servicing at some time instant in the *Score* are located so far apart from one another that $\mathcal{K}$ number of robots are just not enough to induce a connected $\Delta-$ disk proximity graph at that time instant. In other words, condition (b) would never be satisfied, resulting in $\mathcal{F}_r = \emptyset$.

Thus, we proceed to find the minimum number of robots $r^\star$ that ensures that conditions (a) and (b) from Theorem 3 can be met, which in turn would prove the existence of a *feasible* set of trajectories. To keep the problem of finding $r^\star$ independent of the initial positions of the robots, we make the following assumption,

Assumption : The starting position of every robot in R is chosen such that the induced $\Delta-$ disk proximity graph $G(X(t_0), \Delta) = G(P_0, \Delta)$ is connected.

Theorem 4 *Given a set of positions P_i specified at time t_i in the Score, and a sensing range Δ, the problem of finding the minimum number of robots, r_i, that ensures that every position in P_i is occupied, and the induced $\Delta-$ disk proximity graph is connected, is equivalent to the Steiner tree problem with minimum number of Steiner points and bounded edge length (STP-MSPBEL).*

Proof : The *STP-MSPBEL* in its general form (see Lin and Xue [1999]) is stated as follows,

"Given a set of planar positions P, and a positive constant R, the objective of the STP-MSPBEL is to find a tree spanning a superset of P such that each edge in the tree has a length no more than R and the number of points other than those in P, called Steiner points, is minimized."

We see that the problem of finding the minimum number of robots at time instant t_i is identical to the *STP-MSPBEL*, where the position set P_i corresponds to P and the range Δ corresponds to the positive constant R. The positions of the vertices of the solution tree denote the positions of the robots, thus ensuring that each position in P_i is occupied, and the induced $\Delta-$ disk proximity graph is connected. ■

We denote the positions of the robots in the solution tree by S_i, and the number of *Steiner Points* by s_i. Note that $s_i + |P_i| = |S_i| = r_i$. Notice that from Theorem 4, we get the minimum number of robots required at a particular time instant in the *Score*, such that conditions (a) and (b) in Theorem 3 *evaluated at that particular time instant*, can be met. However, each time instant in the *Score* may require a different minimum number of robots, depending on its corresponding specified position set. Thus, in order to obtain a global minimum that ensures that both conditions (a) and (b) in Theorem 3 can be met, we must take the maximum over all time instants, of the minimum number of robots.

Theorem 5 *For a given (Sc, Δ), the minimum number of robots, $r^\star$ that ensures $\mathcal{F}_{r^\star} \neq \emptyset$ for the corresponding $(Sc, R^\star, P_0, \Delta)$ quadruple, where $R^\star = \{1, ..., r^\star\}$ is the set of robots and P_0 is the set of their initial positions, is given by,*

$$r^\star = \min_r \{\mathcal{F}_r \neq \emptyset\} = \max\{r_i | i \in \{1, ..., n\}\} \qquad (16)$$

Proof : Let us assume that at time instant t_i, the minimum number of robots required is indeed the maximum over all time instants in the *Score*, i.e., $r_i = r^\star$. Thus, if the total number of robots r is less than $r^\star$, then at least one of the conditions (a) or (b) in Theorem 3 would never be met, thus resulting in $\mathcal{F}_r = \emptyset$.

Conversely, if the total number of robots r is at least equal to $r^\star$, then both conditions (a) or (b) in Theorem 3 can be met, thereby proving that $\mathcal{F}_r \neq \emptyset$. ■

Calculating the minimum number of robots

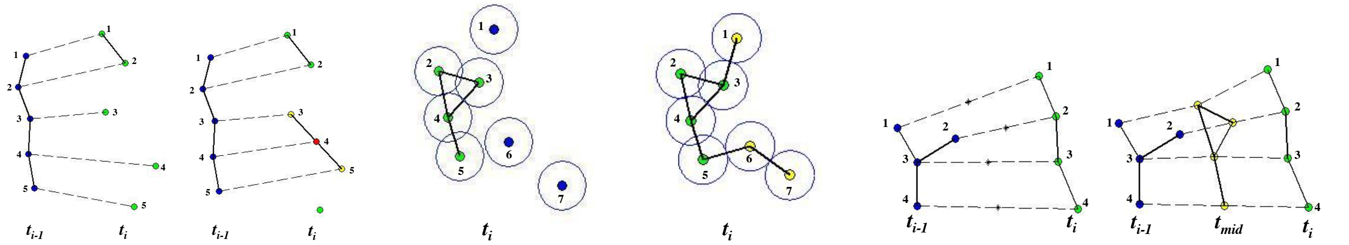
From Theorem 4, it is clear that solving the *STP-MSPBEL* is impertinent to finding $r^\star$. However, the *STP-MSPBEL* is proven to be NP-hard (see Lin and Xue [1999]). Thus, Theorems 4 and 5 provide *theoretical* results on global optimality. To calculate $r^\star$ in practical scenarios, one can use many existing algorithms with varying time complexities and performance ratios that provide an approximation to the *STP-MSPBEL* (see Chen et al. [2000] and Cheng et al. [2008]). For instance, an $\mathcal{O}(N^3)$ time approximate algorithm with performance ratio of at most 3, is presented in Cheng et al. [2008], where N denotes the number of planar positions given in the *STP-MSPBEL*. Another example is the approximate algorithm obtained by the minimum spanning tree (see Lin and Xue [1999]).

3.2 Generating Trajectories

In this section, we go from existence results to the generation of actual *feasible* trajectories. In order to achieve this, one can associate a cost with the trajectories, for example, the total length traveled. Or more precisely,

Given (Sc, R, P_0, Δ), where it is assumed that we have enough robots to ensure the existence of a set of *feasible* trajectories ($r \geq r^\star$), the objective would be to generate $X \in \mathcal{F}_r$ such that the following function is minimized,

$$\sum_{p \in R} \int_{t_0}^{t_n} \sqrt{\dot{x}_{p1}^2 + \dot{x}_{p2}^2}\, \mathrm{d}t \qquad (17)$$



(a) *Assign :* Robots are reassigned to positions at t_i, such that all positions in S_i are occupied (S_i contains the timed positions at t_i (two yellow points) and the added steiner point (red point))

(b) *Connect :* Robots 1, 6 and 7 merge with the connected backbone comprising of robots 2, 3, 4 and 5 (circular halos depict the range $\frac{\Delta}{2}$)

(c) *Mid-Config :* Intermediate positions of the robots are found (yellow points), and the dashed lines represent the resulting connectivity preserving piecewise linear trajectories over the interval (t_{i-1}, t_i)

Fig. 3. Example scenarios (before and after) of the sub-algorithms used in the *Trajectories* algorithm

For convenience, we refer to the Connectivity Constrained Routing Problem, stated above, as the *CCRP*.

Due to the high dimensionality of the multi-robot configuration space, finding a global solution i.e. a set of minimum length trajectories that are *feasible*, is typically not an option. As a result, we relax the requirement for global optimality and instead, propose the *Trajectories* algorithm that guarantees convergence to a sub-optimal solution. The main idea behind this algorithm is to apply the framework of assignment problems towards finding a solution.

In order to generate trajectories for the *CCRP*, we revisit the *URP* discussed previously in this paper. In particular, we use the result from Theorem 1 that uses the *Hungarian Method* to find an *optimal assignment* for a given triple (Sc, R, P_0), and consequently, the trajectories of every robot by linearly interpolating between successive pairs of assigned timed positions in increasing order of specified time instants. We let $X^b \in \mathcal{C}_r$ denote the set of such trajectories.

3.2.1 The Trajectories Algorithm

For a given (Sc, R, P_0), the *Trajectories* algorithm calculates the positions of the robots at *every* time instant in the *Score*, using the *optimal assignment* that solves the URP, i.e., the algorithm calculates $X^b(t_i)$ for all $i \in \{1, ..., n\}$. Using these positions as an *initial* estimate, and for a given Δ, the algorithm (inspired by Theorem 3) uses the *Connect* sub-algorithm to modify these positions in a manner that ensures that the induced proximity graph at every time instant in the *Score* is connected. As a result, conditions (a) and (b) of Theorem 3 are satisfied. Moreover, the algorithm uses the *Assign* sub-algorithm (which essentially solves an assignment problem) to reassign robots from their positions at a particular time instant to positions at the next (successive) time instant in the *Score*. Finally, the algorithm uses the *Mid-Config* sub-algorithm to find *connectivity preserving* motions between sets of such robot positions, specified at successive time instants, thus generating a set of *feasible* sub-optimal piecewise robot trajectories.

Algorithm 1 *Trajectories* (Sc, R, P_0, Δ)

```
$X \leftarrow X^b$, where $X^b(t_0) = P_0$ {$X$ is initially equal to $X^b$}
for $i = 1$ to $n$ do {iterating over all time instants in the $Score$}
  if $G(X(t_i), \Delta)$ is connected then
    if $G(X^b(t_{i-1}), \Delta)$ is not connected then {initial estimates of the robot positions at $t_{i-1}$ required modification}
      $H \leftarrow$ Assign $(X(t_{i-1}), X(t_i), \emptyset)$
      Using $H : X(t_{i-1}) \rightarrow X(t_i)$, update $X(t_i)$ such that the current position of robot $p$ is given by $x_p(t_i) = H(x_p(t_{i-1})) = x_q^b(t_i)$, where $p, q \in R$ {At $t_i$, robot $p$ assumes the position originally occupied by robot $q$}
      Update $X(t_j), j \in \{i+1, ..., n\}$ such that robot $p$ assumes all positions originally occupied by robot $q$, at all future Score time instants, i.e. $x_p(t_j) = x_q^b(t_j)$ $\forall j \in \{i+1, ..., n\}$
    end if
  else {$G(X(t_i), \Delta)$ is not connected}
    Find $S_i$, i.e. the positions of $r_i$ robots at $t_i$, obtained by approximately solving the STP-MSPBEL (see Cheng et al. [2008])
    if $S_i \neq P_i$ then {$P_i$ does not induce a connected proximity graph, i.e. steiner points are added at $t_i$}
      $H \leftarrow$ Assign $(X(t_{i-1}), S_i, X(t_i) \setminus P_i)$ {e.g. Figure 3a}
      Using $H : X(t_{i-1}) \rightarrow S_i \cup X(t_i) \setminus P_i$, update $X(t_i)$ such that the current position of robot $p$ is given by $x_p(t_i) = H(x_p(t_{i-1}))$
    end if
    $X(t_i) \leftarrow$ Connect $(S_i, X(t_i) \setminus S_i, \Delta)$ {e.g. Figure3b}
    $H \leftarrow$ Assign $(X(t_{i-1}), X(t_i), \emptyset)$
    Using $H : X(t_{i-1}) \rightarrow X(t_i)$, update $X(t_i)$ such that the current position of robot $p$ is given by $x_p(t_i) = H(x_p(t_{i-1}))$
  end if
  $X(t_{mid}) \leftarrow$ Mid-Config $(X(t_{i-1}), X(t_i), \Delta)$, $t_{mid} \in (t_{i-1}, t_i)$ {e.g. Figure 3c}
  $X(t) \leftarrow$ linear interpolation between $X(t_{i-1}), X(t_{mid})$ and $X(t_i)$, $t \in (t_{i-1}, t_i)$
end for
return $X$
```

Assign (A, B, C)

Given $A = \{a_1, ..., a_{|A|}\}$ specified at time instant t_{i-1}, and $B = \{b_1, ..., b_{|B|}\}$ and $C = \{c_1, ..., c_{|C|}\}$, both specified at time instant t_i, as three sets of planar positions each, where $|A| \leq |B| + |C|$. Let the cost of assigning a position in A to a position in $B \cup C$ equal the distance between the two positions. The idea is to assign every position in A to a unique position in $B \cup C$, such that all positions in B are assigned, positions in C may or may not be assigned, and the total cost of assignment is minimized. In essence, the *Assign* sub-algorithm solves an unbalanced linear sum assignment problem (see Martello and Toth [1987]).

By defining $\hat{B} \triangleq \{b_1, ..., b_{|B|}, c_1, ..., c_{|C|}\} \triangleq \{\hat{b}_1, ..., \hat{b}_{|B|+|C|}\}$, we can describe the assignment problem as a linear program,

$$\min_{l} \sum_{\alpha=1}^{|A|} \sum_{\beta=1}^{|B|+|C|} ||\hat{b}_\beta - a_\alpha|| \, l(\alpha, \beta) \tag{18}$$

subject to:

$$l(\alpha, \beta) \in \{0, 1\} \tag{19}$$

$$\sum_{\alpha=1}^{|A|} l(\alpha, \beta) = 1, \forall \beta \in \{1, ..., |B|\} \tag{20}$$

$$\sum_{\alpha=1}^{|A|} l(\alpha, \beta) \leq 1, \forall \beta \in \{|B| + 1, ..., |B| + |C|\} \tag{21}$$

$$\sum_{\beta=1}^{|B|+|C|} l(\alpha, \beta) = 1, \forall \alpha \in \{1, ..., |A|\} \tag{22}$$

where $l(\alpha, \beta)$ is 1 if $a_\alpha \in A$ is assigned to $\hat{b}_\beta \in \hat{B}$, and 0 otherwise.

Algorithm 2 *Assign* (A, B, C)

```
$\hat{B} \triangleq \{b_1, ..., b_{|B|}, c_1, ..., c_{|C|}\} \triangleq \{\hat{b}_1, ..., \hat{b}_{|B|+|C|}\}$
Find $l$ that solves Equations (18)-(22)
Find $H : A \to \hat{B}$ such that $l(\alpha, \beta) = 1 \iff H(a_\alpha) = \hat{b}_\beta$,
  $\forall \alpha \in \{1, ..., |A|\}, \beta \in \{1, ..., |B| + |C|\}$
return $H$
```

Connect (A, B, Δ)

Given $A = \{a_1, ..., a_{|A|}\}$ and $B = \{b_1, ..., b_{|B|}\}$, both specified at time instant t_i, as two sets of planar positions each, where the induced graph $G(A, \Delta)$ is connected, i.e. positions in A form a connected backbone, while B contains positions that may or may not be connected to this backbone. The idea is to "grow" this connected backbone by recursively adding to A, updated positions from B such that the updated $G(A, \Delta)$ becomes connected. The algorithm returns this connected backbone A.

Algorithm 3 *Connect* (A, B, Δ)

```
repeat
    Find $\alpha^\star \in \{1, ..., |A|\}, \beta^\star \in \{1, ..., |B|\}$ such that $||a_{\alpha^\star} - b_{\beta^\star}|| = \min(||a_\alpha - b_\beta||) \, \forall \alpha, \beta$
    if $||a_{\alpha^\star} - b_{\beta^\star}|| > \Delta$ then
        $b_{\beta^\star} \leftarrow a_{\alpha^\star} + \frac{b_{\beta^\star} - a_{\alpha^\star}}{||a_{\alpha^\star} - b_{\beta^\star}||} \Delta$
    end if
    $A \leftarrow A \cup \{b_{\beta^\star}\}$
    $B \leftarrow B \setminus \{b_{\beta^\star}\}$
until $B = \emptyset$
return $A$
```

Mid-Config (A, B, Δ)

Given $A = \{a_1, ..., a_{|A|}\}$ specified at time instant t_{i-1}, and $B = \{b_1, ..., b_{|B|}\}$ specified at time instant t_i, as two sets of planar positions each, where $|A| = |B|$ and the induced graphs $G(A, \Delta)$ and $G(B, \Delta)$ are both connected. The idea is to find an equal sized set of *intermediate* planar positions $M = \{m_1, ..., m_{|M|}\}$, specified at some time instant $t_{mid} \in (t_{i-1}, t_i)$, such that the induced proximity graph $G(M, \Delta)$ contains the edges of the *spanning trees* of *both* $G(A, \Delta)$ and $G(B, \Delta)$ (Notice that $G(M, \Delta)$ is connected by definition). Consequently, the set of piecewise linear trajectories formed by linearly interpolating between A, M and B, is guaranteed to ensure a connected proximity graph for all times $t \in (t_{i-1}, t_i)$.

Moreover, let the mid points of each unconstrained straight line path between corresponding positions a_α and b_α, $\alpha \in \{1, ..., |A|\}$ be the so-called *target* points for corresponding planar positions in M. Let $C = \{c_\alpha \,|\, c_\alpha = \frac{a_\alpha + b_\alpha}{2}, \alpha \in \{1, ..., |A|\}\}$ denote the set of these *target* points. The sub-algorithm *Mid-Config* then solves the following constrained optimization problem,

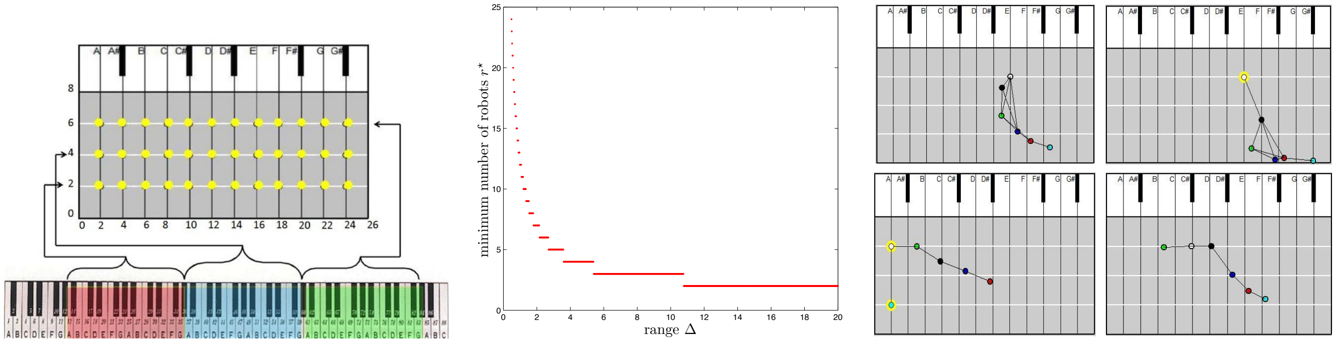
$$\min_{M} \sum_{\alpha=1}^{|A|} ||m_\alpha - c_\alpha|| \tag{23}$$

such that $G(M, \Delta)$ contains the edges of the spanning trees of $G(A, \Delta)$ and $G(B, \Delta)$.

Algorithm 4 *Mid-Config* (A, B, Δ)

```
$C \triangleq \{c_\alpha \,|\, c_\alpha = \frac{a_\alpha + b_\alpha}{2}, \alpha \in \{1, ..., |A|\}\}$
$G_s(A, \Delta) \leftarrow$ euclidean min span tree of $G(A, \Delta)$
$G_s(B, \Delta) \leftarrow$ euclidean min span tree of $G(B, \Delta)$
Find $M$ by solving Equation 23 such that $G(M, \Delta)$ contains
  the edges of $G_s(A, \Delta)$ and $G_s(B, \Delta)$
return $M$
```

Computational Complexity : For a given (Sc, R, P_0, Δ), the *Trajectories* algorithm initially solves the *URP* by employing the *Hungarian Method*, the complexity of which is $O(|Sc|^3)$. Additionally, it computes piecewise linear trajectories using the various sub-algorithms, by iterating over n time instants,



(a) A simulated *Piano Wall* with 36 coordinates (light colored points) representing the notes across three octaves of a piano

(b) A plot of the minimum number of robots ($r^\star$) versus the range (Δ)

(c) four snapshots of the connectivity constrained case: $\Delta = 4$, $r^\star = 4$

Fig. 4. Simulation results

with the complexity of each iteration being $O(r^3)$. Thus, the overall complexity of the algorithm is $O(|Sc|^3 + nr^3)$.

Note: Since the optimization criterion we consider is the total length travelled, every time an assignment problem is solved in the *Trajectories* algorithm, i.e. a one to one correspondence is found between two sets of positions, with minimum total length travelled, we get corresponding paths between the sets of positions, that *do not intersect* (e.g. on the initial step that finds the *optimal assignment* for solving the corresponding *URP*, and on every call of the *Assign* sub-algorithm). Though we do not provide a formal analysis, one can build on such a notion to generate robot trajectories that avoid collisions.

3.2.2 *Optimizing Total Length : A Discussion*

In this section, we highlight various design characteristics targeted towards optimizing the total length of robot trajectories. To begin with, the *Trajectories* algorithm uses *optimal* positions of robots, obtained by solving the *URP*, as initial estimates for finding a sub-optimal solution to the *CCRP*. Moreover, the *Connect* sub-algorithm recursively moves each disconnected robot by a minimal distance, in order to merge it with a connected backbone at a particular time instant in the *Score*. The *Assign* sub-algorithm reassigns robots from their positions at one time instant in the *Score* to the next, such that the total length of the corresponding straight line robot trajectories between assigned positions is minimum, thereby providing a good base for the *Mid-Config* sub-algorithm. In turn, the *Mid-Config* sub-algorithm finds intermediate robot positions that cause a minimum deviation between the the original straight line trajectories and the resulting piecewise linear ones, while satisfying the edge constraints on the induced proximity graph.

As discussed before, finding an optimal solution to the *CCRP* is typically not feasible, since searching the space for all possible coordinated paths for multiple robots requires a prohibitive amount of computation time and memory (for instance, the general optimal motion planning problem for multiple robots is P-Space hard (see Hopcroft et al. [1984])). However, we attempt to characterize ‘how” sub-optimal the solution obtained from the *Trajectories* algorithm is, in the following (designed) instance of the *CCRP* where we do, in fact, know the optimal solution: For a given (Sc, R, P_0), we solve the Unconstrained Routing Problem (URP) using the *Hungarian Method*. Since the solution to the URP provides globally optimal robot trajectories, we calculate the minimum sensing range Δ_{free} that ensures a connected proximity graph for all times, given the robot trajectories. Thus, for the quadruple $(Sc, R, P_0, \Delta_{free})$, these trajectories provide an optimal solution to the corresponding Connectivity Constrained version of the problem ($CCRP$). Now, by running the *Trajectories* algorithm on the same quadruple $(Sc, R, P_0, \Delta_{free})$, we obtain a sub-optimal solution to the $CCRP$, that can be compared to the optimal solution computed earlier.

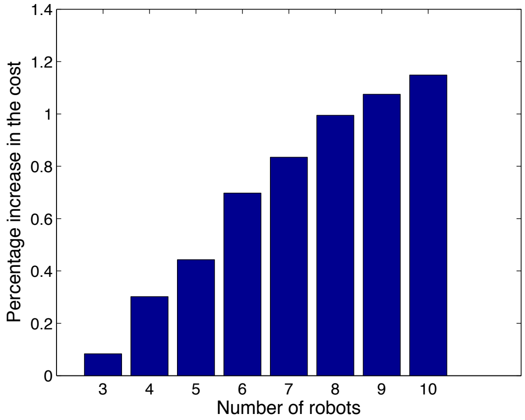


Fig. 5. Average percentage increase in the cost, calculated over 5000 *Scores*, randomly generated each time for a different r

Notice that through this approach, the optimal solution to the $CCRP$ is the very same that is computed by the *Trajec-*

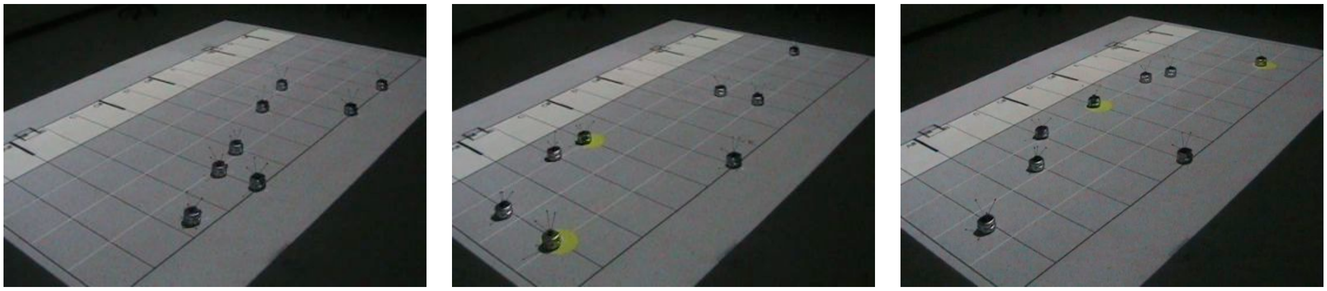


Fig. 6. Hardware Implementation
The complete video can be found at: `http://www.youtube.com/watch?v=YigAzrFoN3E/`

tories algorithm on line (1), as an initialization step. Consequently, the *Trajectories* algorithm never executes lines (10) - (17), and thus, the approximation algorithm that solves the *STP-MSPBEL* on line (10) does not affect the optimality of the solution. In fact, the main deviation from the optimal solution is caused by the *Mid-Config* sub-algorithm, which we characterize through multiple simulation experiments. We acknowledge that this approach is targeting a simpler instance of the *CCRP*, but since the generalized problem is computationally intractable to solve, we hope to provide an insight into cases where we do know the optimal solution, and hence can make a sub-optimality comparison.

In particular, for a fixed number of robots, say r, we randomly generate r number of initial and final positions from a pre-defined uniform distribution. For such a "one-beat" *Score*, we calculate Δ_{free} that allows the *Hungarian Method* to provide an optimal solution to the corresponding $CCRP$. Moreover, we generate the sub-optimal solution through the *Trajectories* algorithm, and calculate the percentage increase in the cost, i.e. the total length travelled, with respect to the optimal solution (see Figure 5). As can be seen, the increase in the deviation from the optimal cost (from $\approx 0.1\%$ to $\approx 1.1\%$), is small on the average across all cases.

4 Implementation

To demonstrate the musically inspired routing problems considered in this paper, we simulated an example of the *Robot Music Wall* in MATLAB, instrumented to sound like a piano (see Figure 4a). We called it the *Piano Wall*. Our goal was to make multiple robots perform the popular composition "Für Elise" by Ludwig van Beethoven on this *Piano Wall*. Note that firstly, all notes in the version of "Für Elise" we presented to the robots, lay amongst the set of notes used to create the *Piano Wall*. Secondly, a pianist was required to hit a maximum of two keys simultaneously throughout its performance ($\mathcal{K} = 2$). With this set-up, we created the *Score* associated with "Für Elise", containing timed positions on the wall corresponding to notes in "Für Elise".

We simulated the robots as 2-d multi-colored circular points on the *Piano Wall*. In our program, the instant a robot reached an assigned timed position on the wall, it was encircled by a light circle (yellow), and the sound of the corresponding piano note was generated. For the Connectivity Constrained Routing Problem (*CCRP*) discussed in Section 3, we chose different values of Δ, and calculated the corresponding minimum number of robots $r^\star$ (see Figure 4b). Then, for different number of robots $r \geq r^\star$ (given some Δ), we constructed the routes for every robot, using the *Trajectories* algorithm (see Figure 4c). These routes/paths were executed by the robots with appropriate velocities that ensured their timely arrival at assigned positions. In addition to MATLAB simulations, we carried out hardware implementations (see Figure 6) to demonstrate the routing problems in this paper.

5 Conclusions

We consider *spatio-temporal* multi-robot routing, where multiple robots are required to service a set of spatially distributed requests at specified time instants. We show that such a routing problem can be formulated as an assignment problem, solvable in polynomial time, and provide the *Robot Music Wall* as a motivating example, to illustrate the musical inspiration behind the problem. We incorporate connectivity maintenance into *spatio-temporal* routing, and discuss the feasibility aspects of the ensuing problem. Additionally, we develop an algorithm for generating explicit trajectories for the robots, and demonstrate our results through MATLAB simulations and hardware implementations, using the *Robot Music Wall* as the basic set-up.

References

A. Arsie, K. Savla, and E. Frazzoli. Efficient routing algorithms for multiple vehicles with no explicit communications. *IEEE Transactions on Automatic Control*, 54(10):2302 –2317, 2009.

T. Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219, 2006.

F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482 –1504, 2011.

W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376 – 378, 2005.

R. Burkard, M. Dell'Amico, and S. Martello. *Assignment problems*. Society for Industrial and Applied Mathematics, 2009.

D. Chen, D. Du, X. Hu, G. Lin, L. Wang, and G. Xue. Approximations for steiner trees with minimum number of steiner points. *Journal of Global Optimization*, 18:17–33, 2000.

X. Cheng, D. Du, L. Wang, and B. Xu. Relay sensor placement in wireless sensor networks. *Wireless Networks*, 14:347–355, 2008.

S. Chopra and M. Egerstedt. Multi-robot routing under connectivity constraints. *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2012a.

S. Chopra and M. Egerstedt. Multi-robot routing for servicing spatio-temporal requests: A musically inspired problem. *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*, 2012b.

C. Dixon and E. Frew. Maintaining optimal communication chains in robotic sensor networks using mobility control. *Mobile Networks and Applications*, 14:281–291, 2009.

John Hopcroft, Jacob Schwartz, and Micha Sharir. On the complexity of motion planning for multiple independent objects; pspace-hardness of the" warehouseman's problem". *The International Journal of Robotics Research*, 3(4):76–88, 1984.

Z. Kan, A. Dani, J. Shea, and W. Dixon. Information flow based connectivity maintenance of a multi-agent system during formation control. *Proceedings of the IEEE Conference on Decision and Control*, pages 2375 –2380, 2011.

R. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics*, 2(1-2):83–97, 1955.

E. Lawler. *Combinatorial optimization: Networks and matroids*. Holt, Rinehart and Winston, New York, 1976.

G. Lin and G. Xue. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Information Processing Letters*, 69(2):53 – 57, 1999.

S. Martello and P. Toth. Linear assignment problems. *Annals of Discrete Mathematics*, 31:259–282, 1987.

A. Mosteo, L. Montano, and M. Lagoudakis. Multi-robot routing under limited communication range. *IEEE International Conference on Robotics and Automation*, pages 1531 –1536, 2008.

H. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J. Spector. Autonomous communication relays for tactical robots. *Proceedings of the 11th International Conference on Advanced Robotics*, 2003.

M. Pavone and E. Frazzoli. Dynamic vehicle routing with stochastic time constraints. *IEEE International Conference on Robotics and Automation*, pages 1460 –1467, 2010.

S. Ponda, L. Johnson, H. Choi, and J. How. Ensuring network connectivity for decentralized planning in dynamic environments. *AIAA Aerospace Information Technologies Conference*, 2011.

D. Spanos and R. Murray. Motion planning with wireless network constraints. *Proceedings of the American Control Conference*, pages 87 – 92, 2005.