Multi-robot routing for servicing spatio-temporal requests: A musically inspired problem

Smriti Chopra * **Magnus Egerstedt** *

* *Georgia Institute of Technology, Atlanta, GA 30332 USA*
(e-mail: {smriti.chopra, magnus}@ece.gatech.edu)

Abstract: In this paper, we consider the problem of routing multiple robots to service spatially distributed requests at specified time instants. As a motivating example, we present the *Robot Music Wall*, a musically instrumented surface where planar positions correspond to distinct notes of an instrument. Multiple robots with the ability to traverse the wall can effectively "play" a piece of music by reaching positions on the wall that correspond to the musical notes in the piece, at specified time instants. We show that the multi-robot routing problem for servicing such spatio-temporal requests can be formulated as a pure assignment problem with the resulting reduction in complexity. Moreover, we derive the minimum number of robots required to service such requests.

Keywords: assignment problems; autonomous mobile robots; optimization problems; vehicle routing.

1. INTRODUCTION

Multi-robot routing is an important, well researched topic in robotics. It 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 travelled, 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. Our problem is musically inspired in that it requires multiple robots to reach a series of planar positions at specified time instants, much like a musician that uses multiple fingers to play a series of notes on an instrument at specified time instants. We show that such a problem can be formulated as a pure assignment problem, solvable in polynomial time. Moreover, we derive the minimum number of robots required to solve it.

Many well known problems in combinatorial optimization can be associated with multi-robot routing, as seen in the field of operations research and theoretical computer science. For instance, the multiple travelling 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 the well known vehicle routing problem (VRP) (see Bodin et al. (1983) and Arsie et al. (2009)), which is a constrained version of the m-TSP. The VRP 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. (2010), Smith et al. (2010), and Pavone and Frazzoli (2010)). Variations like the capacitated VRP (see Ralphs (2003)) and VRP with time windows (see Solomon (1987)) incorporate practical constraints like vehicle capacities and timely deliveries.

Applications of multi-robot routing include surveillance, search and rescue, transportation on demand, and assembly. However, to solve such 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. We show that by encorporating such constraints in servicing 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.

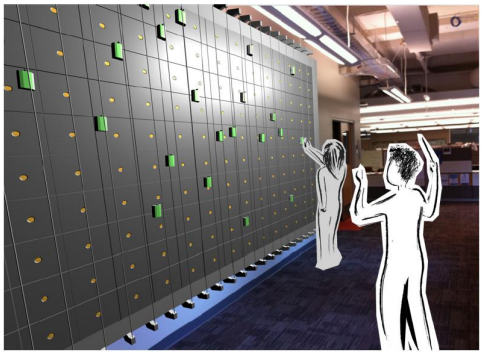

Fig. 1. A rendering of the *Robot Music Wall* concept

2. 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, as illustrated in Figure 1. 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. In other words, we have a musically instrumented wall where a robot can effectively "play" a musical note by reaching its corresponding position on the wall and plucking the string above it.

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. Moreover, we might even require that multiple positions are reached simultaneously, akin to a musician that has to play multiple notes of an instrument simultaneously with different fingers. By routing multiple robots to service such timed positions, we can effectively "play" the piece of music associated with them on the wall. Note that we consider a timed position "serviced" the instant it is reached by a robot (i.e. we neglect on-site servicing time).

3. PROBLEM DEFINITION

We let $t_1, t_2, ..., t_n$ denote 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 $|.|$ 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 some initial 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 travelled 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 $\dot{x}_i = u_i$. 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.

Now, suppose we define a function $A : R \to 2^{Sc}$ that maps the robots in R to sets of timed positions in the *Score*. We call this function a *feasible mapping* if it satisfies the following conditions,

$$\bigcup_{p \in R} A(p) = Sc \tag{4}$$

$$A(p) \cap A(q) = \emptyset \quad \forall p, q \in R, p \neq q \tag{5}$$

$$(P_{i,\alpha}, t_i), (P_{j,\beta}, t_j) \in A(p) \Rightarrow i \neq j \quad if\, \alpha \neq \beta \tag{6}$$

Equation (4) states that every timed position in the *Score* is assigned (or in context to the music wall, every note is played). Equation (5) states that no two robots are assigned the same timed position (i.e. no two robots play the same note at the same time). Equation (6) states that a robot is not assigned more than one position at a given time instant (i.e. a robot does not play more than one note at a given time instant).

Given such a *feasible mapping*, the path of every robot is determined by its assigned set of timed positions, traversed in increasing order of specified time instants. Since robots move between assigned positions in straight line trajectories, the total distance traversed by all robots can be determined. Moreover, if this total distance is minimum, then we call such a mapping an *optimal mapping*, denoted by $A^\star$.

Furthermore, 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 $A^\star$ 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. We denote this by,

$$As = \{(P_{i,\alpha}, t_i) \,|\, i \in \{0, ..., n-1\}, \alpha \in \{1, ..., |P_i|\}\} \tag{7}$$

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

We let $\pi : As \to Sc$ 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 $\pi|_{As'} : As' \to Sc$ 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 travelled 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.

Hence, for a given (Sc, R, P_0), we see that the *optimal assignment* $\pi^\star$ encodes all information contained in the corresponding *optimal mapping* $A^\star$. To express $A^\star$ in terms of $\pi^\star$, it will be convenient to apply $\pi^\star$ repeatedly. We use the following notatation to express this fact,

$$(\pi^\star)^0(a) = a$$
$$(\pi^\star)^k(a) = (\pi^\star(\pi^\star)^{k-1}(a))$$

where $k \in \mathbb{N}$, $a \in As'$.

Note that for a robot $p \in R$, if its initial timed position $(P_{0,p}, t_0)$ is not in As', then clearly, it is not assigned to any timed position in the *Score*. Consequently, the path of p

in the corresponding *optimal mapping* $A^\star$ is an empty set. However, if the initial position of p is indeed in As', then we can determine its corresponding path by applying $\pi^\star$ repeatedly on $(P_{0,p}, t_0)$. The number of times we must apply $\pi^\star$ is given by $k_p \in \{1, ..., |Sc|\}$, where $(\pi^\star)^{k_p-1}((P_{0,p}, t_0)) \in As' \wedge (\pi^\star)^{k_p}((P_{0,p}, t_0)) \notin As'$. Here, $(\pi^\star)^{k_p}$ denotes the terminal timed position in the path of robot p. Thus, we can construct $A^\star$ as follows,

$$A^\star(p) = \begin{cases} \bigcup_{k=1}^{k_p} \{(\pi^\star)^k((P_{0,p}, t_0))\} & \text{if } (P_{0,p}, t_0) \in As' \\ \emptyset & \text{otherwise} \end{cases}, \forall p \in R \quad (8)$$

Consequently, we can search for $\pi^\star$ rather than $A^\star$. Thus, in the remaining part of this paper, we focus on the problem of finding the *optimal assignment* $\pi^\star$ for a given triple (Sc, R, P_0).

3.1 Formal definition of the problem

Given (Sc, R, P_0), and a cost function $\mathcal{C} : (As \times Sc) \to \mathbb{R}$ where $\mathcal{C}((P_{i,\alpha}, t_i) \in As, (P_{j,\beta}, t_j) \in Sc) = ||P_{j,\beta} - P_{i,\alpha}||$, i.e. the cost of assigning a timed position in As to a timed position in Sc equals the distance between the two positions, the objective is to find a *feasible assignment* $\pi|_{As'} : As' \to Sc$, $As' \subseteq As$ such that the following function is minimized,

$$\sum_{(P_{i,\alpha}, t_i) \in As'} \mathcal{C}((P_{i,\alpha}, t_i), \pi((P_{i,\alpha}, t_i)) \quad (9)$$

Equation (9) represents the total distance associated with the individual assignments in $\pi|_{As'}$. Hence, the resulting assignment is an *optimal assignment*, denoted by $\pi^\star$.

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

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

$$\min_l \sum_{i \in \mathcal{I}} \sum_{\alpha \in \mathcal{A}_i} \sum_{j=i+1}^{n} \sum_{\beta \in \mathcal{A}_j} ||P_{j,\beta} - P_{i,\alpha}|| \, l(i, \alpha, j, \beta) \quad (10)$$

subject to:

$$l(i, \alpha, j, \beta) \in \{0, 1\} \quad (11)$$

$$\sum_{i=0}^{j-1} \sum_{\alpha \in \mathcal{A}_i} l(i, \alpha, j, \beta) = 1, \forall \, j \in \mathcal{J}, \beta \in \mathcal{A}_j \quad (12)$$

$$\sum_{j=i+1}^{n} \sum_{\beta \in \mathcal{A}_j} l(i, \alpha, j, \beta) \leq 1, \forall \, i \in \mathcal{I}, \alpha \in \mathcal{A}_i \quad (13)$$

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* $\pi^\star : As \to Sc$, where $l(i, \alpha, j, \beta) = 1 \iff \pi^\star((P_{i,\alpha}, t_i)) = (P_{j,\beta}, t_j)$. Equations (12) and (13) ensure *feasibility* of this assignment, while (10) ensures that the total distance associated with the individual assignments is minimum.

In combination with the distance travelled, one can also associate with $\pi^\star$, the maximum velocity required by the robots to execute the assigned trajectories. This will become important as we, later on, cap the velocities of the individual robots.

Definition 3. The velocity of an *optimal assignment* $\pi^\star$, given by $V(\pi^\star)$ is,

$$V(\pi^\star) = \left\{ \max_{\substack{i \in \mathcal{I}, \alpha \in \mathcal{A}_i \\ j \in \mathcal{J}, \beta \in \mathcal{A}_j}} \left\{ \frac{||P_{j,\beta} - P_{i,\alpha}||}{t_j - t_i} \, l(i, \alpha, j, \beta) \right\} \right\} \quad (14)$$

4. ASSIGNMENT METHOD

The problem in (3.1) is a modified version of the classic *linear sum assignment problem* (*LSAP*) (see Derigs (1985) and Martello and Toth (1987)) that concerns the following: given two equal sized sets P and Q with some non-negative cost function $\mathcal{C} : (P \times Q) \to \mathbb{R}$, the objective is to find a complete assignment, i.e. a bijection $S : P \to Q$ that minimizes the function $\sum_{a \in P} \mathcal{C}(a, S(a))$. As and Sc in (3.1) correspond to P and Q in the *LSAP*, and the *feasible assignment* π corresponds to S. Note that the problem in (3.1) is a slight modification of the *LSAP*, since the *LSAP* insists on P and Q being equal sized while (3.1) insists on $|As| \geq |Sc|$. Moreover, in the *LSAP*, there exist no forbidden individual assignments between P and Q, contrary to (3.1), 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 problem in (3.1) by incorporating certain modifications that we discuss later in this section.

Many algorithms, both sequential and parallel, have been developed for solving the *LSAP* (see survey paper by 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)).

The *Hungarian Method* operates on a square cost matrix $C = [c_{i,j}]$ generated from the given cost function $\mathcal{C} : (P \times Q) \to \mathbb{R}$, where $c_{i,j}$ is the cost of assigning the i^{th} element in P to the j^{th} element in Q. In order to use the *Hungarian Method* towards solving the problem in (3.1), we generate a similar cost matrix from the cost function $\mathcal{C} : (As \times Sc) \to \mathbb{R}$, by constructing a matrix C of size $|As| \times |Sc|$, where the rows and columns in C represent timed positions in As and Sc respectively. We partition C into n^2 blocks, where a block is denoted by $C_{i,j}$, $i \in \mathcal{I}$, $j \in \mathcal{J}$. $C_{i,j}$ is in turn a submatrix of size $|\mathcal{A}_i| \times |\mathcal{A}_j|$, where the rows in $C_{i,j}$ represent timed positions specified at t_i in As and columns represent timed positions specified at t_j in Sc. Now that the structure of C is established, we denote an element in C by c_{i_α, j_β}, where c_{i_α, j_β} denotes the element (α, β), $\alpha \in \mathcal{A}_i$, $\beta \in \mathcal{A}_j$, in the block $C_{i,j}$ in C. Moreover, we let $c_{i_\alpha, j_\beta} = \mathcal{C}((P_{i,\alpha}, t_i) \in As, (P_{j,\beta}, t_j) \in Sc)$. We call C the cost matrix, and denote it by $C = [c_{i_\alpha, j_\beta}]$.

Notice that C is not a square matrix. Hence, in order to use the *Hungarian Method* towards solving the problem in (3.1), we need to incorporate some changes in C. We introduce $(|As| - |Sc|)$ *dummy* positions as targets (in addition to the timed positions in the *Score*) in order to create a square cost matrix. For convenience, we assume that these *dummy* positions are specified at time instant t_{n+1}. We denote the set of such *dummy* positions by $P_{n+1} = \{P_{n+1,\beta} \mid \beta \in \mathcal{A}_{n+1}\}$ where $\mathcal{A}_{n+1} \triangleq \{1, 2, ..., (|As| - |Sc|)\}$. Moreover, we let the cost associated

with reaching these *dummy* positions be zero. In other words, we append $(|As| - |Sc|)$ number of columns with zero cost to the cost matrix, thereby creating a square cost matrix. 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 problem in (3.1), 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 for example Burkard et al. (2009)). We denote this modified cost matrix by $\hat{C} = [\hat{c}_{i_\alpha, j_\beta}]$, where,

$$\hat{c}_{i_\alpha, j_\beta} = \begin{cases} ||P_{j,\beta} - P_{i,\alpha}||, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i, \\ & j \in \{i+1, ..., n\}, \beta \in \mathcal{A}_j \\ M, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i, \\ & j \in \{1, ..., i\}, \beta \in \mathcal{A}_j \\ 0, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i, \\ & j = n+1, \beta \in \mathcal{A}_j \end{cases} \tag{15}$$

The symbol $\star$ represents valid assignments, while M represents forbidden ones. 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. And, as will be shown in subsequent paragraphs, such an assignment does in fact exist. We denote the assignment by $H_r : As \to Sc'$, where r refers to the number of robots.

4.1 Existence of a solution

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 problem in (3.1), the *Hungarian Method* operates on its associated cost matrix $\hat{C}$, given by (15), to produce an *optimal assignment* $H_r^\star$.

Proof. See Appendix A.

Corollary 1.

$$cost(H_r^\star) = cost(H_r) \leq trace(\hat{C})$$

Proof. In the proof of Theorem 1, we introduce a bijective function $\omega_r : As \to Sc'$ that contains individual assignments between all timed positions in As and Sc' that have a corresponding diagonal cost entry in the cost matrix $\hat{C}$. Moreover, we show that ω_r is a complete assignment that avoids forbidden individual assignments. Notice that $cost(\omega_r) = trace(\hat{C})$. Also, from Theorem 1, we know that the *Hungarian Method* produces a complete assignment H_r that avoids forbidden individual assignments, which in turn produces the *optimal assignment* $H_r^\star$ (after removing assignments to *dummy* positions). Since the cost associated with reaching *dummy* positions is zero, $cost(H_r^\star) = cost(H_r)$. Moreover, since the *Hungarian Method* produces an assignment with minimum cost, no other assignment can have a lower associated cost. Thus, $cost(H_r^\star) = cost(H_r) \leq cost(\omega_r) = trace(\hat{C})$. ∎

Theorem 2. The minimum number of robots, given by $r^\star$, required to ensure that a solution exists to the problem in (3.1) equals the maximum number of positions that require simultaneous servicing in the *Score* ($\mathcal{K}$), i.e.,

$$r^\star = \min_r \{Sc \subseteq range(H_r)\} = \mathcal{K} \tag{16}$$

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 problem in (3.1) that can be found through the *Hungarian Method*. Hence, $r^\star = \mathcal{K}$. ∎

5. CONSTRAINED VELOCITY CASE

So far, we have assumed that robots do not have any constraints on their velocities. However, a more realistic approach would be to introduce, say, a maximum velocity $\hat{v}$ that the robots cannot exceed while driving between assigned timed positions. We incorporate this constraint in the problem in (3.1) (unconstrained velocity case) to define its corresponding constrained velocity version as follows,

$$\min_L \sum_{i \in \mathcal{I}} \sum_{\alpha \in \mathcal{A}_i} \sum_{j=i+1}^{n} \sum_{\beta \in \mathcal{A}_j} ||P_{j,\beta} - P_{i,\alpha}||\, l(i, \alpha, j, \beta) \tag{17}$$

subject to:

$$l(i, \alpha, j, \beta) \in \{0, 1\} \tag{18}$$

$$\sum_{i=0}^{j-1} \sum_{\alpha \in \mathcal{A}_i} l(i, \alpha, j, \beta) = 1, \forall\, j \in \mathcal{J}, \beta \in \mathcal{A}_j \tag{19}$$

$$\sum_{j=i+1}^{n} \sum_{\beta \in \mathcal{A}_j} l(i, \alpha, j, \beta) \leq 1, \forall\, i \in \mathcal{I}, \alpha \in \mathcal{A}_i \tag{20}$$

$$\frac{||P_{j,\beta} - P_{i,\alpha}||}{t_j - t_i} < \hat{v} \iff l(i, \alpha, j, \beta) = 1 \tag{21}$$

Using a similar approach to before, we construct the associated cost matrix for the constrained velocity problem, denoted by

$$\hat{cv}_{i_\alpha, j_\beta} = \begin{cases} ||P_{j,\beta} - P_{i,\alpha}||, & \frac{||P_{j,\beta} - P_{i,\alpha}||}{t_j - t_i} \leq \hat{v}, \\ & i \in \mathcal{I}, \alpha \in \mathcal{A}_i \\ & j \in \{i+1, ..., n\}, \beta \in \mathcal{A}_j \\ M, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i \\ & j \in \{1, ..., i\}, \beta \in \mathcal{A}_j \\ 0, & i \in \mathcal{I}, \alpha \in \mathcal{A}_i \\ & j = n+1, \beta \in \mathcal{A}_j \\ M, & \frac{||P_{j,\beta} - P_{i,\alpha}||}{t_j - t_i} > \hat{v}, \\ & i \in \mathcal{I}, \alpha \in \mathcal{A}_i \\ & j \in \{i+1, ..., n\}, \beta \in \mathcal{A}_j \end{cases} \tag{22}$$

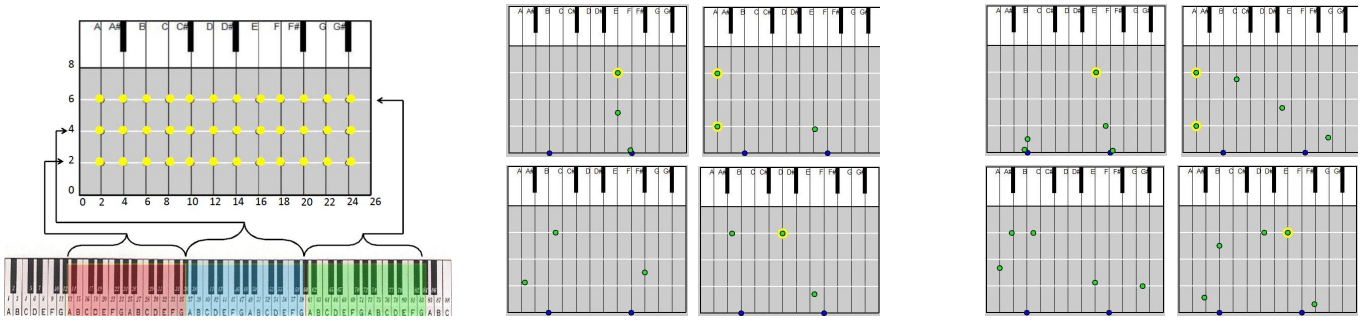

Fig. 2. A simulated *Piano Wall* with 36 coordinates (light colored points) representing the notes across three octaves of a piano (left), four snapshots of the unconstrained velocity case: $r = 3$, $r^\star = 2$ (center), and four snapshots of the constrained velocity case: $\hat{v} = 3$, $r = 2$, $r^\star = 5$ (right). Note that in the constrained velocity case, $r < r^\star$. Hence, $(r^\star - r) = 3$ additional robots were deployed from the base stations to ensure no timed position went unassigned

Note that we include individual assignments that not only violate the directionality constraint, but also the maximum velocity constraint, in the set of forbidden assignments. As always, we denote such assignments by M in the cost matrix.

Recall that if M is large enough, the *Hungarian Method* finds a complete assignment between As and Sc' that avoids forbidden individual assignments, if such an assignment exists. In the unconstrained velocity case, we saw that such an assignment does in fact exist. However, this is not guaranteed in the constrained velocity case, since there may be timed positions in Sc that are at distances so large from all timed positions in As that they simply cannot be reached without causing a robot to violate the given maximum velocity constraint.

In general, if the complete assignment at the termination of the *Hungarian Method* contains forbidden individual assignments, we deal with this by simply removing them from the complete assignment, thus obtaining an incomplete one. We let $a \subseteq As$ and $s \subseteq Sc'$ be the sets of timed positions in As and Sc' respectively that correspond to such forbidden individual assignments. Consequently, we denote the resulting assignment by the bijective function $\hat{H}_r : As \setminus a \to Sc' \setminus s$.

5.1 Existence of a solution

Similar to the unconstrained velocity case, all timed positions in the *Score* are reached, i.e. $Sc \subseteq range(\hat{H}_r)$, *if and only if* the *Hungarian Method* terminates with a complete assignment $\hat{H}_r : As \to Sc'$ that avoids forbidden individual assignments ($a = s = \emptyset$). Moreover, given such a complete assignment, we can construct an *optimal assignment* $\hat{H}_r|_{As'} : As' \to Sc$, where $As' \subseteq As$ is the set of all timed positions in As that are *not* assigned to *dummy* positions. We denote such an assignment by $\hat{H}_r^\star$.

Theorem 3. Given the constrained velocity problem, the *Hungarian Method* operates on its associated cost matrix $\hat{CV}$, given by (22), to produce an *optimal assignment* $\hat{H}_r^\star$ if the velocity of the *optimal assignment* $H_r^\star$ in the corresponding unconstrained velocity case does not violate the maximum velocity $\hat{v}$, i.e.,

$$V(H_r^\star) \leq \hat{v} \Rightarrow Sc \subseteq range(\hat{H}_r)$$

Proof. If the velocity of the *optimal assignment* $H_r^\star$ does not exceed the maximum velocity $\hat{v}$, it implies that none of the costs associated with the individual assignments in $H_r^\star$ equal M in the cost matrices for both unconstrained and constrained velocity cases. In other words, the *Hungarian Method* finds the exact same *optimal assignment* for both cases, i.e. $H_r^\star = \hat{H}_r^\star$, which in turn implies that $Sc \subseteq range(\hat{H}_r)$. ■

Note that for a given $\hat{v}$, the existence of $\hat{H}_r^\star$ depends on the intial positions of the robots. Thus, in order to remove this dependence, we assume the following,

Assumption 1. The starting position of every robot is chosen such that it can reach any timed position in the *Score* without violating the maximum velocity constraint.

Theorem 4. The minimum number of robots, denoted by $r^\star$, needed to ensure that a solution exists to the constrained velocity problem is given by,

$$r^\star = \min_r \{Sc \subseteq range(\hat{H}_r)\} = \mathcal{K} + |As| - |range(\hat{H}_\mathcal{K})|$$

where $\mathcal{K}$ equals the maximum number of positions that require simultaneous servicing in the *Score*.

Proof. Recall that we insist on $r \geq \mathcal{K}$. However, a solution to the constrained velocity problem is not guaranteed to exist with a minimum of $\mathcal{K}$ robots, since the number of timed positions in the *Score* that go unassigned depends on the maximum velocity $\hat{v}$ of the robots. Intuitively, we see that the two hold an inverse dependence (for a larger $\hat{v}$, fewer timed positions go unassigned and so on). Moreover, for every timed positions that goes unassigned, we require a new robot that can reach that position. Hence, in order to calculate the minimum number of robots needed, we restrict r to $\mathcal{K}$ and apply the *Hungarian Method* to the associated cost matrix $\hat{CV}$ given by (22). If we let z be the total number of unassigned timed positions in the *Score*, then,

$$z = |Sc| - (|range(\hat{H}_\mathcal{K})| - (|As| - |Sc|)) \qquad (23)$$

where $(|As| - |Sc|)$ denotes the number of *dummy* positions that are *always* assigned. Hence, using (23), the minimum number of robots, $r^\star = \mathcal{K} + z = \mathcal{K} + |As| - |range(\hat{H}_\mathcal{K})|$. ■

6. SIMULATIONS

To demonstrate the musically inspired problem central to this paper, we simulated an example of a wall in MATLAB, instrumented to sound like a piano (see Figure 4.1 (left)) Our goal was to make multiple robots (simulated as 2-d light (green) colored points in Figure 4.1 (center, right)) perform the popular composition "Für Elise" by Ludwig van Beethoven on this *Piano Wall*.

Note that firstly, all notes in "Für Elise" lie amongst the set of notes used to create the *Piano Wall*, and secondly, a pianist is required to hit a maximum of two keys simultaneously throughout its performance ($\mathcal{K} = 2$).

We created the *Score* associated with "Für Elise", containing timed positions on the wall corresponding to notes in "Für Elise", specified at a beat of one second. Additionally, we created 2 base stations at the bottom of the wall (simulated as 2-d dark (blue) colored points in Figure 4.1 (center, right)), from where robots could start in conjunction with Assumption 1. For different values of $r \geq \mathcal{K}$ and $\hat{v}$, we generated the *optimal assignment*, and the corresponding *optimal mapping* describing each robots path, for both unconstrained and constrained velocity cases. These paths were then executed by the robots with appropriate velocities that ensured their timely arrival at assigned positions. Moreover, in our program, the instant a robot reached an assigned timed position, it was encircled by a light circle (yellow), and the sound of the corresponding piano note was generated. Thus, our robots were able to effectively perform "Für Elise" on the *Piano Wall*. Instances of two such simulations are shown in Figure 4.1 (center, right).

REFERENCES

A. Arsie, K. Savla, and E. Frazzoli. Efficient routing algorithms for multiple vehicles with no explicit communications. *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 23022317, 2009.

T. Bektas. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega*, vol. 34, no. 3, pp. 209-219, 2006.

L. D. Bodin, B. L. Golden, A. A. Assad, and M. O. Ball. Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, vol. 10, no. 2, pp. 63–211, 1983.

F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, pp. 1–23, 2011.

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

U. Derigs. The shortest augmenting path method for solving assignment problems - Motivation and computational experience. *Annals of Operations Research* vol. 4, no. 1, pp. 57-102, 1985.

R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, eds. Plenum Press, New York, pp. 85–103, 1972.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* vol. 2, no. 1–2, pp. 83–97, 1955.

E. L. Lawler. Combinatorial optimization: networks and matroids. Holt, Rinehart and Winston, New York, 1976.

S. Martello and P. Toth. Linear assignment problems. *Annals of Discrete Mathematics*, vol. 31, pp. 259-282, 1987.

M. Pavone and E. Frazzoli. Dynamic vehicle routing with stochastic time constraints. *IEEE Int. Conf. on Robotics and Automation*, (Anchorage, AK), pp. 1460-1467, 2010.

T. K. Ralphs. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, vol. 29, no. 5, pp. 607–629, 2003.

S. L. Smith, M. Pavone, F. Bullo, and E. Frazzoli. Dynamic vehicle routing with priority classes of stochastic demands. *SIAM Journal on Control and Optimization*, vol. 48, no. 5, pp. 3224-3245, 2010.

M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, vol. 35, no. 2, pp. 254–265,1987.

Appendix A. PROOF OF THEOREM 1

Let $\hat{c}_{p,q}$ denote an element of the cost matrix $\hat{C}$, where $p, q \in \{1, ..., |As|\}$. Recall that previously, we denoted an element of $\hat{C}$ by $\hat{c}_{i_\alpha, j_\beta}$, where $i \in \mathcal{I}, \alpha \in \mathcal{A}_i, j \in \mathcal{J} \cup \{n+1\}$ and $\beta \in \mathcal{B}_j$ Hence, we can express p and q as follows,

$$p = \sum_{\theta=0, i>0}^{i-1} |P_\theta| + \alpha \tag{A.1}$$

$$q = \sum_{\theta=1, j>1}^{j-1} |P_\theta| + \beta \tag{A.2}$$

Now consider a bijective function $\omega_r : As \rightarrow Sc'$, such that $\omega_r((P_{i,\alpha}, t_i)) = (P_{j,\beta}, t_j) \Rightarrow \hat{c}_{i_\alpha, j_\beta} = \hat{c}_{p,p\,(p=q)}$, i.e. ω_r contains individual assignments between all timed positions in As and Sc' that have a corresponding diagonal cost entry in the cost matrix $\hat{C}$.

Let us assume there exists a forbidden individual assignment in ω_r, i.e. there exists a diagonal element in $\hat{C}$ that equals M. From (15), we note that $\hat{c}_{i_\alpha, j_\beta} = M \Rightarrow j \leq i$ (forbidden assignments violate the directionality constraint).

Now consider the diagonal element $\hat{c}_{p,p\,(p=q)} = \hat{c}_{i_\alpha, j_\beta}$, for some $i \in \mathcal{I}, \alpha \in \mathcal{A}_i, j \in \mathcal{J} \cup \{n+1\}$ and $\beta \in \mathcal{B}_j$. Clearly, to assume $\hat{c}_{p,p\,(p=q)} = \hat{c}_{i_\alpha, j_\beta} = M$ implies that $j \leq i$. Immediately, we notice that for $i = 0$, there exists no $j \in \mathcal{J} \cup \{n+1\}$ such that $j \leq i$. Similarly, for $j = n$ and $j = (n+1)$, there exists no $i \in \mathcal{I}$ such that $j \leq i$. As a result, we see that $\hat{c}_{p,p\,(p=q)} = \hat{c}_{i_\alpha, j_\beta} = M$ is a contradiction for the above values of i and j. Hence, we direct our focus on the remaining values of i and j, i.e. $i \in \{1, ..., n-1\}$ and $j \in \{1, ..., n-1\}$.

Notice that we can rewrite (A.1) for index p as follows,

$$p = |P_0| + \sum_{\theta=1, j>1}^{j-1} |P_\theta| + \sum_{\theta=j, i>j}^{i-1} |P_\theta| + \alpha \tag{A.3}$$

$$p = r + \sum_{\theta=1, j>1}^{j-1} |P_\theta| + \sum_{\theta=j, i>j}^{i-1} |P_\theta| + \alpha \tag{A.4}$$

Since $p = q$, we can equate (A.4) and (A.2) to get the following,

$$r + \sum_{\theta=j, i>j}^{i-1} |P_\theta| + \alpha = \beta \tag{A.5}$$

Recall that $r \geq \mathcal{K}$. Moreover, $r \geq K \Rightarrow r \geq \beta$, since $1 \leq \beta \leq |P_j| \leq \mathcal{K}, \quad \forall j \in \{1, ..., n-1\}$, thereby making (A.5) a contradiction. Hence, we have shown that no diagonal element in $\hat{C}$ equals M, or in other words, 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 $H_r : As \rightarrow Sc'$ which is, at the very least, equal to ω_r ($cost(H_r) \leq cost(\omega_r)$). Thus, we can construct an *optimal assignment* $H_r|_{As'} : As' \rightarrow Sc$, 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. ■