# On the Traveling Salesman Problem with Simple Temporal Constraints

**T. K. Satish Kumar***
Computer Science Department
University of Southern California
California, USA
tkskwork@gmail.com

**Marcello Cirillo**
AASS Research Centre
Örebro University
Sweden
marcello.cirillo@aass.oru.se

**Sven Koenig**
Computer Science Department
University of Southern California
California, USA
skoenig@usc.edu

## Abstract

Many real-world applications require the successful combination of spatial and temporal reasoning. In this paper, we study the general framework of the Traveling Salesman Problem with Simple Temporal Constraints. Representationally, this framework subsumes the Traveling Salesman Problem, Simple Temporal Problems, as well as many of the frameworks described in the literature. We analyze the theoretical properties of the combined problem providing strong inapproximability results for the general problem, and positive results for some special cases.

## Introduction

Tasks are usually situated in both time and space. While temporal and spatial reasoning are individually well studied, their combination is not straightforward. For example, Simple Temporal Networks (STNs) and Traveling Salesman Problems (TSPs) are two frameworks, for temporal and spatial reasoning respectively, which have been studied extensively over the years. However, little is known about the theoretical properties resulting from combining them.

A unified framework is important in many real-life domains. Imagine a surveillance vehicle which needs to autonomously decide in which order to perform the observation tasks it has been assigned. The tasks, however, are not completely independent of one another. For instance, the vehicle may be instructed to observe nearby areas allowing for a certain amount of time to elapse between observations. This dependence between tasks can be captured by temporal constraints. Under some assumptions on the nature of the temporal constraints, this problem is easily solved – or identified as unsolvable – even for a large number of tasks. Realistically, however, different tasks must be performed in different locations, and the vehicle must move from one place to the next. This adds a new dimension to the problem: a schedule which completely satisfies the temporal constraints between tasks could be infeasible because of the time the vehicle spends traveling from one location to the next. On the other hand, if we want to minimize the time (or distance) traveled by the vehicle, we still have to satisfy the temporal constraints among the tasks. Finding a schedule which satisfies the temporal constraints and minimizes the distance traveled, while taking into account the time necessary for the vehicle to move from location to location, is a problem for which there exists no general efficient solution technique.

In this paper, we first recount the two commonly used frameworks for solving temporal and spatial reasoning problems separately. We then introduce a unified framework, the *Traveling Salesman Problem with Simple Temporal Constraints* (TSP-STC). We analyze the combinatorial properties of the TSP-STC in light of recent results from the theoretical computer science community. This analysis yields results, both positive and negative, about the approximability of TSP-STCs. Such results in turn allow us to identify those aspects of the combined problem which require further research to obtain efficient solutions.

## Temporal Reasoning Problems

In this section, we recount well established formalisms for reasoning about temporal constraints. Many kinds of temporal relations, including the ones considered in this paper, can be represented on a directed graph $\mathcal{G} = \langle \mathcal{X}, \mathcal{E} \rangle$, where a vertex $X_i \in \mathcal{X}$ is an event and a directed edge $e = \langle X_i, X_j \rangle \in \mathcal{E}$ is a constraint on the relative execution times of $X_i$ and $X_j$. Conventionally, a special event $X_0$ is used to represent the "beginning of time" and its execution time is set to 0.

The simplest formalism for temporal reasoning is *precedence ordering*, which is commonly encoded by a directed edge $e = \langle X_i, X_j \rangle$, indicating that event $X_i$ should be executed before event $X_j$. Although their representational power is limited, precedence constraints are useful in practice since they are able to represent causal relationships. For instance, precedence constraints can model a plan whose actions are causally ordered. In our previous example of the surveillance vehicle, causal relationships would dictate that the vehicle should first reach the target area before observing it. Producing a total ordering for a set of precedence constraints, or alternatively identifying that no such ordering exists, can be done in polynomial time.

A more expressive but still tractable formalism for temporal reasoning is the framework of *Simple Temporal Problems* (STPs). Here, each directed edge $e = \langle X_i, X_j \rangle \in \mathcal{E}$,

*Alias: Satish Kumar Thittamaranahalli

annotated with the bounds $[LB(e), UB(e)]$, is a *simple temporal constraint* between $X_i$ and $X_j$, indicating that the relative execution times of events $X_i$ and $X_j$ are constrained by the pair of inequalities $LB(e) \leq X_j - X_i \leq UB(e)$.[1] A solution to an STP is an assignment of execution times to all events such that all simple temporal constraints are satisfied. STPs are one of the most widely used formalisms for reasoning about metric time. They are fairly rich in their expressiveness, although they cannot represent disjunctions. STPs can be solved in polynomial time using shortest path computations on their *distance graph* representations. In the distance graph representation, the constraint $X_j - X_i \leq w$ is represented as an edge from $X_i$ to $X_j$ annotated with a cost $w$. Each simple temporal constraint in the STP is therefore represented as a pair of edges in the distance graph. The absence of negative cost cycles in the distance graph characterizes the consistency of the temporal constraints (Dechter, Meiri, and Pearl 1991), that is, the existence of a solution. Shortest paths in the distance graph are commonly calculated using the Bellman-Ford algorithm. However, recent, more efficient algorithms can be employed for solving STP instances with additional structure (Planken, De Weerdt, and van der Krogt 2008).

There also exist more expressive formalisms for temporal reasoning, such as *Disjunctive Temporal Problems* (DTPs). However, their higher expressiveness comes at the cost of a higher complexity. In particular, DTPs are NP-hard problems, and all known procedures for solving them require exponential time. Here, we limit our analysis of combining temporal and spatial reasoning to cases with precedence and simple temporal constraints. The negative results proved here for the combined problem carry over to extensions in which DTPs are used instead of STPs.

## Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is an established formalism for reasoning about spatial problems and has been extensively studied by different communities. Many variants of the problem exist. In this section, we recount the results associated with those variants which are relevant to our formal definition of TSP-STCs.

The classical TSP is the problem of finding a *Hamiltonian cycle* of minimum cost on an edge-weighted complete undirected graph. A Hamiltonian cycle is a cycle in which each vertex of the graph is visited exactly once. The TSP is NP-hard and even hard to approximate within any polynomial factor. However, many of its variants can be approximated in polynomial time because they allow for tours instead of cycles. A tour visits all vertices of the graph, like a Hamiltonian cycle, but any vertex can be visited more than once. This relaxation is equivalent to the metric assumption, where the triangle inequality holds for the distances between vertices (Chekuri and Pál 2007).

The most common TSP variants consist of different combinations of assumptions and requirements, such as: a. Whether we assume a metric distance function; b. Whether distances between vertices are symmetric; c. Whether we are interested in calculating a path between given start and goal vertices or a cycle (in both cases all vertices should be visited); d. Whether a subset of the vertices should be visited in a given order; and e. Whether a subset of the vertices should be visited in an order consistent with specified precedence constraints. The difference between requirements (d) and (e) is that we have a total ordering over the vertices of the subset for (d), while we have a partial ordering for (e). Other variants based on different assumptions and requirements are well studied in different communities, such as the relaxation of the requirement that every vertex should be visited or the assumption that a price is assigned to each vertex. However, these variants are out of scope for our analysis.

Table 1 summarizes the known results for the TSP and some of its most common variants which are relevant to us (Charikar et al. 1997). There is no polynomial-time approximation algorithm for the classical TSP (unless $P = NP$). However, under the metric assumption, polynomial-time approximation algorithms can be designed. In particular, for the symmetric TSP, where the distance from one vertex to another is the same as that in the opposite direction, there exists a factor-$1.5$ polynomial-time approximation algorithm. In the case of the Asymmetric TSP (ATSP), the distances are not necessarily symmetric. The ATSP is amenable to an $O(\log n)$ polynomial-time approximation algorithm.[2] Next, the table lists both the symmetric and asymmetric *path* variants of the TSP, TSP-Path and ATSP-Path, respectively, where the start and end vertices are given. While the approximation factor of $O(\log n)$ carries over to the ATSP-Path, the best known polynomial-time algorithm for the TSP-Path has a slightly worse approximation factor of $5/3$.

Two other notable variants are the TSP and ATSP with precedence constraints. These variants allow the specification of precedence constraints between vertices, which can be encoded as a directed acyclic graph and interpreted as a partial order. A feasible solution is a total ordering on the vertices which is consistent with the partial order. The cost of a feasible solution is equal to the cost of the tour that it induces. An optimal solution is a feasible solution with minimum cost. There are fairly strong inapproximability results for the TSP and ATSP with precedence constraints, which hold even under the metric assumption. Finally, the TSP and ATSP with path constraints are special cases of the TSP and ATSP with precedence constraints, respectively, where the precedence constraints induce a total ordering on a subset of the vertices. These last two variants have factor-3 and $O(\log n)$ approximation algorithms, respectively.

## TSP-STC: A Formal Definition

Having reviewed STPs and TSPs, we now study a combination of the two, a spatial problem with temporal constraints where traveling from vertex to vertex takes time. In the example from the Introduction, the traversal times depend on both the terrain and the speed of the surveillance vehicle. Thus, the solution of the spatial part of the problem generates temporal constraints in addition to the ones already

[1] Here, for convenience, we use the same notation to indicate events and their execution times.

[2] Here, and in the rest of the paper, $n$ is the number of vertices.

| Variant | Assumptions | Tractable approximations |
|---|---|---|
| TSP | No assumptions | – |
| TSP | Symmetric distances, metric domains | 1.5 (Christofides 1976) |
| ATSP | Asymmetric distances, metric domains | $O(\log n)$ (Frieze, Galbiati, and Maffioli 1982) |
| TSP-Path (given start and goal vertices) | Symmetric distances, metric domains | $5/3$ (Hoogeveen 1991) |
| ATSP-Path (given start and goal vertices) | Asymmetric distances, metric domains | $O(\log n)$ (Chekuri and Pál 2007) |
| TSP with path constraints | Symmetric distances, metric domains | 3 (Bachrach et al. 2005) |
| ATSP with path constraints | Asymmetric distances, metric domains | $O(\log n)$ (Chekuri and Pál 2007) |
| TSP with precedence constraints | Symmetric distances, metric domains | Inapproximability results (Charikar et al. 1997) |
| ATSP with precedence constraints | Asymmetric distances, metric domains | Inapproximability results (Charikar et al. 1997) |

Table 1: A summary of the complexity results associated with different variants of the Traveling Salesman Problem.

specified by the STP. The complexity of the TSP-STC cannot be smaller than the one of TSPs or STPs individually.

Formally, a TSP-STC is a sextuplet $\langle \mathcal{V}, d, t, \mathcal{X}, c, \mathcal{E} \rangle$, where:

- $\mathcal{V}$ is the set of vertices, each of which represents a location;
- $d$ is the distance function, that maps an ordered pair of vertices to a non-negative real number ($d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geq 0}$), which represents the distance from one vertex to another;
- $t$ is the traversal function, that maps an ordered pair of vertices to a non-negative real number ($t : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geq 0}$), which represents the time required to move from one vertex to another;
- $\mathcal{X}$ is the set of events, as defined for a standard STP;
- $c$ is the function mapping events to vertices ($c : \mathcal{X} \to \mathcal{V}$);
- $\mathcal{E}$ is the set of directed edges of the form $e = \langle X_i, X_j \rangle$, annotated with the bounds $[LB(e), UB(e)]$. Each $e$ is a simple temporal constraint between two events $X_i$ and $X_j$.

A feasible solution of a TSP-STC is a total ordering on events in $\mathcal{X}$ and an assignment of execution times to all of them such that: a. The execution times are consistent with the total ordering; b. The execution times are consistent with the constraints in $\mathcal{E}$; and c. The execution times of two consecutive events $X_i$ and $X_{i+1}$ in the total ordering satisfy the induced constraint $X_{i+1} - X_i \geq t(c(X_i), c(X_{i+1}))$. The last condition accounts for the traversal time between vertices (locations) of consecutive events while not penalizing waiting time at any location.[3]

Since every event in $\mathcal{X}$ is mapped to a unique location in $\mathcal{V}$, a feasible solution defines a visit sequence on the locations. The cost of a feasible solution of a TSP-STC is equal to the cost of the induced tour as derived from the distance function $d$. An optimal solution of the TSP-STC is a feasible solution of minimum cost.

Just like for TSPs, we assume metric distances between vertices for TSP-STCs as well. Unless otherwise specified, we also assume that the distance function of TSP-STCs is symmetric. If this is not the case, we refer to these problems as Asymmetric Traveling Salesman Problems with Simple Temporal Constraints (ATSP-STCs).

The above definition is only one possible way to combine TSPs and STPs. However, it is general enough to encode many real-world problems. In fact, both TSPs and STPs are special cases of TSP-STCs. Given a TSP with a set of vertices $\mathcal{V}$ and a distance function $d$, it can be represented in this framework by imposing the following conditions: $t(V_i, V_j) = 0$ for all $1 \leq i, j \leq |\mathcal{V}|$; $|\mathcal{X}| = |\mathcal{V}|$, where each $X_i \in \mathcal{X}$ is a fictitious event; $c$ is a bijective function mapping each event to a unique location and vice versa; and $\mathcal{E} = \emptyset$. These conditions entail that every vertex must be visited in the tour, but no temporal constraints need to be considered. Conversely, a standard STP with a set of events $\mathcal{X}$ and a set of constraints $\mathcal{E}$ can be represented in this framework by imposing the following conditions: $\mathcal{V} = \{V_0\}$; $d(V_0, V_0) = 0$; $t(V_0, V_0) = 0$; and each event $X_i \in \mathcal{X}$ is associated with the same location, meaning that $c(X_i) = V_0$ for all $1 \leq i \leq |\mathcal{X}|$.

## Computational Analysis of TSP-STCs

In this section, we present a complexity analysis of TSP-STCs. We show that this class of problems is subject to strong inapproximability results, even under simplifying assumptions commonly made for TSPs. We first address the complexity of TSP-STCs when temporal constraints are limited to precedence constraints (Theorems 1 and 2). Next, we prove that TSP-STCs are NP-hard to approximate within any polynomial factor, even under the assumption that both the distance and traversal functions are metric and symmetric.

[3] In the rest of the paper, we use "vertex" and "location" synonymously, as one corresponds to the other in our definition of the TSP-STC.

As TSP-STCs are a special case of ATSP-STCs, the inapproximability results for the former carry over to the latter.

**Theorem 1.** *For the TSP-STC, there is no polynomial-time $|\mathcal{V}|^\alpha$-approximation algorithm, for some $\alpha > 0$, unless $P = NP$.*

**Theorem 2.** *For the TSP-STC, there is no polynomial-time $(\log |\mathcal{V}|)^\delta$-approximation algorithm, for any $\delta > 0$, unless $NP \subseteq DTIME(|\mathcal{V}|^{\log \log |\mathcal{V}|})$.*

*Proof.* Consider TSPs with precedence constraints. Since precedence constraints are a special case of simple temporal constraints, TSPs with precedence constraints are a subclass of TSP-STCs and therefore inapproximability results for the former carry over to the latter. Theorems 1 and 2 correspond to Theorems 5 and 9 in (Charikar et al. 1997)[4] after setting $k = n = |\mathcal{V}|$. □

## A Closer Look at TSP-STCs with Precedence Constraints

Theorems 1 and 2 dictate that we cannot design polynomial-time approximation algorithms for TSP-STCs or ATSP-STCs with precedence constraints. However, we can precisely characterize the instance complexity of solving TSP-STCs and ATSP-STCs with only precedence constraints.

The first, trivial case encompasses instances without any temporal constraints, resulting in classical TSPs and ATSPs, for which there exist applicable polynomial-time approximation algorithms (see Table 1).

The second case encompasses instances where the subset of events for which precedence constraints are specified is totally ordered, resulting in TSPs and ATSPs with path constraints, for which there exist factor-3 and $O(\log n)$ approximation algorithms, respectively.

Finally, the third case encompasses those instances where precedence constraints are specified, but whose events cannot be uniquely ordered. For such cases, we can still reduce TSP-STCs and ATSP-STCs with only precedence constraints to multiple instances of TSPs and ATSPs with path constraints, where each instance corresponds to a total ordering over a subset of the vertices consistent with the original precedence constraints. The best guaranteed result obtained by evaluating all possible total orderings is a factor-3 approximation for the given original instance of the TSP-STC with precedence constraints or an $O(\log n)$ approximation for the ATSP-STC instance with precedence constraints. Obviously, the overall running time to find an approximate solution depends on the number of total orderings for that specific subset which are consistent with the specified precedence constraints.

The tractability of approximating TSP-STCs and ATSP-STCs with only precedence constraints depends on the number of total orderings over all events allowed by the precedence constraints. As a rule of thumb, a large space of possible total orderings (as in the first case) and a small space of total orderings (as in the second case) are both amenable to efficient approximations.

## Strong Inapproximability Results

We now prove a strong inapproximability result for TSP-STC. Theorem 3 is stronger than Theorems 1 and 2 as it proves the inapproximability of the problem within any polynomial factor. Moreover, it proves that the negative results hold even under the assumption that both the distance and traversal functions are metric and symmetric. It is worth comparing this with the original TSP, which, although inapproximable within any polynomial factor, becomes amenable to tractable approximations under the metric assumption.

**Theorem 3.** *The TSP-STC is NP-hard and also NP-hard to approximate within any polynomial factor, even under the assumptions that both the distance and traversal functions are metric and symmetric.*

*Proof.* We reduce the Hamiltonian path problem to the TSP-STC. The Hamiltonian path problem is the problem of finding a path in a given undirected graph in which each vertex of the graph is visited exactly once. Consider a Hamiltonian path problem over an undirected graph $G = \langle \mathcal{N}, \mathcal{A} \rangle$, where $\mathcal{N}$ is a set of vertices ($|\mathcal{N}| = n$) and $\mathcal{A}$ is a set of undirected edges ($|\mathcal{A}| = k$). We associate each $N_i \in \mathcal{N}$ with a vertex $V_i \in \mathcal{V}$. For each $1 \le i \le n$, we define a unique event $X_i \in \mathcal{X}$ associated with $V_i$. We define $t$ and $d$ as follows: $t(V_i, V_j) = d(V_i, V_j) = t(V_j, V_i) = d(V_j, V_i)$; $t(V_i, V_j) = 1$ if there is an edge between $N_i$ and $N_j$ in $G$; $t(V_i, V_j) = 1.5$ otherwise. Note that both $t$ and $d$ are symmetric and metric, as the triangle inequality holds by construction. We complete the construction of the TSP-STC instance by defining $2\binom{n}{2}$ temporal constraints of the form $e_{ij} = \langle X_i, X_j \rangle$ for all $1 \le i, j \le n$ and $i \ne j$, with $LB(e_{ij}) = -\infty$ and $UB(e_{ij}) = n - 1$. A Hamiltonian path exists in $G$ *iff* we can visit all vertices in the TSP-STC and satisfy the temporal constraints. This is so because the temporal constraints dictate that, regardless of the starting point of the path, every vertex should be visited within $n - 1$ time units. The traversal function indicates that it takes exactly one time unit to go from one vertex to another if there is an edge between the two in $G$. It is easy to see that a solution of the Hamiltonian path problem maps to a solution of the TSP-STC instance constructed above. In addition, a solution of the TSP-STC instance maps to a solution of the Hamiltonian path problem: no vertex can be visited twice and no edge can be traversed in the TSP-STC which was not present in $G$ because the time constraints would be violated. Finally, we can view the TSP-STC as a constraint optimization problem. The optimization component of the problem is the minimization of the sum of the distances in the tour induced by the visitation order. The satisfaction component is to respect the simple temporal constraints and the constraints induced by the traversal function. If the satisfaction component is itself NP-hard, it follows that the TSP-STC is NP-hard to approximate within any polynomial factor. As we have demonstrated, the reduction from the Hamiltonian

[4]The two Theorems were in turn adapted from (Bhatia, Khuller, and Naor 1995). Note that the two Theorems were originally formulated for *walks* instead of tours. However, when considering metric and symmetric distances, the cost of a tour corresponding to a given walk is at most twice the cost of the walk.

path problem is only to the satisfaction component of the TSP-STC, hence proving the Theorem. □

## A Notable Special Case

We now analyze an important special case of the TSP-STC which is reducible to TSPs with precedence constraints and to which the analysis presented above applies. This notable case occurs when the traversal function is degenerate, i.e., maps all combinations of its arguments to zero, and all simple temporal constraints can be expressed in the form of *time windows*. A time window represents an interval during which an event must be executed. This is more restrictive than simple temporal constraints because, in the latter case, we can constrain the relative execution times of two different events, while, with time windows, we only constrain the execution times of individual events. A time window is defined by the interval $[a_i, b_i]$ for event $X_i$, where $a_i$ is the start time and $b_i$ the end time, with $a_i \leq b_i$. A degenerate traversal function captures the case in which the agent which has to travel between locations can move fast enough so that the traversal times can be disregarded as a factor in the problem domain. The case we analyze can be viewed as a special case of the well studied TSP with Time Windows (TSPTW) which also considers non-degenerate traversal functions (Melvin et al. 2007).

A time window for event $X_i$ is modeled in the STP component of the TSP-STC as a simple temporal constraint $\langle X_0, X_i \rangle$ annotated with the bounds $[a_i, b_i]$, where $X_0$ is a special event which is used to represent the "beginning of time" and is conventionally set to 0. A TSP-STC instance with only time window constraints and a degenerate traversal function can be reduced to an instance of the TSP with precedence constraints as follows. The set of locations $\mathcal{V}$ and the distance function $d$ remain unchanged, but locations are duplicated, if needed, so that at most one event is assigned to each location. For all events $X_i$ and $X_j$ with $b_i < a_j$, a precedence constraint $c(X_i) \prec c(X_j)$ is added.

A solution of the original TSP-STC instance is retrieved from a solution of the corresponding instance of the TSP with precedence constraints as follows. We assign execution times to the events in the TSP-STC using the total ordering obtained as a solution of the TSP with precedence constraints. This assignment of execution times is constructed to satisfy the invariant that the execution time of any event $X_i$ is always equal to the starting point of the time window of an already scheduled event $X_j$. Assume that the event that corresponds to the initial location of the TSP with precedence constraints is $X_i$. We set the execution time of $X_i$ to $a_i$. Thus, it holds that the execution time of $X_i$ corresponds to the starting point of the time window of an already scheduled event (namely, $X_i$). Now, assume that the solution of the TSP with precedence constraints moves from the location that corresponds to event $X_i$ to the location of the next event $X_j$ in the total ordering. It cannot be the case that $b_j < X_i$ for the following reason: $X_i$ corresponds to the starting point of the time window of an already scheduled event $X_k$, which the solution of the TSP with precedence constraints visits no later than event $X_i$. But this is impossible since $X_i = a_k$ and $b_j < X_i$ mean that there is a precedence constraint $c(X_j) \prec c(X_k)$. Thus, a solution of the TSP with precedence constraints cannot visit $X_k$ and then later move to $X_j$. Thus, $X_i \leq b_j$. We now have two distinct cases:

- $a_j \leq X_i$, that is, $X_i \in [a_j, b_j]$. In this case, we set $X_j = X_i$, which satisfies the time window constraint of $X_j$. It holds that $X_j$ corresponds to the starting point of the time window of an already scheduled event (namely, $X_k$).
- $a_j > X_i$. In this case, we set $X_j = a_j$, which satisfies the time window constraint of $X_j$. Here, too, it holds that $X_j$ corresponds to the starting point of an already scheduled event (namely, $X_j$).

We claim that any solution of the TSP-STC instance is also a solution of the corresponding instance of the TSP with precedence constraints and vice versa. It would therefore follow that the optimal solution of the corresponding instance of the TSP with precedence constraints is also an optimal solution of the original instance of the TSP-STC. Consider any solution of the TSP-STC. Assume, for proof by contradiction, that it does not satisfy some precedence constraint $c(X_i) \prec c(X_j)$ of the TSP with precedence constraints because $X_i \geq X_j$. This is a contradiction since $X_i \leq b_i < a_j \leq X_j$ due to the semantics of the precedence constraint $c(X_i) \prec c(X_j)$. Conversely, consider any solution of the TSP with precedence constraints. The procedure described above already provides an algorithmic construction of a corresponding solution for the TSP-STC.

## Related Work

The combination of spatial and temporal aspects is important in different domains, among which are robotics and vehicle routing. Thus, problems similar or identical to TSP-STCs have been studied in operations research, theoretical computer science, artificial intelligence and robotics. Different combinations of spatial and temporal aspects are possible. The spatial aspects can be expressed by minimizing distances or maximizing the (uniform or non-uniform) total reward of the visited vertices. The temporal aspects can be expressed via precedence constraints, absolute time windows or more general temporal constraints. The resulting combined problems are often solved with constraint programming, branch-and-bound search and dynamic programming as exact algorithms; or genetic algorithms, tabu search, cooperative auctions and insertion or interchange heuristics as heuristic algorithms. In this section, we present a general overview of the solution techniques adopted in different research areas.

Robotics researchers have studied multi-robot routing problems with rewards and disjoint time windows (that do not overlap), where robots have to visit targets during given time windows. The objective is to maximize the sum of the rewards of the visited targets minus the sum of the costs incurred for moving from target to target (Melvin et al. 2007). The problem is solvable in pseudo polynomial time for a single robot but is NP-hard for multiple robots, although special cases can be solved in polynomial time (including the case where the robots are identical and the targets have singleton time windows). Robotics researchers have also studied multi-robot routing problems where robots have to visit targets in the presence of precedence constraints between targets. The objective is to maximize the sum of time-decreasing rewards of the targets (Jones, Dias, and Stentz 2011). Cooperative auctions and genetic algorithms have been proposed as heuristic algorithms to solve this problem. An alternative approach for solving multi-robot routing problems is to use multiple constraint solvers which progressively refine trajectory envelopes for each vehicle according to mission requirements, by leveraging the notion of least commitment to obtain easily revisable trajectories for execution (Pecora, Dimitrov, and Cirillo 2012).

Theoretical computer science researchers have studied prize-collecting traveling salesman (or vehicle routing) problems with time windows, where a salesperson (or vehicle) has to visit customers during given time windows. The objective is to maximize the sum of the rewards of the visited customers. Several scheduling problems with sequence-dependent setup times can be reduced to this problem, which can be solved with $O(\log n)$ approximation algorithms (Bansal et al. 2004).

Operations researchers have studied time-constrained TSPs, where a salesperson has to visit customers during given time windows. The objective is to minimize the travel distance or time (Baker 1983). It is already NP-hard to decide whether a feasible solution exists (Savelsberg 1985). Constraint programming, branch and bound search and dynamic programming have been proposed as exact algorithms to solve this problem, and greedy heuristics or interchange heuristics as heuristic algorithms (Pesant et al. 1998).

## Conclusions

In this paper, we defined the framework of the TSP-STC, which combines the temporal aspects of STPs and the spatial aspect of TSPs. We recounted known computational results from the theoretical computer science community for the spatial and temporal aspects of the problem separately. We then analyzed the problem in its entirety, proving strong inapproximability results. These results hold even under common assumptions (such as the metric assumption) which allow for tractable approximations for TSPs.

Despite these negative results, we were able to present special cases which are amenable to tractable approximations with low-order instance complexities. Given the complexity of the TSP-STC and the strong inapproximability results, the most promising avenues for future investigations are heuristic and knowledge engineering approaches. From the heuristic perspective, we can potentially generalize well established heuristics for TSPs (e.g., the 2-opt heuristic). From the knowledge engineering perspective, we can potentially employ mixed-initiative approaches.

Although the TSP-STC is only a particular way of combining spatial and temporal reasoning, it is general enough to capture the requirements of many real-world applications. Therefore, our results bear strong implications on a variety of problems, namely the combination of temporal and spatial planning in agent-based systems and robotics. As we have seen in the previous sections, there are many existing methods for addressing more restrictive cases, whose applicability is, however, very limited. Our analysis covers the cases in which a single agent or robot is present, and therefore the inapproximability results carry over to multi-agent systems as well, which are more complex.

Our future work will focus on the adaptation of well established heuristic techniques developed in the TSP framework to TSP-STCs and on the generalization of our framework to multi-agent systems.

**Acknowledgments.** This paper is based upon research supported by a MURI under contract/grant number N00014-09-1-1031 and by the Swedish Knowledge Foundation (KKS) under project Safe Autonomous Navigation (SAUNA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies or the U.S. government.

## References

Bachrach, A.; Chen, K.; Harrelson, C.; Mihaescu, R.; Rao, S.; and Shah, A. 2005. Lower bounds for maximum parsimony with gene order data. In *Proceedings of the International Conference on Comparative Genomics*.

Baker, E. 1983. An exact algorithm for the time-constrained traveling salesman problem. *Operations Research* 31(5):938–945.

Bansal, N.; Blum, A.; Chawla, S.; and Meyerson, A. 2004. Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *Proceedings of the Annual ACM Symposium on Theory of Computing*.

Bhatia, R.; Khuller, S.; and Naor, J. 1995. The loading time scheduling problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*.

Charikar, M.; Motwani, R.; Raghavan, P.; and Silverstein, C. 1997. Constrained TSP and low-power computing. In Dehne, F.; Rau-Chaplin, A.; Sack, J.-R.; and Tamassia, R., eds., *Algorithms and Data Structures*, volume 1272 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 104–115.

Chekuri, C., and Pál, M. 2007. An $O(\log n)$ approximation ratio for the asymmetric traveling salesman path problem. *Theory of Computing* 3(1):197–209.

Christofides, N. 1976. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1):61–95.

Frieze, A. M.; Galbiati, G.; and Maffioli, F. 1982. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks* 12(1):23–39.

Hoogeveen, J. 1991. Analysis of Christofides' heuristic:

Some paths are more difficult than cycles. *Operations Research Letters* 10(5):291–295.

Jones, E.; Dias, M.; and Stentz, A. 2011. Time-extended multi-robot coordination for domains with intra-path constraints. *Autonomous Robots* 30(1):41–56.

Melvin, J.; Keskinocak, P.; Koenig, S.; Tovey, C.; and Ozkaya, B. Y. 2007. Multi-robot routing with rewards and disjoint time windows. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Pecora, F.; Dimitrov, D.; and Cirillo, M. 2012. On mission-dependent coordination of multiple vehicles under spatial and temporal constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Pesant, G.; Gendreau, M.; Potvin, J.-Y.; and Rousseau, J.-M. 1998. An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science* 32(1):12–29.

Planken, L.; De Weerdt, M.; and van der Krogt, R. 2008. $P^3C$: A new algorithm for the simple temporal problem. In *Proceedings of the International Conference on Planning and Scheduling (ICAPS)*.

Savelsberg, M. 1985. Local search in routing problems with time windows. *Annals of Operations Research* 4:285–305.