

# Cooperative TSP

Amitai Armon, Adi Avidor, and Oded Schwartz

School of Computer Science,  
Tel-Aviv University, Tel-Aviv 69978, Israel  
{armon,adi,odedsc}@tau.ac.il

**Abstract.** In this paper we introduce and study cooperative variants of the Traveling Salesperson Problem. In these problems a salesperson has to make deliveries to customers who are willing to help in the process. Customer cooperativeness may be manifested in several modes: they may assist by approaching the salesperson, by reselling the goods they purchased to other customers, or by doing both.

Several objectives are of interest: minimizing the total distance traveled by all the participants, minimizing the maximal distance traveled by a participant and minimizing the total time until all the deliveries are made.

All the combinations of cooperation-modes and objective functions are considered, both in weighted undirected graphs and in Euclidean space. We show that most of the problems have a constant approximation algorithm, many of the others admit a PTAS, and a few are solvable in polynomial time. On the intractability side we provide NP-hardness proofs and inapproximability factors, some of which are tight.

## 1 Introduction

The Traveling Salesperson Problem (TSP) is a classical problem in combinatorial optimization, which has been studied extensively in many forms. Cooperative TSP is a set of variants of TSP in which the customers are allowed to move in order to assist the selling process. They may move in order to expedite the deliveries, and may also move after meeting the salesperson in order to help the distribution of the goods. For example, consider a secret message that has to be distributed to several people, but is only allowed to be passed in person. Every person who receives the message may then assist by passing it forward. We may want to devise a scheme for delivering the secret to all the recipients as fast as possible. A further illustration is the problem of an ice cream van vendor. The vendor wishes to sell ice cream to all children in town. The children are eager to cooperate, by approaching the van in order to buy ice cream. However, in contrast to the previous example, they are not interested in selling ice cream to others.

Formally, an instance of Cooperative TSP (cTSP) is a set of *agents* and a *salesperson*, located in a finite metric space or a Euclidean space. A solution is a synchronized series of move instructions to all *participants* (i.e., the salesperson

and the agents), such that all the agents eventually receive the delivery. We next elaborate on the various cooperation modes and the cost of solutions.

**Cooperation Modes.** We consider three modes of cooperation. In the PURCHASE-COOPERATION mode the salesperson has to meet all agents, and the agents are allowed to move towards the salesperson. In the SALES-COOPERATION mode, each agent receiving a delivery becomes capable of making deliveries similarly to the salesperson. However, an agent is not allowed to move before receiving a delivery. In the FULL-COOPERATION mode, an agent may cooperate in both the purchase and sales phases. That is, an agent may move before receiving the delivery and may make deliveries after receiving it.

**Goal Functions.** Three objectives are considered for Cooperative TSP: minimizing the total length traversed by all participants (MIN-SUM), minimizing the maximal length traversed by a participant (MIN-MAX), and minimizing the total time until the sales process ends (MIN-MAKESPAN).

We consider Cooperative TSP in a fixed-dimension Euclidean space and in weighted undirected graphs (note that w.l.o.g, we may assume that the graph is complete and weights satisfy the triangle inequality). We consider the ROUNDRIP versions, in which all participants are required to return to their initial location, and the PATH versions in which there is no such requirement.

**Related Studies.** TSP remains NP-hard even in the special planar variant. However, the latter variant has a PTAS [Aro98,Mit99]. When metric space is assumed, the Christofides [Chr76] approximation algorithm yields a  $\frac{3}{2}$ -approximation ratio and an inapproximability factor of  $\frac{203}{202}$  was shown [EK01].

*The Freeze-Tag Problem* was first suggested and studied by Arkin *et al.* in [ABF<sup>+</sup>02]. This problem arises in the context of swarm robotics: how to wake a set of slumbering robots, by having an already awake robot move to their locations. Once a robot is awake it can assist in waking up other slumbering robots. The objective is to have all robots awake as early as possible. In our terminology this is the PATH version of MIN-MAKESPAN SALES cTSP. Arkin *et al.* [ABF<sup>+</sup>02] provided an NP-hardness proof, a PTAS for the Euclidean variant, and a constant approximation for some graph families. A series of studies followed (e.g., [SABM02,ABG<sup>+</sup>03,KLS04]) culminating with an  $O(\sqrt{\log n})$ -approximation for the general weighted graph case [KLS04].

*TSP with Neighborhoods* is a proximity-related variant of TSP. In this problem each customer is willing to meet the salesperson anywhere within some neighborhood. The problem was first studied by Arkin *et al.* [AH94], followed by quite a few papers (e.g., [MM95,DM01,dBGK<sup>+</sup>05,SS05,Mit06]). This problem seems quite related to PURCHASE cTSP, as in both customers are willing to approach the salesperson. However, in TSP with Neighborhoods the customers' travel is not counted in the goal function, while in Cooperative TSP their moves do cost, and are part of the optimization task.

*Other Cooperative Multi-Agents Routing Problems.* As noted in [ABF<sup>+</sup>02], the Freeze-Tag Problem (and thus the Cooperative TSP problems) bears features of broadcasting, routing, scheduling and network design. The *minimum broadcast time*, the *multicast problem* and the *minimum gossip time problem* are all closely

	Goal	PURCHASE COOPERATION		SALES COOPERATION		FULL COOPERATION	
		Approx.	Inapprox.	Approx.	Inapprox.	Approx.	Inapprox.
PATH	MIN-SUM	$2 + \ln 3$	NP-hard	2	NP-hard	$2 + \ln 3$	APX-hard
	MIN-MAX	PTAS	no FPTAS	3	$2 - \varepsilon$	4	$2 - \varepsilon$
	MIN-MAKESPAN	Polynomial		$O(\sqrt{\log n})^1$	$\frac{5}{3} - \varepsilon^2$	2	$2 - \varepsilon$
ROUND	MIN-SUM	$\frac{3}{2}$	$\frac{203}{202} - \varepsilon$	$\frac{3}{2}$	$\frac{203}{202} - \varepsilon$	$\frac{3}{2}$	$\frac{203}{202} - \varepsilon$
	MIN-MAX	PTAS	no FPTAS	3	$\frac{3}{2} - \varepsilon$	2	$2 - \varepsilon$
TRIP	MIN-MAKESPAN	Polynomial		$O(\sqrt{\log n})$	$\frac{5}{4} - \varepsilon$	2	$2 - \varepsilon$

**Table 1.** Summary of approximation factors vs. inapproximability ratios in graphs. The parameter  $\varepsilon$  stands for an arbitrarily small positive constant, or for a positive function that tends to zero as the input size increases. (1) is by [KLS04], (2) is by [ABF<sup>+</sup>02].

related to Cooperative TSP (see [HHL88] for a survey and [Rav94,BNGNS98] for approximation results). Controlling swarms of robots in order to perform a certain task, has also been studied in various algorithmic aspects, including environment exploration, robot formation, searching and recruitment (see [ABF<sup>+</sup>02] for a list of relevant papers). Other researches are trying to confront similar scenarios, but with no central control, where each agent has to make decisions with limited knowledge regarding the environment and the other agents (for example, the problem of routing autonomous agents in wireless sensor network; and ants behavior inspired algorithms; see [ABF<sup>+</sup>02] for a list of relevant papers).

As cTSP is a generalization of both the Freeze-Tag and the TSP with Neighborhoods problems, the algorithms (and intractability results) achieved for cTSP apply to similar scenarios, e.g. cooperative robots tasks (see for example [AH94,ABF<sup>+</sup>02] for other relevant scenarios).

The MIN-MAX cost function is suitable, for example, when there is a bound on the energy that each robot is allowed to spend. The MIN-SUM cost function, on the other hand, is relevant when travel costs of all robots are covered by a single entity, who is therefore interested in minimizing the sum of these costs.

**Our Contribution:** We consider all combinations of cooperation modes, goal functions, path / roundtrip and graph / Euclidean. See Table 1 for the results on graphs and Table 2 for the results in Euclidean space. We obtain constant approximations and hardness results for most of the problems, PTAS for many of the others and polynomial-time exact solutions for a few. On the intractability side we give NP-hardness and inapproximability factors for all the graph problems and for some of the Euclidean problems.

**Paper Organization.** From here on, by cTSP we mean the PATH (rather than the ROUNDTrip) version of the corresponding Cooperative TSP problem, unless otherwise stated.

In Section 2 we present some of the results for EUCLIDEAN cTSP. Section 3 contains some of the results for cTSP problems on graphs. The proofs for the other results are either achieved using similar methods to those presented here, or are straightforward, and are omitted from this version due to space limitation.

	Goal	PURCHASE COOPERATION	SALES COOPERATION	FULL COOPERATION
PATH	MIN-SUM	PTAS	$\frac{5}{3} + \varepsilon$	$2 + \varepsilon$
	MIN-MAX	PTAS	3	4
	MIN-MAKESPAN	Polynomial	PTAS <sup>(1)</sup>	PTAS
ROUND TRIP	MIN-SUM	PTAS	PTAS	PTAS
	MIN-MAX	PTAS	3	2
	MIN-MAKESPAN	Polynomial	PTAS	PTAS

**Table 2.** Summary of approximation factors in Euclidean space, for any fixed dimension. The parameter  $\varepsilon$  stands for an arbitrarily small positive constant, or for a positive function that tends to zero as the input size increases. (1) is by [ABF<sup>+</sup>02].

## 2 EUCLIDEAN CTSP

This section presents some of the results obtained for EUCLIDEAN CTSP.

### 2.1 A PTAS for MIN-SUM PURCHASE EUCLIDEAN-CTSP

We next provide a PTAS for MIN-SUM PURCHASE EUCLIDEAN-CTSP. The algorithm and analysis below use Arora’s technique for the PTAS of Euclidean TSP [Aro98]. Our algorithm differs from Arora’s algorithm in that it has to consider *all* the agents’ paths and not just the salesperson’s path. We show how this can be done while keeping the dynamic programming polynomial. We show:

**Theorem 1.** MIN-SUM PURCHASE EUCLIDEAN-CTSP *admits a PTAS.*

Note that the problem is NP-hard even for the planar case. This follows, since an instance of the planar TSP can be reduced to an instance of MIN-SUM PURCHASE EUCLIDEAN-CTSP by simply replacing each costumer with three agents. This makes an instance where the salesperson is the only participant who moves.

We next describe the PTAS for the planar case. The extension to any fixed dimension is straightforward. Our terminology resembles the one of Arora [Aro98] and is given here for completeness.

Let  $\varepsilon > 0$  be an arbitrary small constant. Denote by  $n$  the number of participants and by  $OPT$  the cost of the optimal solution. Let  $L = 2^{3+\lceil 2 \log n \rceil}$ . Without loss of generality, that all the participants are located inside the *bounding box*  $[0, L/2]^2$  and that  $OPT > L/4$ .

**Super-pixels.** We call each square  $[j, j + 2] \times [j', j' + 2]$ , where  $j, j' \in \{0, 2, 4, \dots, L - 2\}$ , a *pixel*. We name the point  $(j + 1, j' + 1)$  the *center of the pixel*  $[j, j + 2] \times [j', j' + 2]$ . For every  $i = 0, \dots, \log L - 1$ , we call each square  $[j, j + L/2^i] \times [j', j' + L/2^i]$ , where  $j, j' \in \{0, L/2^i, 2 \cdot L/2^i, \dots, L - L/2^i\}$ , a *super-pixel* of level  $i$ . It is not hard to see that, without loss of generality, we may consider only instances for which all the participants are located at pixel centers.

An  $(a, b)$ -**shifting**. Let  $0 \leq a, b < L/2$  be two **even** integers. For a set  $A \subseteq [0, L/2]^2$  we define the  $(a, b)$ -*shift* of  $A$  to be the set  $\{(x + a, y + b) \mid (x, y) \in A\}$ .

**Portals.** Let  $m \in [\frac{8\sqrt{2}\log L}{\varepsilon}, \frac{16\sqrt{2}\log L}{\varepsilon})$  be a power of 2. For each super-pixel we mark each one of its four boundaries with  $m$  equidistant points that we refer to as *portals*. In particular, the portals include the four corners of the super-pixel. Note that, as  $m$  is a power of 2, each portal of a super-pixel of level  $i$  is also a portal of a smaller super-pixel of level  $i + 1$ , for  $i = 0, \dots, \log L - 2$ .

**Portals-limited-solutions.** We define a *portals-limited-solution* as a solution that satisfies the following four conditions:

1. Each participant may cross the boundary of a super-pixel only at its portals.
2. The salesperson does not cross her own route except on portals, where she may visit at most twice.
3. A meeting between an agent and the salesperson occurs only at a pixel center.
4. If two (or more) agents happen to reside at a pixel, then they all travel to (or stay at) the pixel's center and cease to move.

Therefore, in a portals-limited-solution, the tour of each participant is a collection of segments which connect portals to portals, and centers of pixels to portals. Additionally, a meeting between an agent and the salesperson occurs only at a pixel center, and tours of two agents do not cross.

**Lemma 1.** *Let  $a, b$  be two even integers chosen uniformly at random from the set  $\{0, 2, \dots, L/2 - 2\}$ . Then, the expected cost of a minimal cost portals-limited-solution of the  $(a, b)$ -shifted instance, is at most  $(1 + \varepsilon) \cdot OPT$ .*

**Lemma 2.** *A minimal cost portals-limited-solution can be found in time polynomial in  $n$ .*

The proof of Lemma 1 mainly follows arguments from the PTAS of Euclidean TSP and will appear in the full version of the paper. The PTAS enumerates over all  $O(L^2)$  values of  $(a, b)$  pairs. For each pair it applies Lemma 2 to find a minimal cost portals-limited-solution. Finally, it outputs the cheapest solution found, which according to Lemma 1, must have a cost of at most  $(1 + \varepsilon) \cdot OPT$ . Clearly, the  $O(n^4)$  factor in running time, caused by the enumeration over all  $(a, b)$  pairs, can be avoided if only an *expected*  $(1 + \varepsilon) \cdot OPT$  cost is desired.

*Proof. (of Lemma 2)* We use dynamic programming to build a polynomial-size table. For each super-pixel, the table contains  $6^{4m} = n^{O(1/\varepsilon)}$  entries. For each entry we store *portions* of some portals-limited-solutions (the portions of solutions limited to that super-pixel) together with their contribution to the overall cost.

The construction of the table is conducted in a bottom-up manner, starting from the pixels. A minimal value portals-limited-solution for the whole instance is obtained at the bounding box super-pixel.

The entries of the table for each super-pixel are represented by a list of  $4m$  elements, one element for each portal of the super-pixel. Each element takes one of the following six values:

1. The salesperson enters the super-pixel at this portal
2. The salesperson leaves the super-pixel at this portal
3. The salesperson enters and leaves the super-pixel at this portal
4. One agent enters the super-pixel at this portal
5. One agent leaves the super-pixel at this portal
6. None of the participants uses this portal

Note that the conditions defining a portals-limited-solution guarantee that these six cases cover all possible tour portions induced by all portals-limited-solutions (here we use the fact that two agents do not happen to reach the same portal, as they start at pixel centers, their tours do not cross and they end up at pixel centers). Also note that not all the  $4m$ -size lists represent a valid portion of some portals-limited-solution. We use the term *valid-list* for a list that represents a collection of tours that can be extended to some portals-limited-solution. Clearly, there are at most  $6^{4m} = n^{O(1/\varepsilon)}$  (valid-)lists. Finally, note that the salesperson's paths can intersect only at his entrance or exit points. Hence, given a valid-list, *pairings* of the participants' entrance and exit points can be found as in the algorithm of Arora [Aro98].

We now describe the construction in a bottom-up manner. Consider a pixel. Each valid-list of the pixel falls into one of the following three categories:

1. There is no agent in the pixel and the salesperson may visit the pixel one or more times.
2. There is one agent in the pixel. If the salesperson visits the pixel they meet at the pixel's center.
3. Two or more agents pass through the pixel. The salesperson also visits the pixel. In one of the visits she arrives at the center of the pixel. In this case, each agent travels along a straight line from a portal of the pixel to the center of the pixel. Alternatively, an agent's route may be an empty route if the agent is already located at the center of the pixel.

In each case, the computation of the cost for each valid-list of the pixel can be done in polynomial time.

We now turn to the computation of the table's entries for the super-pixels of level  $i$ , assuming all valid-lists of super-pixels of level  $i + 1$  were computed. Let  $S$  be a level  $i$  super-pixel and consider a list of  $1, \dots, 6$  values for its portals. The list already fixes the entrances and exits on the boundary of  $S$ . The super-pixel  $S$  contains four level  $i + 1$  super-pixels, which have four boundaries internal to  $S$ , with a total of at most  $4m$  more portals. Each of these portals may be used in one out of the six ways, giving rise again to  $n^{O(1/\varepsilon)}$  possibilities. The cost for each possibility can be computed by using the values for the four  $i + 1$  level super-pixels previously obtained. Thus, we can find the minimal cost that corresponds to each list in  $O(n^{O(1/\varepsilon)})$  time.

For the top-level super-pixel (the bounding-box) we may only consider the list for which neither the salesperson nor an agent visit a portal. The last table update of level 0 produces the cost of a minimal portals-limited-solution.

### Makespan-Sales PTAS

1. For each subset  $S$  of participants of size up to  $3m^4$ , which includes the salesperson and contains a representative from each non-empty pixel:
  - (a) Find an optimal solution for  $S$  by conducting an exhaustive search.
  - (b) In each non-empty pixel apply a constant-approximation to all original participants of the pixel, where the salesperson is a representative of the pixel.
  - (c) Extend the partial solution of  $S$  to a solution for the original instance: when all the participants in  $S$  return to their pixels - simultaneously perform the solution found in step 1(b).
2. Return the minimal cost solution found

**Fig. 1.** A PTAS for the ROUNDTRIP version of MIN-MAKESPAN SALES EUCLIDEAN-CTSP. The parameter  $m$  is assumed to be  $\lceil 1/\varepsilon \rceil$ .

## 2.2 MIN-MAKESPAN EUCLIDEAN-CTSP

We next present a simple PTAS for the ROUNDTRIP version of MIN-MAKESPAN SALES EUCLIDEAN-CTSP. A PTAS for the corresponding FULL-COOPERATION problem can be obtained by similar means. We note that the corresponding PURCHASE problem is polynomial-time solvable, as there is always an optimal solution in which all participants meet at a single point.

The PTAS for the two dimensional case appears in Figure 1. The generalization to any fixed dimension is straightforward.

**Theorem 2.** *The ROUNDTRIP version of MIN-MAKESPAN SALES EUCLIDEAN-CTSP admits a PTAS. The running time of the PTAS is  $O(n + f(\varepsilon))$ , where  $\varepsilon > 0$  is an arbitrarily small constant,  $f(\varepsilon)$  depends only on  $\varepsilon$ , and  $n$  is the number of participants.*

A constant approximation algorithm for the PATH version of this problem appears in [ABF<sup>+</sup>02]. The solution found by their algorithm is also  $O(1)$  times the diameter (the maximal distance between any two points) of the input. One can adapt this approximation to the ROUNDTRIP version by returning each participant to its origin. The cost of the resulting solution is at most twice the original solution. Since an optimal solution to the PATH version costs less than an optimal solution for the corresponding ROUNDTRIP version, this heuristic is a constant approximation for the ROUNDTRIP version.

We assume, w.l.o.g. that the instance lies inside  $[0, 1]^2$  and has an optimal cost of at least  $1/2$ . Let  $m = \lceil 1/\varepsilon \rceil$ . We divide the unit square  $[0, 1]^2$  into  $m^2$  pixels. I.e., a pixel is a square of the form  $[\frac{j}{m}, \frac{j+1}{m}] \times [\frac{j'}{m}, \frac{j'+1}{m}]$ , where  $j, j' = 0, 1, \dots, m-1$ . The PTAS for the SALES version relies on the next lemma:

**Lemma 3.** *Let  $I$  be an instance of  $n$  participants with an optimal makespan of  $OPT$ . Then, there exists an instance  $S \subseteq I$  with at most  $3m^4$  participants,*

in which each non-empty pixel in  $I$  is also non-empty in  $S$  and the optimal makespan of  $S$  is at most  $(1 + O(\varepsilon))OPT$ .

*Proof.* We may assume, w.l.o.g., that no two participants in  $I$  are located at the same point and that no three participants lie on a straight line. Otherwise, we can perturb each participant's location by at most  $\varepsilon/n$  and obtain an instance with an optimal cost of at most  $(1 + O(\varepsilon))OPT$ .

Let  $\pi$  be an optimal solution to  $I$ . We define the *sales-tree* of  $\pi$  to be a directed graph in which the nodes are the locations of the participants and there is a directed edge from  $u$  to  $v$  if a participant traveled from  $u$  to  $v$  in  $\pi$ . Since no two participants are located at the same point and no three participants lie on a straight line the in-degree of every node is one and the out-degree is at most two. We prune the sales-tree of  $\pi$  by iteratively removing leaves: we remove a leaf  $u$  if there exists another node in the sales-tree which resides in the same pixel as  $u$ . At the end of the process we are left with at most  $m^2/2$  leaves, and at most  $m^2/2$  nodes of degree 3 (in-degree plus out-degree). Note that the makespan of an optimal solution for the new instance, denoted  $\pi_0$ , is at most  $OPT$ . We now further decrease the number of participants by pruning some of the degree-2 vertices. We call a maximal set of participants along a path in which all the nodes are of degree 2 a *chain*. Clearly, each chain ends with a degree 3 node or a leaf. Hence, there are at most  $m^2$  chains. For each chain, and a pixel it intersects with, we intend to keep at most two nodes (participants). All the other nodes are removed from the chain. For a given pixel and a chain, the two participants that we keep are the first and the last (of this chain, inside the pixel) who receive the goods. We call such nodes a *beginner* node and an *ender* node, respectively. Note that, we are left with at most  $2 \cdot m^2$  participants per chain, giving rise to at most  $2m^4$  nodes of degree 2.

The new instance constructed, denoted  $S$  has at most  $3m^4$  participants. We next show that

*Claim.* There exists a solution  $\pi_S$  for  $S$  of cost at most  $OPT + O(\varepsilon)$ .

*Proof.* Recall that  $\pi_0$  (the optimal solution after pruning the leaves) is of cost at most  $OPT$ . We construct the solution  $\pi_S$  from  $\pi_0$  as follows: each participant of a beginner node travels along the corresponding original chain until it reaches the corresponding ender node, and then travels back to its starting location. All other participants travel along the same route they travel in  $\pi_0$ . Since all the non-beginner participants travel the way they do in  $\pi_0$ , they arrive to their original location by the time  $OPT$ . Beginner participants may be delayed by the time it takes to travel from the corresponding ender node back to their original location. This is at most the time it takes to cross a pixel which is at most  $\sqrt{2}\varepsilon$ . Thus, the cost of an optimal solution to  $S$  is at most  $(1 + O(\varepsilon))OPT$ .

The correctness of the PTAS algorithm for the ROUNDTRIP version of MIN-MAKESPAN SALES EUCLIDEAN-CTSP can now be deduced:

*Proof.* (of Theorem 2) Let  $\pi$  be an optimal solution for the instance  $I$  and let  $S \subseteq I$  be an instance that satisfies the condition of Lemma 3. Clearly, the subset

**Hop-visit**( $G(V, E), v$ ):

1. Let  $G' = (V', E')$  be a weighted complete graph, where  $V' \subseteq V$  is the set of vertices which contain participants, and the edge-weights are the corresponding distances in  $G$ .
2. Compute a minimum-spanning-tree  $T$  of  $G'$ , rooted at the salesperson's vertex  $v$ .
3. The salesperson visits an arbitrary child, and doesn't move any further.
4. When an agent receives a delivery:
  - (a) If the agent has a *sibling* in  $T$  who hasn't received the delivery, then the agent visits such a sibling and one of that sibling's children.
  - (b) Otherwise, the agent visits a child of the sibling which was visited first (a child of the "eldest" sibling of that agent), if such a child exists.

**Fig. 2.** A 3-approximation algorithm for MIN-MAX SALES cTSP

of participants  $S$  is included in the enumeration of our algorithm. The cost of an optimal solution to  $S$ , which is  $(1 + \varepsilon)OPT$  is computed at stage 2(b) of our algorithm. The additional cost produced at stage 2(c) is at most a constant times the diameter of the pixel, which is  $O(\varepsilon)$ . Note that this is an additive  $O(\varepsilon)$  increase of the makespan, as after all the participants in  $S$  return to their pixels the delivery to the other participants is done in parallel. Hence, the total cost of the solution produced by our algorithm is at most  $(1 + O(\varepsilon))$  times the cost of  $\pi$ .

Finally, note that there are less than  $O(n^{O(m^4)}) = O(n^{O(1/\varepsilon^4)})$  sets of participants to enumerate on. For each such subset  $S$ , a solution is a sequence of at most  $2|S| - 1$  moves. This follows as in each move either a participant receives the delivery or a participant returns to its original location. In any case, each move can be represented as a pair of two of the original input locations. Hence, for a given subset  $|S|$ , the number of solutions the algorithm enumerates on is at most

$$\binom{|S|}{2}^{2|S|-1} = O\left(\binom{m^4}{2}^{O(m^4)}\right) = \left(\frac{1}{\varepsilon}\right)^{O(1/\varepsilon^4)}.$$

Thus, the algorithm is a PTAS and runs in time  $O\left(n + \left(\frac{1}{\varepsilon}\right)^{O\left(\frac{1}{\varepsilon^4}\right)}\right)$ .

### 3 cTSP in Graphs

In this section we present some of the algorithmic results for cTSP in graphs.

#### 3.1 MIN-MAX SALES cTSP

We present a simple constant approximation algorithm for MIN-MAX SALES cTSP in Figure 2.

**Coarse-Path**( $G(V, E), v, \varepsilon$ ):

1. For each ordered subset  $V' \subseteq V$  of size  $1 + \lfloor 1/\varepsilon \rfloor$  or less, which starts with  $v$ .
  - (a) For each  $u \notin V'$  that contains an agent, find its distance to a closest vertex in  $V'$ . Denote the maximal distance found by  $MaxDist(V')$ .
  - (b) Compute the sum of distances between pairs of consecutive vertices in  $V'$ , and denote it by  $Length(V')$ .
  - (c) Let  $Cost(V')$  be the maximum of  $Length(V')$  and  $MaxDist(V')$ .
2. Pick the ordered subset  $V'$  for which  $Cost(V')$  is minimal.
3. Return the following solution: The salesperson follows the shortest paths between the consecutive vertices of  $V'$ . Each of the agents meets the salesperson at a closest vertex to that agent in  $V'$ . The salesperson waits for all the agents who come to a certain vertex before moving to the next vertex.

**Fig. 3.** A PTAS for MIN-MAX PURCHASE cTSP.

**Theorem 3.** MIN-MAX SALES cTSP is 3-approximable.

*Proof.* We prove that Algorithm **Hop-visit** is a 3-approximation algorithm for this problem. Clearly, all the agents are visited. Each participant traverses at most three edges of the MST, which means that the cost of the solution is at most thrice the weight of the heaviest edge of the MST.

On the other hand, consider an optimal solution, and define  $G'' = (V', E'')$ , such that  $(u_1, u_2) \in E''$  iff the participant from  $u_1$  sold the goods to the participant from  $u_2$ , or vice versa. Let the weight of  $(u_1, u_2) \in E''$  in  $G''$  be the distance between  $u_1$  and  $u_2$  in  $G$ . The optimal cost is clearly at least the weight of the heaviest edge in  $E''$ , since selling to an agent requires traveling to this agent's vertex.

Note that  $G''$  is a connected subgraph of  $G'$ . It is well-known that an MST is lexicographically minimal, i.e., its heaviest edge is not heavier than that of any other spanning-tree or spanning connected-subgraph. Therefore, the cost of the solution found by the above algorithm is at most thrice the cost of an optimal solution.

### 3.2 MIN-MAX PURCHASE cTSP

We next present a simple PTAS, **Coarse-Path**, described in Figure 3.

**Theorem 4.** Algorithm **Coarse-Path** is a PTAS for MIN-MAX PURCHASE cTSP.

*Proof.* Clearly, the MIN-MAX cost of the solution returned by the algorithm is the minimal  $Cost(V')$  of the subsets it considers. We show that one of these subsets has  $Cost(V')$  of at most  $(1 + \varepsilon)$  times the optimum.

Consider an optimal solution to the problem  $\pi$ , in which the cost is  $OPT$ . Choose a subset of the vertices of the path traveled by the salesperson in the

following way. Start with vertex  $v$ , and then choose a vertex iff its distance from the previous vertex chosen is at least  $\varepsilon \cdot OPT$ . Clearly, at most  $1/\varepsilon$  vertices are selected. Denote this subset by  $V'$ . Note that  $Length(V') \leq OPT$ .

For each vertex  $u \notin V'$  that contains an agent, there is a vertex in  $V'$  at a distance of at most  $(1 + \varepsilon) \cdot OPT$ . This holds, since for each vertex  $w$  visited by the salesperson in  $\pi$ ,  $V'$  contains a vertex at a distance of at most  $\varepsilon \cdot OPT$  from  $w$ . Thus,  $Cost(V') \leq MaxDist(V') \leq (1 + \varepsilon)OPT$ . Therefore, Algorithm Coarse-Path indeed finds a  $(1 + \varepsilon)$ -approximate solution. The running-time of the algorithm is  $O((2n)^{\lfloor 1/\varepsilon \rfloor + 3})$ , since it enumerates over ordered subsets of vertices of size at most  $\lfloor 1/\varepsilon \rfloor$ , and the required computation for each ordered subset takes at most  $O(n^3)$  time. Thus, **Coarse-Path** is a PTAS.

## 4 Discussion and Open Problems

We obtained quite tight approximation and intractability results for most of the cTSP problems. Some of the cTSP problems turn out to be easier (in sense of approximation) than the classical TSP, while others are strictly harder.

The status of MIN-MAKESPAN SALES cTSP is not settled, as there is an  $O(\sqrt{\log n})$  approximation and a constant inapproximability factor. Improving the factors of this problem as well as tightening the factors for some others is yet to be achieved. It is also likely that the running time of some of the PTAS can be improved.

There are some disturbing asymmetries in the Euclidean results (see Table 2). For example, while the ROUNDTrip versions of MIN-SUM SALES and FULL-COOPERATION cTSP have a PTAS, the best approximations for the corresponding PATH-cTSP problems only guarantee some constant factors. We conjecture that these two PATH-cTSP versions indeed have a PTAS, but we suspect that this may not be very easy to prove. This follows since it can be shown that a PTAS for the first problem implies a (currently unknown) PTAS for the well-studied 3-bounded-degree-planar MINIMUM SPANNING TREE (e.g., [PV84, KRY96, FKK<sup>+</sup>97, Cha03, AC04]).

## Acknowledgements

We would like to thank Joseph Mitchell and Uri Zwick for many fruitful discussions.

## References

- [ABF<sup>+</sup>02] E. M. Arkin, M. A. Bender, S. P. Fekete, J. S. B. Mitchell, and M. Skutella. The Freeze-Tag Problem: How to Wake Up a Swarm of Robots. In *Proc. of SODA'02*, pages 568–577, 2002.
- [ABG<sup>+</sup>03] E. M. Arkin, M. A. Bender, D. Ge, S. He, and J. S. B. Mitchell. Improved Approximation Algorithms for the Freeze-Tag Problem. In *SPAA'03*, pages 295–303, 2003.

- [AC04] S. Arora and K. L. Chang. Approximation Schemes for Degree-Restricted MST and Red-Blue Separation Problems. *Algorithmica*, 40(3):189–210, 2004.
- [AH94] E. Arkin and R. Hassin. Approximation Algorithms for the Geometric Covering Salesman Problem. *Discrete Applied Math.*, 55:197–218, 1994.
- [Aro98] S. Arora. Polynomial-time Approximation Schemes for Euclidean TSP and other Geometric Problems. *Journal of the ACM*, 45(5):753–782, 1998.
- [BNGNS98] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Multicasting in Heterogeneous Networks. In *Proc. of STOC'98*, pages 448–453, 1998.
- [Cha03] T. M. Chan. Euclidean Bounded-Degree Spanning Tree Ratios. In *Proc. 19th ACM SoCG*, pages 11–19, 2003.
- [Chr76] N. Christofides. Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem. Technical report, Graduate School of Industrial Administration, Carnegie–Mellon University, 1976.
- [dBGK<sup>+</sup>05] M. de Berg, J. Gudmundsson, M. J. Katz, C. Levcopoulos, M. H. Overmars, and A. F. van der Stappen. TSP with Neighborhoods of Varying Size. *Journal of Algorithms*, 57:22–36, 2005.
- [DM01] A. Dumitrescu and J. S. B. Mitchell. Approximation Algorithms for TSP with Neighborhoods in the Plane. In *Proc. of SODA'01*, pages 38–46, 2001.
- [EK01] L. Engebretsen and M. Karpinski. Approximation Hardness of TSP with Bounded Metrics. In *Proc. of ICALP'01*, pages 201–212, 2001.
- [FKK<sup>+</sup>97] S. P. Fekete, S. Khuller, M. Klemmstein, B. Raghavachari, and N. Young. A Network Flow Technique for Finding Low-Weight Bounded-Degree Trees. *Journal of Algorithms*, 24:310–324, 1997.
- [HHL88] S. M. Hedetniemi, S. T. Hedetniemi, and A.L. Liestman. A Survey of Gossiping and Broadcasting in Communication Networks. *Networks*, 18(4):319–359, 1988.
- [KLS04] J. Knemann, A. Levin, and A. Sinha. Approximating the Degree-Bounded Minimum Diameter Spanning Tree Problem. *Algorithmica*, 41(2):117–129, 2004.
- [KRY96] S. Khuller, B. Raghavachari, and N. Young. Low Degree Spanning Trees of Small Weight. *SIAM Journal of Computing*, 25(2):355–368, 1996.
- [Mit99] J. S. B. Mitchell. Guillotine Subdivisions Approximate Polygonal Subdivisions: Part II – A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal of Computing*, 28(4):1298–1309, 1999.
- [Mit06] J. S. B. Mitchell. A PTAS for TSP with Neighborhoods Among Fat Regions in the Plane. Private communication, 2006.
- [MM95] C. Mata and J. S. B. Mitchell. Approximation Algorithms for Geometric Tour and Network Design Problems. In *SCG'95*, pages 360–369, 1995.
- [PV84] C. H. Papadimitriou and U. V. Vazirani. On Two Geometric Problems Related to the Traveling Salesman Problem. *J. of Alg.*, 5:231–246, 1984.
- [Rav94] R. Ravi. Rapid Rumor Ramification: Approximating the Minimum Broadcast Time. In *Proc. of FOCS'94*, pages 202–213, 1994.
- [SABM02] M. O. Sztainberg, E. M. Arkin, M. A. Bender, and J. S. B. Mitchell. Analysis of Heuristics for the Freeze-Tag Problem. In *Proc. of SWAT'02*, pages 270–279, 2002.
- [SS05] S. Safra and O. Schwartz. On the Complexity of Approximating TSP with Neighborhoods and Related Problems. *Computational Complexity*, 14:281–307, 2005.