Theoretical Computer Science 411 (2010) 2847–2863

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Cooperative TSP[☆]

Amitai Armon [a], Adi Avidor [a], Oded Schwartz [b,*,1]

[a] School of Computer-Science, Tel-Aviv University, Tel-Aviv 69978, Israel

[b] Department of Mathematics, TU-Berlin, D-10623, Berlin, Germany

ARTICLE INFO

Article history:
Received 26 August 2009
Received in revised form 26 March 2010
Accepted 7 April 2010
Communicated by D.-Z. Du

Keywords:
Algorithms
Approximation algorithms
TSP
Cooperative TSP

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

All of 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 of 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 the 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 of the agents eventually receive the delivery. We next elaborate on the various cooperation modes, the cost of solutions and other parameters effecting the cTSP.

☆ A preliminary version of this paper appeared in ESA06 Armon et al. (2006) [4].

* Corresponding author. Fax: +49 30 314 79706.
E-mail addresses: amitai.armon@cs.tau.ac.il (A. Armon), adi.avidor@cs.tau.ac.il (A. Avidor), odedsc@math.tu-berlin.de, oded.schwartz@gmail.com (O. Schwartz).
URLs: http://www.cs.tau.ac.il/research/amitai.armon/ (A. Armon), http://www.cs.tau.ac.il/research/adi.avidor/ (A. Avidor), http://www.math.tu-berlin.de/~odedsc/ (O. Schwartz).

1 Part of this work was done while the author was at Tel-Aviv University.

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2010.04.016

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 travel of all of the participants (MIN–SUM), minimizing the maximal travel of any participant (MIN–MAX), and minimizing the total time until the sales process ends (MIN–MAKESPAN). Naturally, the MIN–SUM goal is motivated by scenarios in which the travel of all of the participants is covered by the same entity (e.g., the delivery-service company), which is therefore interested in minimizing the total travel. The MIN–MAX objective is required, for example, when there is a bound on the amount of fuel/battery of each participant, and each of them should spend as little energy as possible on this delivery process. MIN–MAKESPAN is simply motivated by cases in which the completion of the deliveries is urgent.

Metric space. We consider Cooperative TSP in any fixed dimension Euclidean space and in non-negative weighted undirected graphs. Note that w.l.o.g., we may assume that the graph is complete and that the weights of all edges satisfy the triangle inequality, hence this is a finite metric space.

Roundtrip vs. path. We consider the ROUNDTRIP versions, in which all participants are required to return to their initial location, and the PATH versions in which there is no such requirement.

In this paper we consider all the problems arising from combining cooperation modes, goal functions, graph/Euclidean space and PATH/ROUNDTRIP versions. We refer to each of the problems we study using the format: GOAL-FUNCTION-COOPERATION-MODE-[EUCLIDEAN]-cTSP (e.g., "MIN–SUM PURCHASE EUCLIDEAN cTSP"). Unless explicitly stated otherwise, a problem name indicates its PATH version (rather than its ROUNDTRIP version).

1.1. Related studies

The classical TSP problem remains NP-hard even in the special planar variant [17,28]. However, there is a PTAS for any fixed dimension Euclidean space [5,25]. When only metric space is assumed, the best known approximation algorithm yields a $\frac{3}{2}$-approximation ratio [9] and an inapproximability factor of $\frac{131}{130}$ was shown [15].

The Freeze-Tag problem. The Freeze-Tag problem was first suggested and studied by Arkin et al. in [2]. The 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. [2] 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., [34,3,22]) culminating with an $O(\sqrt{\log n})$-approximation for the general weighted graph case [22].

TSP with neighborhoods. 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. [1], followed by quite a few papers (e.g., [23,18,14,11,33,26]). An instance of TSP with Neighborhoods may reside in a weighted graph or in a Euclidean space. The 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. Also, in TSP with Neighborhoods there is an upper bound on the neighborhood size.

Other cooperative multi-agents routing problems. As noted in [2], 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 related to Cooperative TSP (see [19] for a survey and [30,7] 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 [2] for a list of relevant papers). Other researches 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 a wireless sensor network, and ants behavior inspired algorithms; see [2] for a list of relevant papers).

1.2. Our contribution

We consider all combinations of cooperation modes, goal functions, PATH/ROUNDTRIP and graph/Euclidean versions. The results for cTSP in weighted graphs are summarized in Table 1 and the results for cTSP in a fixed dimension Euclidean space are summarized in Table 2. We obtain constant approximations for most of the problems, PTAS for many of the others and polynomial time exact solutions for a few. On the intractability side we obtain NP-hardness proofs and inapproximability factors for all the NP-hard graph problems and for some of the Euclidean problems.

Table 1
Approximation ratios vs. inapproximability ratios for cTSP in weighted Graphs. Next to each result the corresponding subsection appears. (∗) is by [22] and (∗∗) is by [2]. The parameter ε stands for an arbitrarily small positive constant, or for a positive function that tends to zero as the input size increases.

cTSP in graphs							
		Purchase cooperation		Sales cooperation		Full cooperation	
	Goal	Approx.	Inapprox.	Approx.	Inapprox.	Approx.	Inapprox.
Path	Min Sum	$2 + \ln 3$ 3.1	NP-hard 3.1	2 3.1	NP-hard 3.1	$2 + \ln 3$ 3.1	APX-hard 3.1
	Min Max	PTAS 3.2.1	no FPTAS 3.2.2	3 3.2.2	$2 - \varepsilon$ 3.2.3	4 3.2.3	$2 - \varepsilon$ 3.2.3
	Min Makespan	Polynomial 3.3.1		$O(\sqrt{\log n})$ ∗	$\frac{5}{3} - \varepsilon$ ∗∗	2 3.3.3	$2 - \varepsilon$ 3.3.3
Round Trip	Min Sum	$\frac{3}{2}$ 3.1	$\frac{131}{130} - \varepsilon$ 3.1	$\frac{3}{2}$ 3.1	$\frac{131}{130} - \varepsilon$ 3.1	$\frac{3}{2}$ 3.1	$\frac{131}{130} - \varepsilon$ 3.1
	Min Max	PTAS 3.2.1	no FPTAS 3.2.1	3 3.2.2	$\frac{3}{2} - \varepsilon$ 3.2.2	2 3.2.3	$2 - \varepsilon$ 3.2.3
	Min Makespan	Polynomial 3.3.1		$O(\sqrt{\log n})$ 3.3.2	$\frac{5}{4} - \varepsilon$ 3.3.2	2 3.3.3	$2 - \varepsilon$ 3.3.3

Table 2
Approximation ratios for cTSP in any fixed dimension Euclidean space. Next to each result the corresponding subsection appears. (∗) is by [2]. The parameter ε stands for an arbitrarily small positive constant, or for a positive function that tends to zero as the input size increases.

Euclidean cTSP				
	Goal	Purchase cooperation	Sales cooperation	Full cooperation
Path	Min–Sum	PTAS 2.1.1	$\frac{5}{3} + \varepsilon$ 2.1.2	$2 + \varepsilon$ 2.1.3
	Min–Max	PTAS 2.2.1	3 2.2.2	4 2.2.3
	Min–Makespan	Polynomial 2.3	PTAS ∗	PTAS 2.3.1
Roundtrip	Min–Sum	PTAS 2.1	PTAS 2.1	PTAS 2.1
	Min–Max	PTAS 2.2.1	3 2.2.2	2 2.2.3
	Min–Makespan	Polynomial 2.3	PTAS 2.3.1	PTAS 2.3.1

Paper organization. We present the results for Euclidean cTSP in Section 2, and the results for cTSP in weighted graphs in Section 3. Each section is divided into three subsections, one for each of the goal functions (Min–Sum, Min–Max and Min–Makespan, in this order). Furthermore, each subsection is divided according to cooperation modes (Purchase, Sales and Full-Cooperation, in this order). Finally, open problems are discussed in Section 4.

2. Euclidean cTSP

This section presents the results we obtained for the various Euclidean cTSP problems.

2.1. Min–Sum Euclidean-cTSP

In this subsection we consider the various objectives for the path versions of Min–Sum Eucli-dean-cTSP. It is not hard to see that all the roundtrip versions, i.e., with either of the three cooperation modes, are identical to the classical TSP problem. (This is explained in detail in Appendix B, Claim B.1.) Thus, like the classic TSP problem, these roundtrip problems are all NP-hard [17,28], and have a PTAS for any fixed dimension Euclidean space [5,25].

2.1.1. Min–Sum Purchase Euclidean-cTSP

We next provide a PTAS for Min–Sum Purchase Euclidean-cTSP. Note that the problem is NP-hard even for the planar case. This follows, since an instance of the classical planar TSP can be reduced to an instance of Min–Sum Purchase Euclidean-cTSP by simply replacing each customer with three agents. This makes the salesperson the only participant who moves, since it is always cheaper for the salesperson to approach a group of three agents than the other way round.[2]

[2] It is actually sufficient to replace each customer with two agents rather than three. Any solution with agents moving, can be transformed to a solution where only the salesperson moves: w.l.o.g., each pair of agents from the same location move together. Then replace their traversal with a traversal of the salesperson back and forth.

The algorithm and analysis below use Arora's technique for the PTAS of Euclidean TSP [5]. Our algorithm differs from Arora's algorithm in that it has to consider *all* the agents' paths and not only the salesperson's path. We show how this can be done while keeping the dynamic programming polynomial. We get:

Theorem 2.1. Min–Sum Purchase Euclidean-cTSP *admits a PTAS.*

We describe the PTAS for the planar case. The extension to any fixed dimension is straightforward. Roughly speaking, we prove the existence of a coarse solution, which is called a minimal cost *portals-limited solution*, that has a cost of at most $(1+\varepsilon)$ the cost of an optimal solution. We then show how to find a minimal cost portals-limited solution in polynomial time, using dynamic programming. We start by introducing the terminology. Readers familiar with Arora's PTAS for Euclidean TSP may want to skip to the (slightly altered) definition of portals-limited solutions.

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 $L = 2^{3+\lceil 2\log n\rceil}$ (the smallest power of 2 that is at least $8n^2$). By stretching and shifting the input points we may assume, without loss of generality, that all of 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. Thus, each super-pixel of level-$(\log L - 1)$ is a pixel and the super-pixel of level-0 is the entire bounding box. Additionally, note that different super-pixels of the same level may overlap only at their boundaries, and that each super-pixel of level-i contains four super-pixels of level-$(i+1)$, for $i = 0, \dots, \log L - 2$. Clearly, the total number of super-pixels is polynomial in n. From now on we consider, without loss of generality, only instances for which all of the participants are located at pixel centers. This is true since any optimal solution of a general instance, can be changed by instructing each participant to move to the nearest pixel center. This increases the cost of the solution by at most $n \cdot \sqrt{2}$. As $OPT > L/4 \geq 2n^2$, the increase is at most OPT/n, which is less than $\varepsilon/2 \cdot OPT$, for a sufficiently large n.

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 $\{(x+a, y+b) \mid (x, y) \in A\}$. In particular, we are interested in the (a, b) shift of the original instance, the (a, b)-*shifted instance*, which by our choice of parameters lies inside the bounding box $[0, L]^2$.

Portals. Let m be the unique power of 2 in the interval $[\frac{8\sqrt{2}\log L}{\varepsilon}, \frac{16\sqrt{2}\log L}{\varepsilon})$. Note that $m = O(\frac{\log n}{\varepsilon})$. 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 at portals, each of which she visits 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 in the same 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 that connect portals to portals and centers of pixels to portals. Additionally, in an optimal portals-limited solution, tours of two agents do not cross.

Using the above notations, our PTAS relies on the following two Lemmata:

Lemma 2.2. *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 minimum-cost portals-limited solution of the (a, b)-shifted instance, is at most $(1+\varepsilon)\cdot OPT$.*

Lemma 2.3. *A minimal cost portals-limited solution can be found in time polynomial in n.*

The PTAS enumerates over all $O(L^2)$ values of (a, b) pairs. For each pair it applies Lemma 2.3 to find a minimal cost portals-limited solution. Finally, it outputs the cheapest solution found, which according to Lemma 2.2, 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.

The proof of Lemma 2.3 explains how to consider both the salesperson's and the other agents' paths, while keeping the time polynomially bounded.

Proof of Lemma 2.3. 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 assure 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 of 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 [5].

We now describe the construction in a bottom-up manner. Consider a pixel. Each valid-list of the pixel may fall into one 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. To see that this computation is indeed polynomial, note the following. If there is no agent in the pixel (not even visiting the pixel), then the cost is only due to the salesperson, and is computed exactly as in Arora's algorithm. If an agent passes through the pixel, or starts in a pixel and leaves it, then her cost is just the distance from the entry/start point to the exit point. The only subtle case is when the salesperson meets one or more agents in the center of the pixel. In this case, the cost accounted for each agent is the distance from its starting point in the pixel to the center of the pixel. The cost of the salesperson is computed almost the same as in Arora's algorithm: we iterate through all non-crossing traversals of the salesperson which agree with the entry and exit points given by the valid list. The only change is that for each such traversal, which may contain more than one leg in the pixel, we have to decide which leg is changed to visit the agent(s) on the center of the pixel. This introduces at most polynomial increase in the running 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 $1, \ldots, 6$ 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 that 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 level-$(i+1)$ super-pixels previously obtained. Thus, we can find the minimum-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 minimum-cost portals-limited solution. □

The proof of Lemma 2.2 mainly follows arguments from the PTAS of Euclidean TSP, and appears in Appendix A for completeness.

2.1.2. Min–Sum Sales Euclidean-cTSP

We obtain a $5/3 + \varepsilon$ approximation for this problem, and improve the ratio to $3/2 + \varepsilon$ for its planar version (for an arbitrarily small $\varepsilon > 0$). We also prove NP-hardness, even for the planar case.

Lemma 2.4. *Consider an instance of* Min–Sum Sales Euclidean cTSP *where there is no more than one participant in any single point, and there are no three participants on the same straight line. Solving it is equivalent to finding a bounded-degree minimum spanning tree, spanning the initial locations of the participants, where the degree-bound is 1 for the salesperson's tree-node and 3 for all of the other nodes.*

Proof. Consider an optimal solution for Min–Sum Sales Euclidean cTSP. Clearly, the participants move in straight lines between the initial locations of agents, to sell them the goods. We assume w.l.o.g. that *all* the intersections between the participants' routes occur in initial locations of agents (if the routes of two agents intersect at another location, we can switch between them and thus lower the cost of the solution). We can also assume w.l.o.g. that any initial location of an agent is only visited once in an optimal solution.

Thus, in an optimal solution, the collection of the routes used by the participants forms a spanning tree of their initial locations. The degrees of the spanning tree are bounded by 3, since at most one participant enters a tree-node and at most two leave it. The node corresponding to the salesperson must have degree 1.

On the other hand, any such bounded-degree spanning tree produces a solution to our problem. Such a solution can be obtained by simply directing the edges of the spanning tree from parent to child and letting the participants follow these directed edges, starting with the salesperson (such that a single participant traverses each tree-edge). Therefore, finding a minimum spanning tree that satisfies these degree-constraints is equivalent to solving our problem in this case. □

Algorithm Sales-Bounded-MST:

1. Compute a minimum spanning tree, spanning all the agents' initial locations (not including the salesperson), such that its degrees are bounded by 5.
   Let p be the location of the salesperson, and let q be the location of the agent closest to her. Let r be the location of one of the agents connected to q in the above tree.
   Remove the edge (q, r).
2. Transform the tree containing r into a tree with degree at most 3, such that the degree of r is 1.
3. Transform the tree containing q, into a tree with degree at most 3, such that the degree of q is 1.
4. Restore the edge (q, r) and connect q to p. Output the resulting tree.

Fig. 1. A 3/2-approximation algorithm for a 3-bounded degree tree that spans the participants' locations and has salesperson's node of degree 1.

Corollary 2.5. MIN–SUM SALES EUCLIDEAN cTSP *can be approximated within a factor of* $5/3 + \varepsilon$, *for any* $\varepsilon > 0$.

Proof. We slightly perturb the input locations, such that they satisfy the conditions of Lemma 2.4. Khuller et al. [21] showed that a minimum spanning tree in any fixed dimension Euclidean space can be modified to satisfy the degree-constraints we require (1 for a pre-specified node and 3 for the others), while increasing its weight by a factor of at most 5/3. Thus, the Corollary follows. □

For the planar case, we manage to improve the approximation ratio to $3/2 + \varepsilon$:

Theorem 2.6. MIN–SUM SALES PLANAR-cTSP *can be approximated within a factor of* $3/2 + \varepsilon$, *for any* $\varepsilon > 0$.

Proof. By slightly perturbing the initial locations of the participants, we can assume that the assumptions of Lemma 2.4 hold (this increases the optimal cost by a factor of $1 + \varepsilon$, for an arbitrarily small $\varepsilon > 0$). We look for the optimal solution of this slightly perturbed input, i.e., we look for a minimum spanning tree satisfying the degree-constraints stated in Lemma 2.4. We find an approximate solution using the algorithm **Sales-Bounded-MST** of Fig. 1.

The first stage of the algorithm can easily be performed in polynomial time [27]. Note that connecting p and q right after this stage would have yielded a minimum spanning tree that spans all of the initial locations. Stages 2 and 3 are performed using the $\frac{3}{2}$-approximation algorithm of Khuller et al. [21], which requires that the degree of each node will be at most 5 and the degree of the root will be at most 4. Thus, each of these stages increases the weight of the transformed subtree by a factor of at most 3/2 [21].

So all in all, we obtain a spanning tree that satisfies the degree bounds and has a weight of at most 3/2 times a minimum spanning tree, which also means at most 3/2 times the cost of an optimal solution.

As was explained in Lemma 2.4, the participants can now follow the edges of this tree (rooted at p with edges directed from parent to child), and form a solution whose cost is the cost of the tree. Thus, the theorem follows. □

Claim 2.7. MIN–SUM SALES PLANAR-cTSP *is NP-hard.*

Proof. Finding the minimum spanning tree whose degree is bounded by 3 is NP-hard [29]. We note that the proof of [29] holds even if it is guaranteed that no three points of the input lie on a straight line (since the input points can be slightly perturbed in their construction). Requiring that a certain node will be a leaf clearly remains *NP-hard* (by solving the problem for all of the possible locations of a leaf one can find the solution for the problem without this requirement).

Since according to Lemma 2.4 this bounded-MST problem can be easily reduced to our problem (with the same input, where the salesperson is at the point that is required to be a leaf), our problem is NP-hard. □

2.1.3. MIN–SUM FULL-COOPERATION EUCLIDEAN-cTSP

A $(1+\varepsilon)$-approximate minimum Steiner tree, spanning all of the participants' initial locations, can be computed by using the PTAS of Arora [5]. Clearly, by letting the salesperson tour the Steiner tree we obtain a $(2 + 2\varepsilon)$-approximate solution for our problem. Thus,

Corollary 2.8. MIN–SUM FULL-COOPERATION EUCLIDEAN-cTSP *can be approximated within a factor of* $2 + \varepsilon$, *for any* $\varepsilon > 0$.

2.2. MIN–MAX EUCLIDEAN-cTSP

2.2.1. MIN–MAX PURCHASE EUCLIDEAN-cTSP

We first show that both the PATH and the ROUNDTRIP versions of MIN–MAX PURCHASE EUCLI-DEAN-cTSP have a PTAS. We do so by manipulating the input instance such that it fits the PTAS for the graph version of the problem (see Section 3.2.1).

Claim 2.9. *The* ROUNDTRIP *and* PATH *versions of* MIN–MAX PURCHASE EUCLIDEAN-cTSP *admit a PTAS.*

Proof. Consider an instance of the PATH 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 $m = \frac{n}{\varepsilon'}$, where n is the number of participants and ε' is a parameter to be determined later. 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 $j, j' = 0, 1, \ldots, m-1$. We consider a slightly changed input, where each participant is located in the center of its pixel. This instance can be approximated using **Coarse-Path**(G, W, v, ε'') — the PTAS for the corresponding graph variant of the problem (see Section 3.2.1) as follows. Let G be a complete graph, with the m^2 pixels as vertices. Let $W(e)$, the weight of each edge, be the Euclidean distance between the corresponding pixels' centers. Let v be the pixel that contains the salesperson and let ε'' be a sufficiently small constant (to be determined shortly).

The solution for the altered instance is amended into a solution for the original instance by adding legs between the original location of a participant and the center of its pixel (each leg is of length at most $\frac{\sqrt{2}}{2m} = \frac{\sqrt{2}\varepsilon'}{2n}$).

Denote by OPT the optimal solution for the original instance, by OPT' the optimal solution for G, by ALG' the output of the PTAS for G, and by ALG the output of the whole algorithm. Both OPT and OPT' consist of at most $2n$ segments (one for each agent and at most n for the salesperson). Therefore, OPT' is at most $2n \cdot \frac{\sqrt{2}\varepsilon'}{2n} = \sqrt{2}\varepsilon'$ larger than OPT. Similarly, ALG is at most $2\sqrt{2}\varepsilon'$ larger than ALG'. Thus,

$$\begin{aligned} ALG \le ALG' + 2\sqrt{2}\varepsilon' &\le (1+\varepsilon'')OPT' + 2\sqrt{2}\varepsilon' \le (1+\varepsilon'')(OPT + \sqrt{2}\varepsilon') + 2\sqrt{2}\varepsilon' \\ &\le (1+\varepsilon'')OPT + 4\sqrt{2}\varepsilon' \end{aligned}$$

The running time of the algorithm is dominated by the running time of **Coarse-Path**. Thus, it is $O\left(\left(n/\varepsilon'\right)^{(2/\varepsilon'')+6}\right)$. Choose $\varepsilon' = 1/\lg n$ and $\varepsilon'' = \varepsilon - 12/\lg n$ (where n is assumed to be sufficiently large, so that ε'' is positive). Since $1/2 \le OPT$ we obtain that $ALG \le (1+\varepsilon)OPT$ and the running time is $O\left((n \lg n)^{2/(\varepsilon - 12/\lg n)+6}\right)$.

The same arguments yield a PTAS for the ROUNDTRIP version with the same running time. □

2.2.2. MIN–MAX SALES EUCLIDEAN-cTSP

Claim 2.10. MIN–MAX SALES EUCLIDEAN-cTSP *has a* 3*-approximation algorithm for both the* PATH *and the* ROUNDTRIP *versions of the problem.*

Proof. We consider the complete graph whose vertices are the initial locations of the participants and whose edge weights are the distances. We solve the problem for that graph using algorithm **Hop-Visit**, described for graphs in Section 3.2.2. The same analysis holds here as well. □

2.2.3. MIN–MAX FULL-COOPERATION EUCLIDEAN-cTSP

Similarly to Claim 2.10, we can obtained approximation ratios by considering the complete graph whose vertices are the initial locations of the participants and whose edge weights are the distances. We use here the algorithm described in Section 3.2.3, and the same analysis holds here as well. We thus have the following Corollaries:

Corollary 2.11. MIN–MAX FULL-COOPERATION EUCLIDEAN-cTSP *has a* 4*-approximation algorithm.*

Corollary 2.12. *The* ROUNDTRIP *version of* MIN–MAX FULL-COOPERATION EUCLIDEAN-cTSP *has a* 2*-approximation algorithm.*

2.3. MIN–MAKESPAN EUCLIDEAN-cTSP

In this Subsection we mainly present a simple PTAS for the ROUNDTRIP version of MIN–MAKESPAN SALES EUCLIDEAN-cTSP. A PTAS for the corresponding FULL-COOPERATION problem, in both the PATH and ROUNDTRIP versions, can be obtained by similar means.

In addition, we note that both the PATH and the ROUNDTRIP versions of the corresponding PURCHASE problem, namely MIN–MAKESPAN PURCHASE EUCLIDEAN-cTSP, are polynomial time solvable (in fact, linear time for fixed dimension). This can be observed as follows. For both the PATH and the ROUNDTRIP versions any optimal solution can be modified to an optimal solution in which all participants meet at a single point. For the PATH version, the modification can be done by letting all of the participants meet the salesperson at the last point she visits. For the ROUNDTRIP version, denote the value of an optimal solution by OPT. Then, the modification of the optimal solution can be done by letting all of the participants meet at the point where the salesperson resides at time $OPT/2$ (afterwards, all participants return to their initial location). Hence, for both the PATH and the ROUNDTRIP cases, the single meeting point is the center of the enclosing sphere, and can thus be found in polynomial time (see for example [24,35,12]).

A PTAS for the PATH version of MIN–MAKESPAN SALES EUCLIDEAN-cTSP has been obtained by Arkin et al. [2]. We next present a PTAS for the ROUNDTRIP version of this problem.

Makespan-Sales PTAS

1. For each subset S of the participants of size up to $3m^4$ that 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 optimal solution of S to a solution for the original instance: when all of the participants in S return to their pixels - simultaneously perform the solutions found in step 1(b).
2. Return the minimal cost solution found

Fig. 2. A PTAS for the ROUNDTRIP version of MIN–MAKESPAN SALES EUCLIDEAN-cTSP. The parameter m is assumed to be $\lceil 1/\varepsilon \rceil$.

2.3.1. *The* ROUNDTRIP *version of* MIN–MAKESPAN SALES EUCLIDEAN-cTSP

This problem seems quite close in nature to the corresponding PATH version, and thus calls for a similar PTAS. However, note that converting an optimal solution for the PATH version into a solution for the ROUNDTRIP version (by simply letting all of the participants return) only guarantees a $(2+\varepsilon)$-approximation for this problem. This is true since a participant's way back may double the makespan, while the ROUNDTRIP version may have a solution with the same makespan as the PATH version. Thus, constructing a PTAS for the ROUNDTRIP version requires considering in advance that the participants should return to their initial locations. We therefore use a different approach than the one used by [2] for the PATH version of the problem.

We show the PTAS for the two dimensional case (see Fig. 2). The generalization to any fixed dimension is straightforward.

Theorem 2.13. *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 ε, and n is the number of participants.*

A constant-factor approximation algorithm for the PATH version of this problem appears in [2]. 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 simply 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 $j, j' = 0, 1, \ldots, m-1$.

The PTAS for the SALES version relies on the next lemma:

Lemma 2.14. *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 ε/n. An optimal solution to the perturbed instance has a cost of at most $OPT + O(\varepsilon)$ (as the cost increase per participant is at most $\frac{2\varepsilon}{n}$). Since $OPT \geq 1/2$ this cost is less than $(1+O(\varepsilon))OPT$.

We show how we can remove participants from I while keeping the cost of an optimal solution to be at most $OPT+O(\varepsilon)$. Let π be an optimal solution to I. We define the *sales-tree* of π 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 π. Since no two participants are located at the same point and no three participants lie on a straight line, the in-degree of each node is one and the out-degree is at most two. We prune the sales-tree of π by iteratively removing leaves (nodes of out-degree zero): we remove a leaf u if there exists another node in the sales-tree that 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 π_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 of 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, we intend to keep at most two nodes (participants). All of 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. Since there are at most $m^2/2$ leaves and at most $m^2/2$ nodes of degree 3, the new instance constructed, denoted S, has at most $3m^4$ participants. We next show that

Claim 2.15. *There exists a solution π_S for S of cost at most $OPT + O(\varepsilon)$.*

Proof. Recall that π_0 (an optimal solution after pruning the leaves) is of cost at most OPT. We construct the solution π_S from π_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 π_0. Thus, 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 $(1+O(\varepsilon))OPT$. □

This concludes the proof of Lemma 2.14. □

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

Proof of Theorem 2.13. Let π be an optimal solution for the instance I and let $S \subseteq I$ be an instance that satisfies the condition of Lemma 2.14. Clearly, the subset 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 1(b) of our algorithm. The additional cost produced at stage 1(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 of 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 π.

Finally, note that there are less than $O(n^{O(m^4)}) = O(n^{O(1/\varepsilon^4)})$ sets of participants to enumerate. For each such subset S, a solution is a sequence of at most $2|S|-1$ moves. This follows since, 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 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 $O\left(n + \left(\frac{1}{\varepsilon}\right)^{O(\frac{1}{\varepsilon^4})}\right)$. □

3. cTSP **in graphs**

In this section we present the algorithmic and hardness results for cTSP in graphs.

3.1. MIN–SUM cTSP

We start by considering cTSP with the MIN–SUM objective. For the PATH versions, we provide simple approximation algorithms and hardness results for each cooperation mode.

For the ROUNDTRIP versions, the corresponding PURCHASE, SALES and FULL-COOPERATION problems are all equivalent to the classical METRIC-TSP, and thus have the same approximation and hardness results (see Appendix B for more details). These observations do not hold for the other objectives, in which there is also a significant difference between the different cooperation modes.

We begin with the approximability results for the various PATH versions:

Claim 3.1. *Under the* MIN–SUM *objective,* PURCHASE *and* FULL-COOPERATION cTSP *have* $(2+\ln 3)$*-approximation algorithms. If each vertex contains a participant, the approximation ratio improves to* 2. MIN–SUM SALES cTSP *has a* 2*-approximation algorithm.*

Proof. For the first two problems, we simply find an approximate minimum *Steiner tree* that spans the vertices that contain participants, and the salesperson visits all of the agents by touring this tree (e.g., in an "infix-order"). The total distance traveled is twice the tree's weight. We use the approximation algorithm of [32] for the minimum Steiner tree problem, which has an approximation ratio of $1+\ln 3/2$ ($\simeq$1.55). Therefore, the distance traveled is at most $(2+\ln 3)$ times the weight of the minimum Steiner tree.

On the other hand, the edges used by any solution to these problems must form a connected subgraph that spans the vertices that contain participants (since all of the agents receive the goods). This means that the total distance traveled is at least the minimum Steiner Tree weight. Therefore, the simple algorithm described has an approximation ratio of $2+\ln 3$. If each vertex contains a participant, then a minimum spanning tree can be computed exactly. Thus, the approximation ratio in this case is 2.

For SALES cTSP it is again sufficient to compute a minimum spanning tree, since the goods can be delivered to an agent only at the original vertices. □

We next provide hardness results for each cooperation-mode. Similar to the Euclidean case, the PURCHASE version is NP-hard, since the classical PATH-TSP [20] can be reduced to an instance of MIN–SUM PURCHASE cTSP. The reduction is again done simply by replacing each customer with three agents. Thus, the salesperson is the only participant who moves.[3] Like TSP, the PATH-TSP problem has a 3/2 approximation when the triangle inequality holds [20]. Therefore, improving the approximation ratio for our problem strictly below 3/2 will also improve the approximation for PATH-TSP.

The NP-hardness of MIN–SUM SALES cTSP is addressed in Appendix C. For MIN–SUM FULL-COOPERATION we have:

---

[3] As in the Euclidean case, replacing each customer with two agents is sufficient.

Theorem 3.2. Min–Sum Full-Cooperation cTSP *is APX-hard.*

Proof. We use a reduction from a variant of Set Cover, in which each element appears in exactly k sets and each set is of size d. We call this variant (k, d)-*Set-Cover*. We rely on the following theorem of [13]:

Theorem 3.3 (*[13]*)**.** *For every $\varepsilon > 0$ and a sufficiently large n, given an instance of (k, d)-Set-Cover with n elements and m sets, it is NP-hard to decide whether there exists a solution of size $\frac{m}{k-1-\varepsilon}$ or every solution is of size at least $(1-\varepsilon)m$.*

Let (C, k, d) be an instance of (k, d)-Set-Cover, where C is a collection of m subsets of a finite set S ($|S| = n$), and k and d are positive integers. We construct the following instance of our problem. Let the graph $G = (V, E)$ be constructed as follows. We have a vertex v_c for every set $c \in C$, a vertex v_s for every element $s \in S$, and two other vertices u, v. Namely,

$$V = \{u, v\} \cup V_C \cup V_S,$$

where

$$V_C = \{v_c \mid c \in C\}, \qquad V_S = \{v_s \mid s \in S\}.$$

The edge set E is defined as follows: each vertex $v_c \in V_C$ is connected to v, there is an edge between $v_s \in V_S$ and $v_c \in V_C$ iff $s \in C$, and u is connected to v. Namely,

$$E = \{(u, v)\} \cup \{(v, v_c) \mid c \in C\} \cup \{(v_c, v_s) \mid s \in c, c \in C\}.$$

The weight of each edge is 1, except for the edge (u, v), whose weight is 0. Each vertex $v_s \in V_S$ contains one agent, v contains $\lceil \frac{m}{k-1-\varepsilon} \rceil - 1$ agents, and u contains the *salesperson*. Let A be the instance of (k, d)-*Set-Cover* and let B be the instance constructed for our problem.

Claim 3.4 (*Completeness*)**.** *If there is a solution for A of size at most $\lceil \frac{m}{k-1-\varepsilon} \rceil$, then there is a solution for B of cost at most $\lceil \frac{m}{k-1-\varepsilon} \rceil + n + 1$.*

Proof. Let C' be the solution to A of size at most $\lceil \frac{m}{k-1-\varepsilon} \rceil$. The solution for B is as follows. The salesperson moves to v, and then $\lceil \frac{m}{k-1-\varepsilon} \rceil$ participants traverse from v to $V_{C'}$ where $V_{C'} = \{v_c \mid c \in C'\}$. Each of the n agents at V_S moves to a neighbor in $V_{C'}$ (since C' is a cover, every vertex in V_S has such a neighbor). Thus, the total cost of the solution for B is at most $\lceil \frac{m}{k-1-\varepsilon} \rceil + n$. □

Claim 3.5 (*Soundness*)**.** *If every solution for A is of size at least $\lfloor (1-\varepsilon)m \rfloor$ then every solution for B is of cost at least $n + \lfloor (1-\varepsilon)m \rfloor$.*

Proof. Note that there is an optimal solution in which every agent in V_S makes at least one step. This holds, since otherwise another participant has to traverse an edge adjacent to that agent, so the solution can only be cheaper if that agent from V_S moves towards the other participant.

Since each solution for A is of size at least $\lfloor (1-\varepsilon)m \rfloor$, at least $\lfloor (1-\varepsilon)m \rfloor$ of the vertices of V_C are populated after the agents in V_S make one step. Therefore, at least $\lfloor (1-\varepsilon)m \rfloor$ more steps are needed for that optimal solution. Thus, every solution to B is of cost at least $n + \lfloor (1-\varepsilon)m \rfloor$. □

By the completeness and soundness claims we obtain that it is NP-hard to approximate the problem to within $\frac{n+\lfloor (1-\varepsilon)m \rfloor}{n+\lceil \frac{m}{k-1-\varepsilon} \rceil + 1}$, which is about $\frac{d+(1-\varepsilon)k}{d+1}$ (since $m = \frac{kn}{d}$). Since $k \geq 2$, the problem is APX-hard. □

3.2. Min–Max cTSP

We first present a simple PTAS for the Purchase version of this problem, and this cannot be improved since we show that there is no FPTAS, assuming P ≠ NP (see Appendix D). For the other cooperation modes, we present constant lower bounds on the approximation ratio, assuming $P \neq NP$. We also provide algorithms that find constant-factor approximations for these problems, which are tight in one case, and are at most twice the lower bounds in the other cases. The results for the roundtrip versions resemble the results for the corresponding path versions.

3.2.1. Min–Max Purchase cTSP

We start by presenting the PTAS for the Purchase version that appears in Fig. 3 (algorithm **Coarse-Path**).

Theorem 3.6. *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 π, 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 π, V' contains a vertex at a distance of at most $\varepsilon \cdot OPT$ from w. Thus,

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

1. For each ordered subset $V' \subseteq V$ of size $1 + \lfloor 1/\varepsilon \rfloor$ or less that starts with v.
   (a) For each $u \notin V'$ that contains an agent, find its minimum distance to a 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 of the agents who come to a certain vertex before moving to the next vertex.

Fig. 3. A PTAS for Min–Max Purchase cTSP.

Hop-Visit($G(V, E), W, v$):

1. Let $G' = (V', E')$ be a weighted complete graph, where $V' \subseteq V$ is the set of vertices that contain participants, and the edge weights are the corresponding distances in G.
2. Compute a minimum spanning tree T of G'. Let its root be the salesperson's vertex v.
3. The salesperson visits an arbitrary child, and does not move any further.
4. When an agent receives a delivery:
   (a) If the agent is the only child of her parent, then she visits one of her children and stops (or does nothing if she has no children).
   (b) If the agent has a *sibling* in T who has not received the delivery yet, then the agent visits such a sibling and one of that sibling's children.
   (c) If the agent has siblings in T, and all of them have already received the delivery, then 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. 4. A 3-approximation algorithm for Min–Max Sales cTSP.

MaxDist$(V') \leq (1+\varepsilon)OPT$, and *Cost*$(V') \leq (1+\varepsilon)OPT$. Therefore, Algorithm Coarse-Path indeed finds a $(1+\varepsilon)$-approximate solution. The running time of the algorithm is $O(n^{\lfloor 1/\varepsilon \rfloor + 3})$, 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. □

We similarly have a PTAS for the roundtrip version of the problem. Simply let all of the participants return to their initial vertex at the end of Algorithm **Coarse-Path**, and compute the costs accordingly. It is easy to see that the arguments used for the path version hold here as well. Thus, we have:

Corollary 3.7. *The* roundtrip *version of* Min–Max Purchase cTSP *admits a PTAS.*

3.2.2. Min–Max Sales cTSP

In this Subsection we present a 3-approximation algorithm for both the path and roundtrip versions of the problem. These problems cannot be approximated better than factors of 2 and 3/2, respectively, unless $P = NP$ (see Appendix E for details).

The simple constant approximation algorithm for Min–Max Sales cTSP is presented in Fig. 4.

Theorem 3.8. Min–Max Sales cTSP *is* 3*-approximable.*

Proof. We prove that Algorithm **Hop-Visit** is a 3-approximation algorithm for this problem. Clearly, all of 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. □

A similar argument holds for the roundtrip version. We use algorithm **Hop-visit**, and then let each participant return to its original vertex (using the shortest path). Clearly, the Min–Max value is at most 6 times the weight of the heaviest edge of the MST of G'. On the other hand, the optimal cost is at least twice the weight of the heaviest edge of G'' (since selling to an agent requires reaching her and then returning back). Thus, we have:

Corollary 3.9. *The* roundtrip *version of* Min–Max Sales cTSP *is* 3*-approximable.*

3.2.3. MIN–MAX FULL-COOPERATION CTSP

The MIN–MAX FULL-COOPERATION CTSP problem allows only constant-factor approximations. We prove a lower bound of 2 on the approximation ratio for both the PATH and ROUNDTRIP versions of the problem. Additionally, we provide a simple algorithm that obtains a 4-approximate solution for the PATH version and a tight 2-approximate solution for the ROUNDTRIP. We start by considering the special case in which each vertex contains at least one participant.

Claim 3.10. MIN–MAX FULL-COOPERATION CTSP *is* 2-*approximable if each vertex contains at least one participant.*

Proof. Compute an MST rooted at the salesperson's vertex, and let one agent from each vertex move to her parent's vertex, and return after receiving the goods (the agents initially located at the leaves do not need to return). The maximum distance traveled by any of the agents is at most twice the weight of the heaviest edge in the MST. On the other hand, any solution to the problem forms a spanning connected subgraph, and its MIN–MAX value is at least the weight of the heaviest edge in that subgraph. As we noted before, since the MST is lexicographically minimal, its heaviest edge is not heavier than that of any other spanning connected subgraph. Hence, the algorithm is a 2-approximation algorithm. □

Claim 3.11. MIN–MAX FULL-COOPERATION CTSP *is* 4-*approximable.*

Proof. The proof is similar to the proof of the approximation for the MIN–MAX SALES version. We define the weighted complete graph $G' = (V', E')$, where V' is the set of vertices that contain participants, and the edge weights are the distances between these vertices in the original graph. We now perform the same algorithm as in the last proof: We compute an MST rooted at the salesperson's vertex, one agent from each vertex moves to the vertex of her parent, and she returns after receiving the goods. The maximum distance traveled by any of the agents is again at most twice the weight of the heaviest edge in the MST.

On the other hand, consider an optimal solution, and define $G'' = (V', E'')$, s.t. $(u, v) \in E''$ iff participants from u and v meet during that solution (the weight is again the distance between them). The optimal MIN–MAX value is clearly at least half the weight of the heaviest edge in E'', since a meeting of two participants requires that at least one of them traversed half the distance between them. Also, G'' is clearly a spanning connected subgraph of G', and its heaviest edge has at least the cost of the heaviest edge of the MST of G' found by the above algorithm. Therefore, this algorithm achieves an approximation ratio of 4. □

Note that the simple algorithm described in the last proof also solves the ROUNDTRIP version of the problem (with the same cost). On the other hand, the bound on the cost of the optimum is doubled for the ROUNDTRIP version (if two participants meet and return then one of them must travel at least the distance between them). Thus:

Corollary 3.12. *The* ROUNDTRIP *version of* MIN–MAX FULL-COOPERATION CTSP *is* 2-*approximable.*

We now turn to presenting hardness results.

Theorem 3.13. MIN–MAX FULL-COOPERATION CTSP *cannot be approximated better than a factor of* 2, *unless* $P = NP$.

Proof. We prove this by a reduction from the Set Cover problem.

The reduction. Let (C, k) be an instance of Set Cover, where C is a collection of subsets of a finite set S, and k is an integer. It is NP-hard to decide whether there is a set cover for S of size at most k, i.e., a subset $C' \subseteq C$ such that every element in S belongs to at least one member of C'.

We use the same construction as in the hardness proof for the MIN–SUM objective (Theorem 3.2), except that v contains only $k - 1$ agents.

Claim 3.14 (*Completeness*)**.** *If* (C, k) *is a "yes" instance of Set Cover, then there is a solution for our problem with maximum distance 1.*

Proof. Let $C' \subseteq C$ be a set cover of S of size $|C'| \leq k$. Let $V_{C'} = \{v_c \mid c \in C'\}$. Then the salesperson moves to v, and the k participants now populating v go to the vertices of $V_{C'}$ (at least one participant to each vertex); the agents populating the vertices V_S move to C' as well (each to a closest vertex in C'). Since C' is a set cover, each of the agents at V_S has a vertex $v_c \in V_{C'}$ at distance 1. Hence this scheme ends in one step. □

Claim 3.15 (*Soundness*)**.** *If there is a solution with maximum distance less than 2, then* (C, k) *is a "yes" instance of set cover.*

Proof. Since all edges are of length 1, each agent ends up at some vertex in V_C. Let $V_{C'}$ be the set of those vertices. Clearly $|V_{C'}| \leq k$ as there are originally only $k - 1$ agents at v and a salesperson in u, and every other agent has to meet one of them. Thus, $C' = \{c \mid v_c \in V_{C'}\}$ is a set cover for S of size at most k, hence (C, k) is a "yes" instance of Set Cover. □

Corollary 3.16 (*Hardness of Approximation*)**.** *It is NP-hard to distinguish between an instance of* MIN–MAX FULL-COOPERATION CTSP *with value 1 and an instance with value 2. Hence, this problem is NP-hard to approximate to within any factor smaller than 2.* □

The reduction for the ROUNDTRIP version is identical. By using the same considerations, a set cover of size k or less exists iff there is a solution with MIN–MAX value 2 to the new problem. Note that there can be no solution with MIN–MAX value 3, since there are no triangles in the constructed graph. We thus have:

Corollary 3.17. *The* ROUNDTRIP *version of* MIN–MAX FULL-COOPERATION CTSP *cannot be approximated better than a factor of* 2, *unless* $P = NP$.

3.3. Min–Makespan cTSP

The Min–Makespan objective is the most diverse out of the three. The Purchase problem has a polynomial time solution for both the path and the roundtrip versions. The Full-Cooperation version can be approximated within a ratio of 2, and this cannot be improved, unless $P \neq NP$. For the Sales version, only an $O(\sqrt{\log n})$ approximation is known [22], while the lower bounds for that version are smaller than 2 (5/3 for the path version, shown by [2], and 5/4 for the roundtrip version, which we show below).

3.3.1. Min–Makespan Purchase cTSP

Claim 3.18. Min–Makespan Purchase cTSP *can be solved in* $O(mn + n^2 \log n)$ *time.*

Proof. We observe that there is an optimal solution in which all of the agents meet the salesperson at a single vertex. This holds, since the value of a solution does not change if each agent who meets the salesperson joins her in her journey. Thus, they could have all met at the last vertex visited by the salesperson in that solution, without increasing the completion time. Specifically, this argument is true for the optimal solution. Hence, an optimal solution can be found simply by computing all-pairs shortest paths in the graph and finding the vertex whose maximum distance from any of the participants is minimal. This takes the above stated time using Johnson's algorithm (e.g. [10]). □

Claim 3.19. *The* roundtrip *version of* Min–Makespan Purchase cTSP *can be solved in* $O(mn + n^2 \log n)$ *time.*

Proof. The idea of this proof is similar to the idea of the previous one, but it is slightly more involved. We first show that there exists an optimal solution in which all of the agents meet the salesperson either in a single vertex or in two adjacent vertices.

Consider an optimal solution of cost OPT. Let u be the last vertex visited by the salesperson by time $OPT/2$ in that solution, and let v be the first vertex she visited after that time. Clearly, all of the agents that the salesperson met before leaving u can join her in her way to u and then return, without increasing the makespan.

We observe that all of the agents whom the salesperson met after leaving u can come to meet her at v and return to their initial vertex before time OPT. Let the meeting of such an agent with the salesperson occur at vertex w. Then clearly that agent's travel to w plus the salesperson's travel from v to w takes less than $OPT/2$ time. Thus, all of these agents can reach v before the salesperson (in less than $OPT/2$ time). They can return to their initial vertices before time OPT, since they can join the salesperson's tour until the vertex where they originally met, and then return to their initial vertex, just as in the original optimal solution.

This means that an optimal solution can be found by computing all-pairs shortest paths and enumerating single vertices and pairs of adjacent vertices where the meetings may take place. The makespan for each suggestion for meeting-place(s) is computed in $O(n)$ time (according to the distances from the participants). Again, computing all-pairs shortest paths requires an overall time of $O(mn + n^2 \log n)$ using Johnson's algorithm [10], which means that the total time required for solving the problem is $O(mn + n^2 \log n)$. □

3.3.2. Min–Makespan Sales cTSP

As noted before, the path version has an $O(\sqrt{\log n})$ approximation algorithm [22], and there is a lower bound of 5/3 on its approximation ratio (assuming $P \neq NP$) [2]. We therefore consider here only the roundtrip version.

Claim 3.20. *The* roundtrip *version of* Min–Makespan Sales cTSP *is* $O(\sqrt{\log n})$*-approximable.*

Proof. The known algorithm for the path version finds an $O(\sqrt{\log n})$ approximate solution [22]. Requiring that all of the participants return to their original vertex at the end may increase the cost of the solution found by the algorithm by a factor of at most 2. Clearly, the optimal cost for the roundtrip version is at least the optimal cost for the path version. Therefore, this problem also has an $O(\sqrt{\log n})$ approximation algorithm. □

Claim 3.21. *The* roundtrip *version of* Min–Makespan Sales cTSP *cannot be approximated better than a factor of* 5/4*, unless* $P = NP$*.*

Proof. We use a reduction from Set Cover, similar to the reduction used in Theorem 3.13 for Min–Max Full-Cooperation cTSP. We also use the same notations as in the proof of Theorem 3.13.

There are two differences in the construction of the reduction. First, we add another vertex w that contains m agents (the number of sets) and is connected to all of the vertices in V_C. Second, each of the vertices of V_C contains a number of agents equal to its degree (which is the size of the corresponding set). As in the above mentioned reduction, all of the edges have weight 1, the vertices of V_S contain one agent each, and v contains $k - 1$ agents.

A set cover of size k (or less) gives a solution of cost 4 to our problem: The salesperson and the agents in v visit the vertices in V_C corresponding to the cover. The agents in these vertices visit all of the vertices in V_S, and one of the agents who came from v visits w. Then the agents from w visit all of the non-visited vertices in V_C. It is easy to verify that all of the participants can return from these visits without exceeding a makespan of 4.

On the other hand, assume there is a solution to the new problem with a makespan of 4. It takes at least 2 time units to visit an agent in V_S or w, so clearly these agents could not visit other agents in V_S, and the same is true for agents in V_C

whom were first visited by agents from w or V_S. Hence, agents in V_S could either be visited by the salesperson, the $k-1$ agents from v, or agents in V_C whom these participants visited at the first time unit. Since the salesperson and the agents from v could visit at most k vertices of V_C in the first time unit, there is a set cover of size at most k.

Therefore, it is NP-hard to distinguish between an instance with minimum makespan of 5 and an instance with minimum makespan of 4, which yields the required result. □

3.3.3. Min–Makespan Full-Cooperation cTSP

Here we have tight upper and lower bounds of 2 on the approximation ratio. The upper bound for the path version is obtained by simply letting all of the agents go to the salesperson. Since delivering the goods to the agent who is farthest from the salesperson takes at least half the travel-duration between them, this yields a 2-approximation. The same can be done for the roundtrip version, followed by a return of all of the agents to their initial vertices. The optimum clearly requires here at least the travel-duration to the furthest agent (at least the time until she receives the goods plus the time for returning to her initial vertex if she moved). We thus have:

Corollary 3.22. Min–Makespan Full-Cooperation cTSP *is* 2*-approximable for both the* path *and* roundtrip *versions.*

The hardness proofs for both the path and roundtrip versions are identical to the corresponding Min–Max problems (see Theorem 3.13 and Corollary 3.17). Therefore, we have:

Corollary 3.23. *Both the* path *and the* roundtrip *versions of* Min–Makespan Full-Coope-ration cTSP *cannot be approximated better than a factor of* 2*, assuming P≠NP.*

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 the sense of approximation) than the classical TSP, while others are strictly harder. Several problems remain open for future work.

The status of Min–Makespan Sales cTSP is not settled, as there is an $O(\sqrt{\log n})$ approximation algorithm and a constant inapproximability factor. Moreover, for some of the Euclidean variants, the approximation factor is not better than the guarantee for the graph corresponding variants. These could most likely be improved.

It is also likely that the running time of some of the PTAS can be improved. A subsequent work of Remy, Spöhel and Weißl [31] has already improved the running time of a PTAS for Min–Sum Sales Euclidean-cTSP.

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-MST (e.g., [29,21,16,8,6]).

Finally, many other generalizations of cTSP may also be of interest. For example, different participants may have different costs for traversing an edge (recall the ice cream van vendor example: the speed of the van and the children is not the same). Other variants may require a roundtrip tour for the salesperson, but not for the agents. These remain for future research.

Acknowledgements

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

Appendix A. Proof of Lemma 2.2

Proof of Lemma 2.2. Let π be an optimal solution. For every $a, b \in \{0, 2, \ldots, L/2-2\}$ denote by π_{ab} the (a, b)-shift of π. We have to show, given a randomly chosen a and b, how to change π_{ab} to a portals-limited solution such that the expected increase in cost is at most $\varepsilon \cdot OPT$.

We refer to the axis-parallel lines of the form $x = 2k$ or $y = 2k$, where k is an integer, as *even grid lines*. Note that all portals are located on even grid lines.

Suppose that in π, a participant travels along a segment that crosses an even grid line ℓ. Let a and b be two even numbers chosen uniformly at random from $0, 2, \ldots, L/2-2$. Denote by ℓ_{ab} the (a, b)-shift of ℓ. Note that the probability (over the choices of a and b) that ℓ_{ab} contains a boundary of a level-i super-pixel is $2^i/(L/4)$. Following the choice of a and b, we replace the segment traveled by the participant by two segments, so that the crossing of ℓ_{ab} is at the closest portal on ℓ_{ab}. The corresponding increase in cost is at most the interportal distance on ℓ_{ab}, which is $(L/2^i)/m$. Thus, we may bound the expected increase in cost due to this crossing by

$$\sum_{i=1}^{\log L} \frac{L}{2^i m} \frac{2^i}{L/4} = \frac{4 \log L}{m} \le \frac{\varepsilon}{2\sqrt{2}}.$$

The last inequality holds as $m \in \left[\frac{8\sqrt{2}\log L}{\varepsilon}, \frac{16\sqrt{2}\log L}{\varepsilon}\right)$.

Now, consider a solution of π'_{ab} that is obtained by replacing each segment of the π_{ab} by two axis-parallel segments Clearly, the number of even grid lines crossings in π_{ab} is at most the number of even grid lines crossings in π'_{ab}, which is at most $\sqrt{2} \cdot OPT$.

By combining the last two arguments we obtain that the total expected increase of cost is at most $\varepsilon/2 \cdot OPT$. Thus, we showed how to obtain a solution with an expected total cost of at most $(1 + \varepsilon/2) \cdot OPT$, which satisfies condition (1) of the portals-limited solution definition.

Now we may remove self-intersections by "short-cutting". In addition, if a portal is used more than twice, we can keep "short-cutting" on the two sides of the portal until the portal is used at most twice. (If this introduces additional self-intersections, they can also be removed.) The obtained solution has an expected total cost of at most $(1 + \varepsilon/2) \cdot OPT$ and it satisfies conditions (1) and (2) of the portals-limited solution definition.

Note that changing the solution by moving each meeting point between an agent and the salesperson to the nearest pixel center increases the cost by at most $O(n) = O(OPT/n)$. Additionally, note that if in our solution two (or more) agents happen to meet, then they may cease to move. This holds, since in such a case the salesperson may come to meet the agents (and return) without increasing the total cost. Combined with the previous argument, we obtain that we can change the solution to also satisfy conditions (3) and (4) of the portals-limited solution definition, without increasing the total cost by more than $O(1/n) \cdot OPT$. The latter cost is less than $\varepsilon/2 \cdot OPT$, for a sufficiently large n.

Thus, we obtain a portals-limited solution that has an expected total cost of at most $(1 + \varepsilon) \cdot OPT$. Therefore, the proof is complete. □

Appendix B. ROUNDTRIP Versions under the MIN–SUM Objective

In this Appendix we consider the various ROUNDTRIP versions under the MIN–SUM objective, and show that each of these problems is equivalent to the classical TSP, regardless of the cooperation-mode (i.e., PURCHASE, SALES or FULL-COOPERATION). Specifically,

Claim B.1. *For any metric space $\mathcal{M}$, the* ROUNDTRIP *versions of* MIN–SUM PURCHASE cTSP *in $\mathcal{M}$,* MIN–SUM SALES cTSP *in $\mathcal{M}$, and* MIN–SUM FULL-COOPERATION cTSP *in $\mathcal{M}$ are all equivalent to* TSP *in $\mathcal{M}$.*

Proof. Consider a solution for any of the above cTSP problems in $\mathcal{M}$. In addition, consider the first meeting between the salesperson and an agent who moves in this solution. Let x be the point in which this meeting occurs, and let y be the initial location of that agent.

We observe that since each agent's moves form a cycle, there is a solution with the same cost in which that agent does not move. This holds since the salesperson can travel from x to y along the path traveled by the agent, meet the agent at x, then follow the rest of the cycle traveled by that agent (in reverse order), and return back to y. Thus, exactly the same points are visited and the cost of travel remains the same.

Therefore, w.l.o.g., in any solution of the above mentioned cTSP problems, no participant moves except the salesperson. Hence, all of the above mentioned cTSP problems in $\mathcal{M}$ are equivalent to TSP in $\mathcal{M}$. □

We thus have the following two corollaries:

Corollary B.2. *The* ROUNDTRIP *versions of* MIN–SUM PURCHASE cTSP, MIN–SUM SALES cTSP *and* MIN–SUM FULL-COOPERATION cTSP *can all be approximated within a factor of* 3/2*, and cannot be approximated within a factor of* 131/130*, unless $P \neq NP$.*

Corollary B.3. *The* ROUNDTRIP *versions of* MIN–SUM PURCHASE EUCLIDEAN-cTSP, MIN–SUM SALES EUCLIDEAN-cTSP *and* MIN–SUM FULL-COOPERATION EUCLIDEAN-cTSP *are all NP-hard but have a PTAS.*

Appendix C. NP-hardness for MIN–SUM SALES cTSP

Claim C.1. MIN–SUM SALES cTSP *is NP-hard.*

Proof. We use a reduction from PATH-TSP [20]. Recall that an instance of PATH-TSP consists of a complete weighted undirected graph, $G = (V, E)$, in which the weight function satisfies the triangle inequality, and a vertex $v \in V$, in which the salesperson is located. A solution is a Hamiltonian path that has v as one of its endpoints. The goal is to find a solution of minimum weight.

Given an instance of PATH-TSP, we construct an instance of MIN–SUM SALES cTSP as follows. For each vertex $u \neq v$, we add a vertex u', and connect it to u by an M-weighted edge, where M is twice the sum of the edge weights of G plus 1. We denote this new graph by $G' = (V', E')$. Each vertex of G' contains a participant, and the participant in v is defined to be the salesperson.

Clearly, if the optimal PATH-TSP solution is of length C, then there is a solution for the new problem with total length $(n - 1)M + C$.

On the other hand, assume there is an optimal solution for the new problem with a total cost of $(n - 1)M + C$. Clearly, the agents at new vertices do not move in such a solution (since it already costs at least $(n - 1)M$ to reach them, and if such an agent moves the cost is increased by at least M). We prove that there is an optimal solution for that problem in which agents adjacent to new vertices only move to the new vertex adjacent to them, and therefore the salesperson visits all of the vertices of G by traversing a path of length C (this path is simple since the triangle inequality holds in the original graph G).

Assume this is not true. Hence, there is an agent who travels to a vertex that is not the new vertex adjacent to it. Let the agent who started at vertex u be the first such agent whom the salesperson meets.

Clearly, some agent must visit u'. We can assume w.l.o.g. that this agent receives the goods through the agent of vertex u (not necessarily directly from her), since otherwise we can simply "switch names" between the agent of vertex u and the salesperson when they meet. It is easy to see that by switching names between agents when they meet, we can obtain a solution with the same cost, in which the agent of vertex u is the agent who returns to u and moves to u'. Therefore, the salesperson could have done the tour of that agent by herself and return to u, and the agent of vertex u could go immediately to u', without affecting the cost of the solution or the visited agents.

This argument can also be applied to each of the next agents that the salesperson meets in the given optimal solution. Therefore, there is an optimal solution in which these agents only move to new vertices, as required. Thus, there is a Hamiltonian Path in G that starts at v and has total length C, and the proof is complete. □

Appendix D. Tight NP-hardness for Min–Max Purchase cTSP

In this Appendix we show strong NP-hardness of the path and the roundtrip versions of Min–Max Purchase cTSP. This is tight, as each of these problems has a PTAS.

Claim D.1. Min–Max Purchase cTSP *has no FPTAS, unless $P = NP$.*

Proof. We show a reduction from the Hamiltonian Path problem, where a given vertex $v \in V$ must be an endpoint of the path. Given an input to that problem, $G = (V, E)$, $v \in V$, we construct an instance of our problem in the following way. For each $u \in V$, we add a vertex u' and an edge (u, u'), with a weight of $n - 1$ (the weights of the original edges remain 1). We locate the salesperson at v, and we locate an agent at each of the newly added vertices. It is easy to see that the instance of the Hamiltonian Path decision problem is a "yes" instance iff the value of the optimal solution of the new instance is $n - 1$. Thus, our problem is strongly *NP-hard* (the $n - 1$ weight used in the reduction is obviously polynomial in the input size), and therefore has no FPTAS. □

Claim D.2. *The* roundtrip *version of* Min–Max Purchase cTSP *has no FPTAS, unless $P = NP$.*

Proof. Similar to the proof of Claim D.1, we apply a reduction from Hamiltonian Cycle. Given an input $G = (V, E)$, we locate the salesperson at one of the vertices, v. Additionally, for each $u \in V$, we add a vertex u', connected by an edge (u, u') with a weight of $n/2$ (the weights of the original edges remain 1). Each of the newly added vertices contains an agent. It is easy to see that the instance of the Hamiltonian Cycle problem is a "yes" instance iff the value of the optimal solution of the new instance is n. Thus, our problem is strongly NP-hard and has no FPTAS. □

Appendix E. Hardness of approximation for Min–Max Sales cTSP

In this Appendix we show lower bounds on the approximability of both the path and the roundtrip version of Min–Max Sales cTSP.

Claim E.1. Min–Max Sales cTSP *cannot be approximated better than a factor of* 2, *unless $P \neq NP$.*

Proof. The reduction is from the Hamiltonian Path problem where one endpoint of the path, vertex u, is specified in the input.

Given an instance of that Hamiltonian Path problem, an unweighted undirected graph $G = (V, E)$ and a vertex u, we construct an instance for our problem by simply locating the salesperson at u and putting an agent at each of the other vertices.

If there is a Hamiltonian path in G that starts at u, then there is a solution for the new problem where the Min–Max value is 1, as follows. The salesperson moves to the next vertex in the path, sells to the agent there, and does not move any further. Then, each agent moves to the next vertex on the path, sells the goods to the agent there, and also does not move any further. Thus, all of the agents are visited, and the maximal distance traveled is 1.

On the other hand, if there is a solution with Min–Max value 1, then the salesperson and the agents each make at most one step and stop (i.e., they traverse at most one edge each). Since all of the agents are visited in a solution, each of the first $|V| - 1$ steps must have visited a vertex that had not been visited before. Thus, following the steps in this sequence gives a Hamiltonian path that starts at u.

It is therefore NP-hard to distinguish between an instance that has a solution with Min–Max value 1 and an instance that only has solutions with Min–Max values 2 or more. Thus, it is NP-hard to approximate the value of the optimal solution within a factor lower than 2. □

The reduction for the roundtrip version of the problem is identical. However, here a "yes" instance implies a cost of 2, and it is NP-hard to distinguish between an instance that has a solution of cost 2 and an instance that only has solutions of cost 3 or more. We thus have:

Claim E.2. *The* roundtrip *version of* Min–Max Sales cTSP *cannot be approximated better than a factor of* 3/2, *unless $P \neq NP$.*

References

[1] E. Arkin, R. Hassin, Approximation algorithms for the geometric covering salesman problem, Discrete Applied Mathematics 55 (1994) 197–218.
[2] E.M. Arkin, M.A. Bender, S.P. Fekete, J.S.B. Mitchell, M. Skutella, The freeze-tag problem: how to wake up a swarm of robots, in: Proceedings of SODA'02, 2002, pp. 568–577.
[3] E.M. Arkin, M.A. Bender, D. Ge, S. He, J.S.B. Mitchell, Improved approximation algorithms or the freeze-tag problem, in: SPAA'03, 2003, pp. 295–303.
[4] A. Armon, A. Avidor, O. Schwartz, Cooperative TSP, in: ESA'06: Proceedings of the 14th Annual European Symposium on Algorithms, Springer-Verlag, 2006, pp. 40–51.
[5] S. Arora, Polynomial-time approximation schemes for Euclidean TSP and other geometric problems, Journal of the ACM 45 (5) (1998) 753–782.
[6] S. Arora, K.L. Chang, Approximation schemes for degree-restricted MST and red–blue separation problems, Algorithmica 40 (3) (2004) 189–210.
[7] A. Bar-Noy, S. Guha, J. Naor, B. Schieber, Multicasting in heterogeneous networks, in: Proceedings of STOC'98, 1998, pp. 448–453.
[8] T.M. Chan, Euclidean bounded-degree spanning tree ratios, in: Proc. 19th ACM SoCG, 2003, pp. 11–19.
[9] 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.
[10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2001, p. 636–640.
[11] M. de Berg, J. Gudmundsson, M.J. Katz, C. Levcopoulos, M.H. Overmars, A.F. van der Stappen, TSP with neighborhoods of varying size, Journal of Algorithms 57 (2005) 22–36.
[12] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry: Algorithms and Applications, 2nd edn, Springer, Berlin, 2000, Section 4.7 "Smallest Enclosing Disks".
[13] I. Dinur, V. Guruswami, S. Khot, O. Regev, A new multilayered PCP and the hardness of hypergraph vertex cover, SIAM Journal of Computing 34 (5) (2005) 1129–1146.
[14] A. Dumitrescu, J.S.B. Mitchell, Approximation algorithms for tsp with neighborhoods in the plane, Journal of Algorithms 48 (1) (2003) 135–159.
[15] L. Engebretsen, M. Karpinski, Approximation Hardness of TSP with bounded metrics, in: Proceedings of ICALP'01, 2001, pp. 201–212.
[16] S.P. Fekete, S. Khuller, M. Klemmstein, B. Raghavachari, N. Young, A network flow technique for finding low-weight bounded-degree trees, Journal of Algorithms 24 (1997) 310–324.
[17] M.R. Garey, R.L. Graham, D.S. Johnson, Some NP-complete geometric problems, in: Proceedings of STOC'76, 1976, pp. 10–22.
[18] J. Gudmundsson, C. Levcopoulos, A fast approximation algorithm for TSP with neighborhoods, Nordic Journal of Computing 6 (4) (1999) 469–488.
[19] S.M. Hedetniemi, S.T. Hedetniemi, A.L. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (4) (1988) 319–359.
[20] J.A. Hoogeveen, Analysis of christofides' heuristic: some paths are more difficult than cycles, Operation Research Letters 10 (5) (1991) 291–295.
[21] S. Khuller, B. Raghavachari, N. Young, Low degree spanning trees of small weight, SIAM Journal of Computing 25 (2) (1996) 355–368.
[22] J. Könemann, A. Levin, A. Sinha, Approximating the degree-bounded minimum diameter spanning tree problem, Algorithmica 41 (2) (2004) 117–129.
[23] C. Mata, J.S.B. Mitchell, Approximation algorithms for geometric tour and network design problems, in: SCG'95, 1995, pp. 360–369.
[24] N. Megiddo, Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems, SIAM Journal of Computing 12 (4) (1983) 759–776.
[25] 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) (1999) 1298–1309.
[26] J.S.B. Mitchell, A ptas for tsp with neighborhoods among fat regions in the plane, in: SODA'07: Proceedings of the Eighteenth Annual ACM–SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 11–18.
[27] C.L. Monma, S. Suri, Transitions in geometric minimum spanning trees, Discrete & Computational Geometry 8 (1992) 265–293.
[28] C.H. Papadimitriou, Euclidean TSP is NP-complete, Theoretical Computer Science 4 (1977) 237–244.
[29] C.H. Papadimitriou, U.V. Vazirani, On two geometric problems related to the traveling salesman problem, Journal of Algorithms 5 (1984) 231–246.
[30] R. Ravi, Rapid rumor ramification: approximating the minimum broadcast time, in: Proceedings of FOCS'94, 1994, pp. 202–213.
[31] J. Remy, R. Spöhel, A. Weißl, On Euclidean vehicle routing with allocation, in: The 10th Workshop on Algorithms and Data Structures, WADS 2007, in: Lecture Notes in Computer Science, vol. 4619, 2007, pp. 601–612.
[32] G. Robins, A. Zelikovsky, Improved steiner tree approximation in graphs, in: Proceedings of SODA'00, 2000, pp. 770–779.
[33] S. Safra, O. Schwartz, On the complexity of approximating TSP with neighborhoods and related problems, Computational Complexity 14 (2005) 281–307.
[34] M.O. Sztainberg, E.M. Arkin, M.A. Bender, J.S.B. Mitchell, Analysis of heuristics for the freeze-tag problem, in: Proceedings of SWAT'02, 2002, pp. 270–279.
[35] E. Welzl, Smallest enclosing disks (balls and ellipsoids), in: New Results and New Trends in Computer Science, in: Lecture Notes in Computer Science, vol. 555, 1991, pp. 359–370.