TSP with Bounded Metrics

Lars Engebretsen[1,⋆] and Marek Karpinski[2,⋆⋆]

[1] Department of Numerical Analysis and Computer Science
Royal Institute of Technology
SE-100 44 Stockholm
SWEDEN
E-mail: enge@kth.se

[2] Department of Computer Science
University of Bonn
53117 Bonn
GERMANY
E-mail: marek@cs.uni-bonn.de

November 2005

Proposed running head TSP with Bounded Metrics

⋆Research partly performed while the author was visiting MIT with support from the Marcus Wallenberg Foundation and the Royal Swedish Academy of Sciences.

⋆⋆Supported in part by DFG grant, DIMACS, and IST grant 14036 (RAND-APX).

Abstract. The general asymmetric TSP with triangle inequality is known to be approximable only within logarithmic factors. In this paper we study the asymmetric and symmetric TSP problems with *bounded metrics*, i.e., metrics where the distances are integers between one and some constant upper bound. In this case, the problem is known to be approximable within a constant factor. We prove that it is **NP**-hard to approximate the asymmetric TSP with distances one and two within $321/320 - \varepsilon$ and that it is **NP**-hard to approximate the symmetric TSP with distances one and two within $741/740 - \varepsilon$ for every constant $\varepsilon > 0$.

Recently, Papadimitriou and Vempala announced improved approximation hardness results for both symmetric and asymmetric TSP with graph metric. We show that a similar construction can be used to obtain only slightly weaker approximation hardness results for TSP with triangle inequality and distances that are integers between one and eight. This shows that the Papadimitriou-Vempala construction is “local” in nature and, intuitively, indicates that it cannot be used to obtain hardness factors that grow with the size of the instance.

Key words. Approximation Hardness; Metric TSP; Bounded Metric.

Corresponding author

Marek Karpinski

Department of Computer Science

University of Bonn

53117 Bonn

GERMANY

E-mail: marek@cs.uni-bonn.de

Phone: +49 (0)228 73-4224 (office), +49 (0)228 73-4327 (secretary)

Fax: +49 (0)228 73-4440

List of symbols

δ Greek delta

ε Greek epsilon

ν Greek nu

$\lfloor$ left floor sign

$\rfloor$ right floor sign

§ section sign

1 Introduction

A common special case of the Traveling Salesman Problem (TSP) is the *metric TSP*, where the distances between the cities satisfy the triangle inequality. The decision version of this special case was shown to be **NP**-complete by Karp [13], which means that we have little hope of computing exact solutions in polynomial time. Christofides [7] has constructed an elegant algorithm approximating the metric TSP within $3/2$, i.e., an algorithm that always produces a tour whose weight is at most a factor $3/2$ from the weight of the optimal tour. For the case when the distance function may be asymmetric, the best known algorithm approximates the solution within $O(\log n)$, where n is the number of cities [12]. As for lower bounds, the PCP Theorem [1] and a result due to Papadimitriou and Yannakakis [16] together imply that there exists some constant such that it is **NP**-hard to approximate TSP where the distances are constrained to be either one or two—note that such a distance function always satisfies the triangle inequality—within that constant. This hardness result was improved by Engebretsen [9], who proved that it is, for every constant $\varepsilon > 0$, **NP**-hard to approximate TSP with distances one and two within $2805/2804 - \varepsilon$ for the asymmetric and $5381/5380 - \varepsilon$ for the symmetric, respectively, version of the problem. Böckenhauer and Seibert [5] considered the symmetric TSP with distances one, two and three, and proved a lower bound of $3813/3812 - \varepsilon$. For a discussion of bounded metric TSP, see also Trevisan [17]. It appears that the metric TSP lacks the good definability properties which seem to be needed for proving strong inapproximability results. Therefore, any new insights into explicit lower bounds here are of considerable interest.

Papadimitriou and Vempala [14] recently announced stronger approximation hardness results for the asymmetric and symmetric versions of the TSP with graph metric, but left the case of TSP with *bounded metric* open. However, their original proof contained an error influencing the explicit constants. A new proof with the new constants of $117/116 - \varepsilon$ and $220/219 - \varepsilon$, respectively, was announced by Papadimitriou and Vempala in May 2002 (the latest version of the paper is available from URL http://www-math.mit.edu/~vempala/papers/tspinapprox.ps). Apart from being an interesting question on its own, it is conceivable that the special cases with bounded metric are easier to approximate than the cases when the distance between two points can grow with the number of cities in the instance. Indeed, the asymmetric TSP with distances bounded by B can be approximated within B by just picking any tour as the solution and the asymmetric TSP with distances one and two can be approximated within $4/3$ [4]. The symmetric version of the latter problem can be approximated within $7/6$ [16]. Very recently, Berman and Karpinski [2] announced an improved algorithm, with approximation ratio $8/7$, for the latter problem.

Definition 1.1. *The* Asymmetric Traveling Salesman Problem (ATSP) *is the following minimization problem: Given a collection of cities and a matrix whose entries are interpreted as the distance from a city to another, find the shortest tour starting and ending in the same city and visiting every city exactly once.*

Definition 1.2. $(1,B)$-ATSP *is the special case of ATSP where the entries in the distance matrix obey the triangle inequality and the off-diagonal entries in the distance matrix are integers between 1 and B.* $(1,B)$-TSP *is the special case of $(1,B)$-ATSP where the distance matrix is symmetric.*

In this paper, we prove that it is, for any constant $\varepsilon > 0$, **NP**-hard to approximate (1,2)-ATSP within $321/320 - \varepsilon$ (Corollary 2.2), and that it is, for any constant $\varepsilon > 0$, **NP**-hard to approximate (1,2)-TSP within $741/740 - \varepsilon$ (Corollary 3.1). This shows that the currently best known bounds for TSP with bounded metrics are, in some sense, not that far from the best currently known bounds for general TSP with triangle inequality. Specifically, the bounds for TSP with graph metric announced by Papadimitriou and Vempala in May 2002 can be written as $1 + \varepsilon$, where $\varepsilon \approx 0.01$ for the asymmetric TSP and $\varepsilon \approx 0.005$ for the symmetric TSP. We show, on the other hand, bounds for (1,B)-(A)TSP that are of the same form but with $\varepsilon \approx 0.003$ and $\varepsilon \approx 0.0013$, respectively.

By relaxing the requirement on the "boundedness" of the metric, i.e., by allowing some larger, but still constant, B in the (1,B)-(A)TSP problem, the actual constants in the approximation hardness results for TSP with bounded metrics can be made even closer to the constants obtained by Papadimitriou and Vempala. In fact, we prove in this paper that a slight modification to the recent construction of Papadimitriou and Vempala shows that it is, for any constant $\varepsilon > 0$, **NP**-hard to approximate (1,8)-ATSP within $135/134 - \varepsilon$ (Theorem 5.1) and that it is, for any constant $\varepsilon > 0$, **NP**-hard to approximate (1,8)-TSP with $389/388 - \varepsilon$ (Theorem 6.1). In a preliminary version of this paper [10], we erroneously claimed slightly better bounds.

The proofs of our approximation hardness results follow by reduction from the problem *Hybrid* introduced by Berman and Karpinski [3]. Another way to improve our bounds is therefore to establish stronger approximation hardness results for Hybrid. Some such progress has recently been reported by Chlebíková and Chlebík [6].

2 The approximation hardness of (1,2)-ATSP

As mentioned above, we prove our hardness results by reduction from the problem *Hybrid*, introduced by Berman and Karpinski [3] to prove hardness results for special cases of several combinatorial optimization problems where the number of occurrences of every variable is bounded by some constant. Essentially, Hybrid is the problem of maximizing, given a system of linear equations with special structure, the number of satisfied equations. The special structure of the linear equations in Hybrid is particularly well-suited for our reduction: The equations have either two or three unknowns and each variable occurs exactly three times in the instance.

The main idea in the reduction is the same as in earlier reductions [9, 16]; the reduction is *local* and *gadget based*. Specifically, each equation in the Hybrid instance is transformed into a certain subgraph of the TSP instance—a so called *gadget*. Different parts of the gadget correspond to the different variables participating in the equation. The gadgets are then linked together to form a circle. By the construction of the gadgets, there is a natural way to interpret a TSP tour in the resulting graph as an assignment to the variables in the Hybrid instance. To ensure that there is a certain connection between the length of the TSP tour and the number of equations satisfied by the corresponding assignment, the parts of the instance corresponding to the same variable are connected to each other in a certain way.

To obtain a good approximation hardness result, the gadgets must, loosely speaking, contain as few nodes as possible. On the other hand, the major challenge in the proof of correctness is to prove that *every* TSP tour in the resulting graph can be interpreted as an assignment to the variables in the Hybrid instance with the property that the number of satisfied equations is connected to the cost of the tour. Such connections are usually easier to establish when the gadgets contain more nodes. In this work, we are able to improve the approximation hardness constants by, firstly, observing that the Hybrid instances actually have even more structure than is explicitly stated by Berman and Karpinski [3] and, secondly, using gadgets with few nodes. This requires a fairly involved argument to establish that our reduction is correct.

2.1 The Hybrid problem and its connection to TSP

In their paper on approximation hardness of bounded occurrence instances of several combinatorial optimization problems, Berman and Karpinski [3] introduced the problem *Hybrid* and proved that it is hard to approximate.

Definition 2.1. Hybrid *is the following maximization problem: Given a system of linear equations mod* 2 *containing* n *variables,* m_2 *equations with exactly two unknowns, and* m_3 *equations with exactly three unknowns, find an assignment to the variables that satisfies as many equations as possible.*

Theorem 2.1 [3]. *For any constant* $\delta > 0$*, there exists instances of Hybrid with* 42ν *variables,* 60ν *equations with exactly two variables, and* 2ν *equations with exactly three variables such that:*

1. *Each variable occurs exactly three times.*
2. *Either there is an assignment to the variables that leaves at most* $\delta\nu$ *equations unsatisfied, or else every assignment to the variables leaves at least* $(1-\delta)\nu$ *equations unsatisfied.*
3. *It is **NP**-hard to decide which of the two cases in item 2 above holds.*

Delving into the details of the Berman-Karpinski construction, it can be seen that every instance of Hybrid produced by it has an even more special structure: The equations containing three unknowns are of the form $x+y+z = \{0,1\}$; the number of such equations with right-hand side 0 is equal to the number of such equations with right-hand side 1. The equations containing two unknowns are all of the form $x_i + x_j = 0$. Moreover, the set of variables can be partitioned into classes with the property that for each class $\{x_1, x_2, \dots, x_k\}$ of variables there are equations $x_i + x_{i+1} = 0$ $(1 \leq i < k)$ and one equation $x_k + x_1 = 0$.

By rewriting the latter equations mentioned above as $x_i + \bar{x}_{i+1} = 1$ $(1 \leq i < k)$ and $x_k + \bar{x}_1 = 1$, we have established the following corollary of Theorem 2.1:

Corollary 2.1. *There are instances of Hybrid with* 42ν *variables,* 42ν *equations of the form* $x + \bar{y} = 1 \bmod 2$*,* 18ν *equations of the form* $x + y = 0 \bmod 2$*,* ν *equations of the form* $x + y + z = 0 \bmod 2$*, and* ν *equations of the form* $x + y + z = 1 \bmod 2$ *such that:*

1. *Each variable occurs exactly three times, two times positively and one time negatively.*

2. *Either there is an assignment to the variables that leaves at most* $\delta\nu$ *equations unsatisfied, or else every assignment to the variables leaves at least* $(1 - \delta)\nu$ *equations unsatisfied.*

3. *It is* ***NP****-hard to decide which of the two cases in item 2 above holds.*

To prove our hardness result for (1,2)-ATSP, we reduce instances of Hybrid having the form described in Corollary 2.1 to instances of (1,2)-ATSP:

Theorem 2.2. *Suppose that we are given an arbitrary instance of Hybrid with* n *variables,* $m_{2,0}$ *equations of the form* $x + y = 0 \bmod 2$*,* $m_{2,1}$ *equations of the form* $x + \bar{y} = 1 \bmod 2$*,* $m_{3,0}$ *equations of the form* $x + y + z = 0 \bmod 2$*, and* $m_{3,1}$ *equations of the form* $x + y + z = 1 \bmod 2$ *such that each variable occurs exactly three times, two times positively and one time negatively.*

Then it is possible to construct in polynomial time an instance of (1,2)-ATSP, with size polynomial in the size of the Hybrid instance, such that

1. *If there is an assignment to the variables in the Hybrid instance that leaves at most* u *equations unsatisfied, then there is a TSP tour of length* $6n + m_{2,0} + m_{2,1} + 4m_{3,0} + 4m_{3,1} + u$*.*

2. *From any TSP tour of length* $6n + m_{2,0} + m_{2,1} + 4m_{3,0} + 4m_{3,1} + u$*, it is possible to construct in polynomial time an assignment to the variables in the Hybrid instance that leaves at most u equations unsatisfied.*

The rest of this section is devoted to the proof of Theorem 2.2. Before turning to that, however, let us use the theorem to establish approximation hardness of (1,2)-ATSP:

Corollary 2.2. *For any constant* $\varepsilon > 0$*, it is* ***NP****-hard to approximate (1,2)-ATSP within* $321/320 - \varepsilon$.

Proof. Select $\delta > 0$ such that $(321 - \delta)/(320 + \delta) \geq 321/320 - \varepsilon$. From an instance of Hybrid with the structure described in Corollary 2.1, construct an instance of (1,2)-ATSP with the properties guaranteed by Theorem 2.2. Combining Theorem 2.2 with item 2 in Corollary 2.1 shows that the constructed (1,2)-ATSP instance either has a tour of length at most $6{\cdot}42\nu{+}42\nu{+}18\nu{+}4\nu{+}4\nu{+}\delta\nu = (320{+}\delta)\nu$ or that every TSP tour has length at least $6 \cdot 42\nu + 42\nu + 18\nu + 4\nu + 4\nu + (1 - \delta)\nu = (321 - \delta)\nu$. Furthermore, item 3 in Corollary 2.1 states that it is **NP**-hard to distinguish those two cases. Therefore it is **NP**-hard to approximate (1,2)-ATSP within $(321 - \delta)/(320 + \delta) \geq 321/320 - \varepsilon$. ∎

2.2 Main ideas in the proof of Theorem 2.2

To describe a (1,2)-(A)TSP instance, it is enough to specify the edges of weight one. We do this by constructing a graph G and then let the (1,2)-(A)TSP instance have the nodes of G as cities. The distance between two cities u and v is defined to be one if (u, v) is an edge in G and two otherwise. To compute the weight of a tour, it is enough to study the parts of the tour traversing edges of G. In the asymmetric case G is a directed graph.

Definition 2.2. *We call a node where the tour leaves or enters* **G** *an* endpoint. *A node with the property that the tour both enters and leaves* **G** *in that particular node is called a* double endpoint *and counts as two endpoints.*

If c is the number of cities and $2e$ is the total number of endpoints, the weight of the tour is $c + e$ since every edge of weight two corresponds to two endpoints. Conversely, any tour of weight $c + e$ has exactly $2e$ endpoints.

On a high level, the (1,2)-ATSP instance in our reduction consists of a circle formed by *equation gadgets* representing the equations occurring in the corresponding instance of Hybrid. These equation gadgets are also connected through *consistency checkers*. We first show that any assignment satisfying all but u equations in the Hybrid instance can be transformed into a tour with exactly $2u$ endpoints. We then show that *any* TSP tour can be transformed by local transformations into another tour with equal or lower cost, and that it is possible to extract an assignment to the variables in the Hybrid instance from the way that this new tour traverses certain parts of TSP instance. This assignment satisfies all but at most $\lfloor e/2 \rfloor$ equations in the Hybrid instance, where e is the number of endpoints in the tour.

The proof of Theorem 2.2 now proceeds by first defining the gadgets and the consistency checkers, then defining the local transformations of an arbitrary TSP tour, and finally describing how an assignment can be found from the resulting tour.

2.3 Constructing a (1,2)-ATSP instance from Hybrid

The equation gadgets for equations of the form $x + y + z = \{0, 1\}$ are shown in Fig. 2; gadgets for equations of the form $x + y = 0$ and $x + \bar{y} = 1$ are shown in Fig. 3. The gadget we use for equations of the form $x + y + z = 0$ is very similar to the gadget used by Papdimitrou and Vempala [15]. The ticked edges in the gadgets correspond to the variables in the corresponding equation as indicated in the figures. The following properties of the gadgets can be checked by exhausting all possibilities:

Proposition 2.1. *There is a Hamiltonian path from A to B in the left gadget in Fig. 2 if and only if an even number of ticked edges is traversed and a Hamiltonian path from A to B in the right gadget in Fig. 2 if and only if an odd number of the ticked edges is traversed.*

There is a Hamiltonian path from A to B in the left gadget in Fig. 3 if and only if an even number of the ticked edges is traversed. There is a Hamiltonian path from A to B in the right gadget in Fig. 3 if and only if an odd number of the ticked edges is traversed.

The ticked edges corresponding to the same variable are joined together in a consistency checker. Specifically, the ticked edges are syntactic sugar for parts of the corresponding consistency checker. An entire consistency checker is shown in Fig. 4. A ticked edge in the equation gadgets shown in Fig. 2 corresponds to one of the three structures enclosed by a curve in Fig. 4. The correspondence is such that negated variables always correspond to the part enclosed by a dashed curve in Fig. 4—recall that each variable occurs one times negated and two times unnegated.

Note that there is no node between the two ticked edges in the gadget corresponding to equations of the form $x + y = 0$. Instead, the edge leaving the consistency checker corresponding to the first ticked edge is merged with the edge entering the consistency checker corresponding to the second ticked edge as shown in Fig. 5. This simplifies, and improves, our accounting procedure used to compute the actual approximation hardness constant.

The equation gadgets are hooked together in a circle in such a way that the node B in each gadget is identified with the node A in another gadget. The order of the gadgets is as follows: first all gadgets for equations of the form $x + y + z = 1$, then the gadgets for equations of the form $x + y + z = 0$, and finally the gadgets for equations containing two variables.

The connection between two gadgets corresponding to equations of the form $x+y+z = 1$ is "optimized" as indicated in Fig. 6. To the left, this figure shows the edges incident to B in one gadget and the edges leaving A in the other gadget; the bipartite graph on the right shows how this connection is actually implemented in our construction. This optimization improves the inapproximability factor slightly since the total number of nodes in the graph is reduced. Also the connection between the last gadget corresponding to an equation of the form $x + y + z = 1$ and the first gadget corresponding to an equation of the form $x+y+z = 0$ is optimized similarly. There is one node at A in the first gadget corresponding to an equation of the form $x + y + z = 1$; this node is shared with one gadget corresponding to an equation containing two variables.

Lemma 2.1. *A graph constructed as described above from an instance of Hybrid with n variables, $m_{2,0}$ equations of the form $x + y = 0 \bmod 2$, $m_{2,1}$ equations of the form $x + \bar{y} = 1 \bmod 2$, $m_{3,0}$ equations of the form $x + y + z = 0 \bmod 2$, and $m_{3,1}$ equations of the form $x + y + z = 1 \bmod 2$ has in total $6n + m_{2,0} + m_{2,1} + 4m_{3,0} + 4m_{3,1}$ nodes.*

Proof. There is one consistency checker for every variable; each one of them contains six nodes. Not counting the nodes inside the consistency checkers, the gadgets for equations with two variables contain two nodes; both those nodes are shared between two gadgets. Hence each gadget corresponding to a two-variable equation contains, on average, one node.

Gadgets for equations of the form $x + y + z = 1$ as shown in Fig. 2 contain four nodes—except for the "leftmost" one which contains one extra node that is shared with another gadget. Similarly, gadgets for equations of the form $x + y +$

$z = 0$ contain five nodes, two of which are shared between two gadgets—again except for the "leftmost" gadget which contains four nodes, one of which is shared with another gadget. Hence each gadget corresponding to a three-variable equation contains, on average, four nodes. ∎

2.4 Constructing a tour from an assignment

Consider an instance of Hybrid and an instance of (1,2)-ATSP constructed from it as described in § 2.3. Let π be an assignment to the variables in the Hybrid instance. We now describe a TSP tour corresponding to this assignment.

Consider the tour that 1) For each variable x traverses the consistency checker corresponding to x as shown in Fig. 7a if $\pi(x) = 0$ and as shown in Fig. 7b if $\pi(x) = 1$. 2) For each equation gadget enters each equation gadget at A, takes the shortest possible way to B under the condition that the ticked edges are traversed as prescribed by the traversals of the consistency checkers described above, and then exits the equation gadget at B.

Such a tour has precisely two endpoints in each equation gadget corresponding to an unsatisfied equation and no endpoints elsewhere. (A slight technicality arises here, however, since the three ticked edges in a gadget corresponding to equations of the form $x + y + z = 0$ cannot be simultaneously traversed—that would result in a short cycle. Similarly, both edges in gadgets corresponding to equations of the form $x + \bar{y} = 1$ cannot be simultaneously traversed. We resolve these issues by defining the tour as shown in Figs. 8 and 9, thereby maintaining the property that the tour has two endpoints for each unsatisfied equation and no other endpoints.) The properties of the above construction can be summarized as follows:

Proposition 2.2. *Consider an instance of Hybrid and an instance of (1,2)-ATSP constructed from it as described in § 2.3. Let π be an assignment to the variables* *in the Hybrid instance that satisfies all but u equations. Then the tour constructed as described above has exactly $2u$ endpoints.*

2.5 Constructing an assignment from a tour

To construct an assignment from a given TSP tour, we consider how the tour behaves on the edges of the graph defining the TSP instance. The main idea in the construction is that if the tour traverses a consistency gadget as shown in Fig. 7a the corresponding variable should be given the value 0, and if the consistency gadget is traversed as shown in Fig. 7b the corresponding variable should be given the value 1. Complications arise, of course, from the fact that an arbitrary TSP tour may enter, or leave, a consistency checker somewhere in the middle. Such traversals cannot immediately be interpreted as an assignment to the corresponding variable.

Considering the equation gadgets, the ticked "edges" in Figs. 2 and 3 are not really edges since they correspond to parts of the corresponding consistency checker. Hence a TSP tour may leave or enter a ticked "edge" in the middle—we call such edges *semitraversed*. With slight abuse of notation, we also say that an occurrence of a literal is *traversed* if both of its connecting edges in the corresponding consistency checker are traversed, *untraversed* if none of its connecting edges are traversed, and *semitraversed* otherwise.

We resolve the problem of semitraversed occurrences by performing a sequence of local transformations of the given tour. These transformations convert an arbitrary TSP tour into a TSP tour with equal or lower cost that does not contain any semitraversed occurrences. From this resulting tour, an assignment can be constructed and it can be shown that every equation that is unsatisfied under this assignment can be associated with two unique endpoints in the TSP tour.

2.5.1 Obtaining structure inside consistency checkers

In the first phase, we first make all *bridges*, i.e., all pairs of undirected edges in the consistency checkers, traversed. Knowing that all bridges are traversed by the tour then makes it possible to prove results about further transformations of the tour.

Lemma 2.2. *Consider an instance of Hybrid and an instance of (1,2)-ATSP constructed from it as described in § 2.3. In such an instance, any TSP tour can be modified into a TSP tour that traverses both bridges in every consistency checker. Moreover, this transformation can be done in polynomial time and it does not increase the length of the tour.*

Proof. For every bridge, it can be seen by considering all possibilities exhaustively that any TSP tour that traverses some set E of the four connection edges can be modified into a tour with fewer endpoints that traverses the bridge and a subset of the edges in E. The less obvious cases are shown in Fig. 10. ∎

Lemma 2.3. *Consider an instance of Hybrid and an instance of (1,2)-ATSP constructed from it as described in § 2.3. In such an instance, any TSP tour that traverses both bridges in every consistency checker can be modified into a TSP tour where the consistency checkers are traversed as shown in Figs. 7, 11, 12a–d, and 13. Moreover, this transformation can be done in polynomial time and it does not increase the length of the tour.*

Proof. The assumption that the TSP tour traverses both bridges in every consistency checker implies that the consistency checkers are traversed as shown in Figs. 7, 11, 12, and 13. Without increasing the number of endpoints in the tour, we can replace the traversals shown in Figs. 12e, g and i with the traversal shown in Fig. 12a; and the ones shown in Figs. 12f, h and j with the one shown in Fig. 12b. ∎

2.5.2 Removing semitraversals

The transformations described in this section have the purpose of removing all semitraversals from the TSP tour. This is performed by a two-step procedure. First, we take care of variables x for which the negative occurrence of x is semitraversed. After this procedure, the only possible remaining semitraversals are on positive occurrences of variables. An exhaustive case analysis then shows that it is possible to get rid of also those semitraversals without increasing the total number of endpoints in the graph.

Lemma 2.4. *Consider an instance of Hybrid and an instance of (1,2)-ATSP constructed from it as described in § 2.3. In such an instance, any TSP tour that traverses the consistency checkers as shown in Figs. 7, 11, 12a–d, and 13 can be transformed into a tour where the consistency checkers are traversed as shown in Figs. 7, 11, and 13. Moreover, this transformation can be done in polynomial time and it does not increase the length of the tour.*

Proof. We need to prove that we can get rid of traversals shown in Figs. 12a–d. To this end, consider an arbitrary consistency checker traversed as shown in Fig. 12c. The corresponding variable y occurs negatively in some equation $x + \bar{y} = 1$. We claim that it is possible to modify the tour, without increasing the total number of endpoints, in such a way that the considered consistency checker is traversed either as in Fig. 7a or as in Fig. 11a. In particular, first suppose that the gadget is traversed as shown in Fig. 14a, i.e., that none of the two edges leading to node B is traversed. Then we can remove two endpoints inside the gadget by traversing the consistency checker for $\bar{y}$ as shown in Fig. 7a. This may introduce at most two endpoints elsewhere, so the net effect is that the total number of endpoints is not increased. Secondly, if there is one traversed edge leading to node B, the equation gadget must be traversed as shown in Fig. 14b. We can then change the traversal inside the gadget so that the upper edge leaving node A in Fig. 14b is traversed instead of the lower edge. This does not change the total number of endpoints in the graph and it makes the consistency checker for $\bar{y}$ traversed as in Fig. 11a.

The procedure described above can also be used to change traversals shown in Fig. 12a into traversals shown in Figs. 7a and 11c. A very similar procedure changes traversals shown in Fig. 12d into traversals shown in Figs. 7a and 11b. and traversals shown in Fig. 12b into traversals shown in Figs. 7a and 11c. ∎

Lemma 2.5. *Consider an instance of Hybrid and an instance of (1,2)-ATSP constructed from it as described in § 2.3. In such an instance, any TSP tour that traverses the consistency checkers as shown in Figs. 7, 11, and 13 can be transformed into a tour where the consistency checkers are traversed as shown in Figs. 7 and 11. Moreover, this transformation can be done in polynomial time and it does not increase the length of the tour.*

Proof. First note that each semitraversed occurrence contains one endpoint. By making a semitraversed occurrence traversed, one endpoint is therefore removed from the consistency checker.

Since only positive occurrences of variables can be semitraversed according to the assumptions in the lemma, the only possibility to consider for gadgets corresponding to equations of the form $x + \bar{y} = 1$ is that x is semitraversed and $\bar{y}$ untraversed. In that case, however, we can remove two endpoints from the tour by making x traversed.

Gadgets corresponding to equations of the form $x+y = 0$ can contain either two semitraversed ticked edges or one semitraversed and one untraversed ticked edge since the two ticked edges are connected as shown in Fig. 5. In the former case, we make both semitraversed edges traversed and the edge from A to B untraversed by the tour; in the latter case we make the semitraversed edge untraversed and let the tour traverse the edge from A to B. It is easy to see that the resulting tours do not have more endpoints than the original tours.

In gadgets corresponding to equations of the form $x + y + z = 0$, the tour is modified as follows: If there are three semitraversed occurrences we modify the tour so that the gadget is traversed according to Fig. 8—since every semitraversed occurrence contains one endpoint that removes at least one endpoint. If there are two semitraversed occurrences and one traversed we again modify the tour so that the gadget is traversed according to Fig. 8—that does not increase the number of endpoints. If there are two semitraversed occurrences and one untraversed we make both semitraversed occurrences traversed and modify the tour on the equation gadget so that there is a Hamiltonian path from A to B—that removes two endpoints. For the remaining case, one semitraversed edge, an exhaustive case analysis shows that by changing the traversal of the equation gadget in such a way that there is an even number of traversed edges and a Hamiltonian path from A to B, the total number of endpoints is not increased.

In gadgets corresponding to equations of the form $x + y + z = 1$, we make all semitraversed occurrences traversed and then adjust the tour on the rest of the gadget in such a way that the total number of endpoints is minimized. If there are initially three semitraversed occurrences we remove at least two endpoints. If there are initially two semitraversed occurrences and one traversed, we remove two endpoints. If there are initially two semitraversed occurrences and one untraversed, we keep the number of endpoints constant. If there is initially one semitraversed occurrence and either two traversed or two untraversed, we remove two endpoints. Finally, if there is initially one semitraversed, one traversed and one untraversed occurrence, we keep the number of endpoints constant. ∎

2.5.3 Defining the assignment

By the local transformations described in the previous two subsections, we can assume that the consistency checkers are traversed as shown in Figs. 7 and 11, i.e., there are no semitraversed occurrences. Turning to the equation gadgets, this means that each ticked edge is either traversed or untraversed; there are no semitraversed ticked edges. If we look at each equation locally, and assume that the variables participating in the equation are given assignments according to how the corresponding ticked edge is traversed—0 for untraversed edges; 1 for traversed edges—Proposition 2.1 states that there will be at least two endpoints in equation gadgets corresponding to unsatisfied equations. Hence, if all consistency checkers were traversed as shown in Fig. 7, we could assign values to variables according to the traversal of the consistency checkers and directly attribute two endpoints to every unsatisfied equation.

However, some consistency checkers may be traversed as shown in Fig. 11. Suppose that the consistency checker corresponding to some variable x is traversed as shown in Fig. 11a and suppose that we assign the value 1 to x. In the equation where x occurs negated and in one of the two equations where x occurs positively, the corresponding ticked edges then "announce" the correct value for x. In the remaining equation, though, the ticked edge corresponding to the second positive occurrence of x looks untraversed although x has been assigned the value 1. Since the ticked edges announces that $x = 0$ although in fact $x = 1$, the number of endpoints in this equation gadget could be zero even though the equation will not be satisfied by the assignment. But there are in this case two endpoints in the consistency checker for x; these two endpoints correspond precisely to the occurrence for which the consistency checker announces the wrong assignment. Announcing a wrong assignment in the worst case makes an equation gadget "think" that an equation is satisfied although it is not, but then the two endpoints that come with this erroneous announcement can pay for this unsatisfied equation.

Lemma 2.6. *Consider an instance of Hybrid and an instance of (1,2)-ATSP constructed from it as described in § 2.3. From any TSP tour with e endpoints that traverses the consistency checkers as shown in Figs. 7 and 11 it is possible to construct an assignment to the variables in the Hybrid instance with the property that at most $\lfloor e/2 \rfloor$ equations are left unsatisfied.*

Proof. The assignment is constructed as follows: Variables whose consistency checker is traversed as shown in Figs. 7a and 11c–d are given the value 0; all other variables are given the value 1.

Consider an arbitrary equation gadget. Since all consistency checkers are traversed as shown in Figs. 7 and 11, there are no semitraversed ticked edges. Under the assumption that each variable in the considered equation is given an assignment according to the traversal of the corresponding ticked edge in the considered equation gadget—the value 0 if the ticked edge is untraversed and the value 1 otherwise—there will be at least two endpoints in the gadget if the assignment does not satisfy the equation.

Consider now an arbitrary consistency checker. If it is traversed as shown in Fig. 11, there is one equation where the ticked edge is not traversed according to the assignment defined in the first paragraph of this proof. Hence it may happen that there is no endpoint in the corresponding equation gadget although the equation is in fact not satisfied under the assignment defined above. However, each consistency checker traversed as shown in in Fig. 11 contains at least two endpoints. To sum up, there is at least two distinct endpoints for each unsatisfied equation if the assignment is defined as in the first paragraph of this proof. ∎

2.6 Proof of Theorem 2.2

Given an instance of Hybrid with the properties described in Theorem 2.2, an instance of (1,2)-ATSP is constructed as described in § 2.3. By Lemma 2.1, this instance has in total $6n + m_{2,0} + m_{2,1} + 4m_{3,0} + 4m_{3,1}$ cities.

If there is an assignment to the variables in the Hybrid instance that leaves at most u equations unsatisfied, it follows from Proposition 2.2 that the tour constructed from this assignment as described in § 2.4 has length $6n + m_{2,0} + m_{2,1} + 4m_{3,0} + 4m_{3,1} + u$.

Conversely, given a TSP tour of length $6n + m_{2,0} + m_{2,1} + 4m_{3,0} + 4m_{3,1} + u$, Lemmas 2.2–2.6 show that we can construct in polynomial time an assignment to the variables in the Hybrid instance that leaves at most u equations unsatisfied by first applying the transformations described in §§ 2.5.1 and 2.5.2 and then defining the assignment as described in § 2.5.3.

3 The hardness of (1,2)-TSP

It is possible to adapt the above construction for (1,2)-ATSP to prove a lower bound also for (1,2)-TSP, yielding the following result:

Theorem 3.1. *Suppose that we are given an arbitrary instance of Hybrid with n variables, $m_{2,0}$ equations of the form $x + y = 0 \bmod 2$, $m_{2,1}$ equations of the form $x + \bar{y} = 1 \bmod 2$, $m_{3,0}$ equations of the form $x + y + z = 0 \bmod 2$, and $m_{3,1}$ equations of the form $x + y + z = 1 \bmod 2$ such that each variable occurs exactly three times, two times positively and one time negatively.*

Then it is possible to construct in polynomial time an instance of (1,2)-TSP, with size polynomial in the size of the Hybrid instance, such that

1. *If there is an assignment to the variables in the Hybrid instance that leaves at most u equations unsatisfied, then there is a TSP tour of length $16n + m_{2,0} +$*

$m_{2,1} + 3m_{3,0} + 5m_{3,1} + u$.

2. *From any TSP tour of length* $16n + m_{2,0} + m_{2,1} + 3m_{3,0} + 5m_{3,1} + u$*, it is possible to construct in polynomial time an assignment to the variables in the Hybrid instance that leaves at most u equations unsatisfied.*

Corollary 3.1. *For any constant* $\varepsilon > 0$*, it is* ***NP****-hard to approximate (1,2)-TSP within* $741/740 - \varepsilon$.

Proof. Select $\delta > 0$ such that $(741 - \delta)/(740 + \delta) \geq 741/740 - \varepsilon$. From an instance of Hybrid with the structure described in Corollary 2.1, construct an instance of (1,2)-TSP with the properties guaranteed by Theorem 3.1. Combining Theorem 3.1 with item 2 in Corollary 2.1 shows that the constructed (1,2)-ATSP instance either has a tour of length at most $16 \cdot 42\nu + 42\nu + 18\nu + 3\nu + 5\nu + \delta\nu = (740 + \delta)\nu$ or that every TSP tour has length at least $16 \cdot 42\nu + 42\nu + 18\nu + 3\nu + 5\nu + (1 - \delta)\nu = (741 - \delta)\nu$. Furthermore, item 3 in Corollary 2.1 states that it is **NP**-hard to distinguish those two cases. Therefore it is **NP**-hard to approximate (1,2)-TSP within $(741 - \delta)/(740 + \delta) \geq 741/740 - \varepsilon$. ∎

The details of the construction leading to Theorem 3.1, as well as the proof of correctness, is very similar to the construction for the asymmetric case. Therefore, we describe most of the construction on a high level, delving into details only where the argument differs from the asymmetric case.

3.1 Constructing a (1,2)-TSP instance from Hybrid

Given an instance Hybrid with n variables, $m_{2,0}$ equations of the form $x + y = 0 \bmod 2$, $m_{2,1}$ equations of the form $x + \bar{y} = 1 \bmod 2$, $m_{3,0}$ equations of the form $x + y + z = 0 \bmod 2$, and $m_{3,1}$ equations of the form $x + y + z = 1 \bmod 2$, the corresponding instance of (1,2)-TSP is constructed as described below:

The equation gadgets for equations of the form $x + y + z = \{0, 1\}$ are shown in Fig. 15; gadgets for equations of the form $x + y = 0$ and $x + \bar{y} = 1$ are shown in Fig. 16. The ticked edges in the gadgets correspond to the variables in the corresponding equation as indicated in the figures.

The ticked edges corresponding to the same variable are joined together in a consistency checker as shown in Fig. 17. The correspondence is such that negated variables always correspond to the part enclosed by a dashed curve in Fig. 17—recall that each variable occurs one times negated and two times unnegated.

As in the asymmetric case, there is no node between the two ticked edges in the gadget corresponding to equations of the form $x + y = 0$. Instead, the edge leaving the consistency checker corresponding to the first ticked edge is merged with the edge entering the consistency checker corresponding to the second ticked edge as shown in Fig. 18. Similarly, there is no node in the center of the gadget for equations of the form $x + y + z = 0$. Instead, the consistency checkers are joined as shown in Fig. 19.

The equation gadgets are hooked together in a circle in such a way that node B in each gadget is identified with node A in another gadget. With an argument similar to the proof of Lemma 2.1, it can be seen that the instance produced as described above has $16n + m_{2,0} + m_{2,1} + 3m_{3,0} + 5m_{3,1}$ cities.

3.2 Constructing a tour from an assignment

Consider an instance of Hybrid and an instance of (1,2)-TSP constructed from it as described in § 3.1. Let π be an assignment to the variables in the Hybrid instance. We now describe a TSP tour corresponding to this assignment.

Consider the tour that 1) For each variable x traverses the consistency checker corresponding to x as shown in Fig. 20a if $\pi(x) = 0$ and as shown in Fig. 20b if $\pi(x) = 1$. 2) For each equation gadget enters each equation gadget at node A, takes the shortest possible way to B under the condition that the ticked edges are traversed as prescribed by the traversals of the consistency checkers described above, and then exits the equation gadget at node B.

It can be seen by case analysis that such a tour has precisely two endpoints in each equation gadget corresponding to an unsatisfied equation and no endpoints elsewhere. (As in the asymmetric case, slight technicalities arise here since the three ticked edges in a gadget corresponding to equations of the form $x + y + z = 0$ cannot be simultaneously traversed, nor can the two ticked edges in gadgets corresponding to equations of the form $x + \bar{y} = 1$. These technicalities are resolved in the same way as in the asymmetric case.)

3.3 Constructing an assignment from a tour

As in the asymmetric case, it remains to show that *any* TSP tour with e endpoints in a (1,2)-TSP instance constructed from a Hybrid instance as described in § 3.1 can be associated with an assignment to the variables in the Hybrid instance and that this assignment satisfies all but at most $\lfloor e/2 \rfloor$ equations.

The proof of this fact follows in exactly the same way as in the asymmetric case. The only additional complication follows from that fact that some consistency checkers have two connection edges on one side due to the gadgets corresponding to equations of the form $x + y + z = 0$ (Fig. 19). However, any tour that traverses two connection edges on some consistency checker can be transformed into a tour without this property by a simple local transformation as indicated in Fig. 22. Having established this, it can be seen by a case analysis that any tour can be transformed into a tour that traverses all bridges and does not have more endpoints than the original tour in precisely the same way as indicated in the proof of Lemma 2.2 and Fig. 20. The remaining transformations described in §§ 2.5.1 and 2.5.2 can be straightforwardly adapted to the symmetric case since they only work with the connection edges of the consistency checkers. Having transformed the tour, the assignment to the variables in the Hybrid instance is defined as follows: Variables whose consistency checker is traversed as shown in Figs. 20a and 21c–d are given the value 0; all other variables are given the value 1. It can then be seen in the same way as in § 2.5.3 that this assignment has the properties required by Theorem 3.1.

4 Trading "boundedness" for approximation hardness

Papadimitriou and Vempala [15] prove their hardness result for TSP with graph metric by reduction from Håstad's approximation hardness result for systems of linear equations [11].

Theorem 4.1 [11]. *For any constant* $\delta \in (0, 1/2)$*, there exists systems of linear equations mod 2 with* $2m$ *equations and exactly three unknowns in each equation such that: 1) Each variable in the instance occurs a constant number of times, half of them negated and half of them unnegated. This constant grows as* $\Omega(2^{1/\delta})$*. 2) Either there is an assignment satisfying all but at most* δm *equations, or every assignment leaves at least* $(1 - \delta)m$ *equations unsatisfied. 3) It is* ***NP****-hard to distinguish between these two cases.*

From a system of linear equations with the properties described in Theorem 4.1, Papadimitriou and Vempala construct an instance of ATSP by hooking several *gadgets* together. Each equation is represented by an *equation gadget* of the form shown in Fig. 24. The ticked edges in that figure in fact correspond to gadgets themselves; these gadgets are shown in Fig. 25. The construction is parameterised; Papadimitriou and Vempala set $a = 4$, $b = 2$ and $d = 6$ in the current version of their paper [15]. The main idea in the construction is that the way a TSP tour traverses the latter gadgets mentioned above, the so called *edge gadgets*, gives an assignment to the variables in the underlying system of linear equations (see Figs. 26 and 27). The main technical challenge is to prove that there is a correspondence between the length of TSP tours in the constructed graph and the number of equations satisfied by the corresponding assignment. To this end, Papadimitriou and Vempala devised a way to connect the edge gadgets corresponding to the same variable in a net with certain expander-type properties. Informally, the structure of this net is such that any attempt to construct a TSP-tour that represents the value of a certain variable inconsistently in the gadgets corresponding to the equations where that variable occurs gives a tour of high cost. Intuitively, it is therefore always suboptimal to construct such "cheating" TSP-tours.

More formally, Papadimitriou and Vempala introduces the notion of a *b-pusher* [15, Definition 1] to precisely describe the structure that is needed to thwart "cheating" TSP-tours: A d-regular bipartite graph with vertex set $V_1 \cup V_2$ is called a *b-pusher* if, for any partition of V_1 into subsets U_1, S_1, T_1 and any partition of V_2 into subsets U_2, S_2, T_2 such that there are no edges from vertices in U_1 to vertices in U_2, the number (T_1, T_2) of edges between vertices in T_1 and T_2 satisfies

$$\left(b + \frac{1}{2}\right)(T_1, T_2) \geq \min\bigl\{|U_1| + |T_2|, |T_1| + |U_2|\bigr\} - \left(b - \frac{1}{2}\right)\bigl(|S_1| + |S_2|\bigr).$$

Papadimitriou and Vempala establish the existence of 6-regular 2-pushers [15, Theorem 5.1] and use such graphs to construct the precise coupling between different edge gadgets.

4.1 Modifications for bounded metrics

An inspection of the details of the Papadimitriou-Vempala construction shows that it, essentially, uses a metric which is bounded, in our sense of the word, by some constant that depends on ε. Qualitatively, their result is therefore of the form "there exists a constant c such that for every $\varepsilon > 0$ it is hard to approximate TSP within $c - \varepsilon$ in instances with metrics bounded by $B(\varepsilon)$". Our result in this paper is, again qualitatively, that the order of the quantifiers may be reversed, i.e., our result is of the form "there exists constants B and c such that for every $\varepsilon > 0$ it is hard to approximate TSP within $c - \varepsilon$ in instances with metrics bounded by B". Quantitatively, Papadimitriou and Vempala [15] have $c = 117/116$ for the asymmetric TSP and $c = 220/219$ for the symmetric TSP. For our case, the result is a trade-off between B and c. We settle for $B = 8$ which gives $c = 135/134$ for the asymmetric TSP and $c = 389/388$ for the symmetric TSP.

As mentioned in the caption of Fig. 25, the edge gadgets devised by Papadimitriou and Vempala [15] contain edges with very small weight. Specifically, the weight of the lightest edge in the instance is negligible compared to the constant ε in the main hardness result. In our model for bounded metrics, we only allow distances that are integers between one and some bound B. Consequently, we must modify the bridges in the edge gadgets so that they contain a edges of weight one instead of aL edges of weight $1/L$. This modification implies that the analysis must be modified. In particular, the so called "doubly traversed bridges", that incur an extra cost of $a + b$ in the Papadimitriou-Vempala construction, only incur a cost of $a + b - 2$ in our case. We believe that it is more natural to view those bridges as a kind of "semitraversed edge gadget" in our case. This change implies that a certain trick used by Papadimitriou and Vempala to associate a larger cost with the semitraversed edge gadgets does not work.

To conclude, we obtain weaker bounds on the cost incurred by "cheating TSP tours" in our case. This means that we cannot use the 6-regular 2-pushers used by Papadimitriou and Vempala—to use the straightforward reduction, we would instead need 2.5-pushers. It is easy to prove that 8-regular 2.5-pushers exist. However, using 8-regular graphs instead of 6-regular ones gives weaker approximation hardness results. To improve our results somewhat, we use a slightly more elaborate reduction, that does not need pushers but bipartite graphs with slightly weaker properties. As the final link in the proof of our hardness results, we show that there exist 7-regular graphs with the properties we need for our analysis to go through.

5 The hardness of (1,B)-ATSP

The purpose of this section is to show that the Papadimitriou-Vempala construction can be analysed also in the setting of *bounded metrics* with only small modifications. Specifically, we prove the following result:

Theorem 5.1. *For any sufficiently small constant* $\varepsilon > 0$, *there exists for any large enough integer m instances of (1,8)-ATSP with* $113m$ *cities such that: 1) Either there is a TSP tour with length at most* $(134 + \varepsilon)m$ *or else every TSP tour has length at least* $(135 - \varepsilon)m$. *2) It is* ***NP****-hard to distinguish these two cases.*

The proof of this theorem follows from Lemmas 5.1 and 5.2 described below.

We describe our instance of (1,B)-ATSP by constructing a weighted directed graph and then let the (1,B)-ATSP instance have the nodes of this graph as cities. In this paper we denote by $\ell(u, v)$ the distance from u to v in this weighted graph and define the distance between two cities u and v is the (1,B)-ATSP instance, denoted by $c(u, v)$, as $c(u, v) = \min\{\ell(u, v), B\}$.

5.1 The gadgets

The gadgets are parameterised by the parameters a, b and d; they will be specified later. The equation gadget for equations of the form $x + y + z = 0$ is shown in Fig. 24. The following property of the equation gadget was established by Papadimitriou and Vempala [15]:

Proposition 5.1. *There is a Hamiltonian path of length four through the gadget only if zero or two of the ticked edges are traversed. All other traversals have cost at least five.*

The equation gadgets are connected in a circle by identifying vertex B in one gadget with vertex A in the next gadget in the circle.

The ticked edges in Fig. 24 are gadgets themselves. This gadget is shown in Fig. 25. Each of the bridges is shared between two different edge gadgets, one corresponding to a positive occurrence of the literal and one corresponding to a negative occurrence. The precise coupling is provided by a certain d-regular bipartite multigraph. Specifically, proceed as follows for each literal x: Let k be the number of occurrences of x (and therefore also of $\bar{x}$); Take a bipartite d-regular multigraph with vertex set $V_1 \cup V_2$ ($|V_1| = |V_2| = k$); Label the vertices in V_1 with the occurrences of x and the vertices in V_2 with the occurrences of $\bar{x}$; Let a positive and a negative occurrence correspond to the same edge gadget if there is an edge between the corresponding vertices in the bipartite graph—the order of the occurrences inside the edge gadget is not important. Later, we describe some additional required properties of the bipartite multigraph, for now it only remains to mention that it can be constructed in constant time since it is of constant size.

5.2 Constructing a tour from an assignment

Consider a system of linear equations with the properties described in Theorem 4.1 and an instance of $(1,B)$-ATSP constructed from it as described in § 5.1. Let π be an assignment to the variables in the system of linear equations and consider the tour that 1) For each variable x traverses the edge gadget corresponding to x as shown in Fig. 26 if $\pi(x) = 0$ and as shown in Fig. 27 if $\pi(x) = 1$. 2) For each equation gadget enters each equation gadget at node A, takes the shortest possible way to B under the condition that the ticked edges are traversed as prescribed by the traversals of the edge gadgets, and then exits the equation gadget at node B.

Since there are $2m$ equations in the system of linear equations, the number of cities contained in the equation gadgets is $4 \cdot 2m = 8m$. Similarly, since every edge gadget is shared between two equation gadgets, there are $2m \cdot \frac{3}{2}d(a+1) = 3md(a+1)$ cities inside the equation gadget.

The length of the tour described above "inside" the edge gadgets is $d(a+b)$. The "extra" cost of one that comes from the two "outermost" horizontal edges in Fig. 25 is attributed to the equation gadget; in this way we can assign a cost of one to all edges in Fig. 24. Since there are $2m$ equations, three edge gadgets per equation gadget, and every edge gadget is shared between two equation gadgets, it follows that the total cost of the tour inside the edge gadgets is $3md(a+b)$. Considering an arbitrary equation gadget, the path from A to B in a tour constructed as described above has length four if the corresponding equation in the system of linear equations is satisfied by the assignment π and length five otherwise. (Strictly speaking, it is impossible to have three traversed edge gadgets in an equation gadget, since this does not result in a TSP tour. However, we can regard the case when the tour of the third edge gadget leaves the edge gadget by jumping directly to the exit node of the equation gadget as a tour with three traversals; such a tour gives a cost of five, in addition to the cost attributed to the edge gadgets.) Hence, the total cost accounted to the equation gadgets is $8m + u$, where u is the number of unsatisfied equations. We summarise the above discussion:

Lemma 5.1. *Consider a system of linear equations with the properties described in Theorem 4.1 and an instance of (1, B)-ATSP constructed from it as described in § 5.1. This instance contains* $3md(a+1) + 8m$ *cities. Given an assignment to the variables in the system of linear equations that satisfies all but u equations, the tour produced from this assignment as described above has length* $3md(a+b) + 8m + u$.

5.3 Constructing an assignment from a tour

The main challenge now is to prove that the above correspondence between the length of the optimum tour and the number of unsatisfied equation holds also when we drop the assumption that the tour is shaped in the intended way. Specifically, the aim is to show the following:

Lemma 5.2. *Consider a system of linear equations with the properties described in Theorem 4.1 with δ sufficiently small and an instance of (1, B)-ATSP constructed from it as described in § 5.1 with $a = 4$, $b = 2$, $d = 7$, and $B = 8$. Any TSP tour of length $3md(a + b) + 8m + u$ in this instance can be used to construct in polynomial time an assignment satisfying all but at most u equations.*

Our proof uses three technical lemmas. The first one shows that any tour can be transformed into a tour with a certain behaviour inside the bridges. The second lemma lower bounds the additional cost caused by non-standard traversals of an edge gadget and the last lemma establishes that the bipartite graph used has a certain expansion-related property.

Lemma 5.3. *Consider a system of linear equations with the properties described in Theorem 4.1 and an instance of (1, B)-ATSP constructed from it as described in § 5.1. If $B \geq a$, any TSP tour in such an instance can be transformed in polynomial time into a tour with smaller, or equal, length with the following properties:*

1) Let (u, v) be an edge of the tour and suppose that u and v both belong to the same bridge. Then u and v are neighbours in the graph defining the (1, B)-ATSP instance.

2 Let u and v be neighbours on the same bridge and assume that there is no edge between u and v in the tour. Let (u, u') and (v, v') be edges of the tour and assume that $c(u, u') = \ell(u, u')$ and that $c(v, v') = \ell(v, v')$. Then the shortest path from u to u' does not intersect the shortest path from v to v'.

Definition 5.1. *A bridge has a* defined traversal *if the tour restricted to the bridge is a path of length a; otherwise the bridge has an* undefined traversal.

Definition 5.2. *An edge gadget is* traversed *if all bridges have defined traversals and the connection edges (horizontal in Fig. 25) are traversed by the tour; it is* untraversed *if all bridges have defined traversals and none of the the connection edges are traversed by the tour. All other edge gadgets are* semitraversed*.*

Lemma 5.4. *Consider a system of linear equations with the properties described in Theorem 4.1 and an instance of (1, B)-ATSP constructed from it as described in § 5.1. From a tour with the properties guaranteed by Lemma 5.3, it is possible to associate a cost of at least* $\min\{a/2, b, a/2 + b/2 - 1\}$ *with every semitraversed edge gadget given that* $B \geq \max\{3b, a + b, 2a + b - 2\}$.

Lemma 5.5. *For every large enough constant k, there exists a 7-regular bipartite multigraph with vertex set $V_1 \cup V_2$ ($|V_1| = |V_2| = k$) such that for every partition of V_1 into sets T_1, U_1 and S_1 and every partition of V_2 into sets T_2, U_2 and S_2 such that there are no edges from T_1 to T_2, and there are no edges from U_1 to U_2,*

$$2\big(|S_1| + |S_2|\big) \geq \min\big\{k, |U_1| + |T_2| + |S_1| + |S_2|, |U_2| + |T_1| + |S_1| + |S_2|\big\}.$$

Before proving these lemmas, we show that they give—by appropriate choice of parameters—the desired connection between the length of an arbitrary TSP tour and the number of satisfied equations in the corresponding system of linear equations.

Proof of Lemma 5.2. Set $a = 4$, $b = 2$, $d = 7$, and $B = 8$. Then it follows from Lemma 5.4 that every semitraversed edge gadget incurs a cost of at least two.

For every variable x, let the bipartite multigraph used to construct the edge gadget have the property stated in Lemma 5.5 with k equal to the number of occurrences of x (and hence also of $\bar{x}$). Lemma 5.5 asserts that such graphs exist for sufficiently large k; hence we must assume that δ in Theorem 4.1 is small enough.

The assignment to an arbitrary variable x is constructed as follows: Suppose that x occurs k times positively and k times negatively. Let T_1 be the set of traversed positive occurrences and T_2 be the set of traversed negative occurrences. Define U_1, U_2, S_1, and S_2 similarly. If $|S_1| + |S_2| \geq k/2$, set $\pi(x) = 0$ with probability $1/2$ and $\pi(x) = 1$ with probability $1/2$. Otherwise define $\pi(x)$ deterministically as follows: If $|T_1| + |U_2| \geq |T_2| + |U_1|$, let $\pi(x) = 1$, otherwise let $\pi(x) = 0$. The resulting probabilistic assignment is then derandomised, using the method of conditional probabilities, to produce an assignment satisfying at least as many equations as the expected number of equations satisfied by π.

We need to prove that there is at most one unsatisfied equation per unit of the "extra" cost u, i.e., per unit of the cost in addition to the "normal" cost of $3md(a+b)$ for the edge gadgets and $8m$ for the equation gadgets. To this end, we show that it is possible to associate a cost of at least $1/2$ with every equation containing a variable that has been set at random and a cost of at least 1 with every other equation that could be unsatisfied by π.

Let x be an arbitrary variable and suppose that x occurs $2k$ times. Define T_1, T_2, U_1, U_2, S_1, and S_2 as above. Since variables are given probabilistic assignments only when $|S_1| + |S_2| \geq k/2$ and every semitraversed edge gadgets incurs an extra cost of 2, there is an extra cost of at least $1/2$ associated with every equation containing a variable that has been assigned a random value. Since every such eqation is satisfied with probability $1/2$, no matter the number of variables in the equation that were given random assignments, the extra cost attributed to variables with a random assignment is equal to the expected number of unsatisfied equations from this assignment.

Consider next the case when $|S_1| + |S_2| \leq k/2$. Since Lemma 5.5 guarantees that the extra cost incurred by the semitraversed occurrences of x and $\bar{x}$ is no less than

$$\min\left\{|U_1| + |T_2| + |S_1| + |S_2|, |U_2| + |T_1| + |S_1| + |S_2|\right\}$$

in this case, the extra cost incurred by the semitraversed occurrences pays for the potential unsatisfaction of every equation that contains a variable that has been assigned a value contradicting the traversal of the corresponding edge gadget. The only remaining possibility for equations that are unsatisfied under π comes from equations where all variables have been assigned values according to the traversal of the corresponding edge gadgets and that assignment does not satisfy the equation. However, for such equations, there is an extra cost of one in the equation gadget according to Proposition 5.1. ∎

5.4 Proof of Lemma 5.3

To ensure property 1, repeat the following for all edges (u, v) of the tour such that u and v belong to the same bridge but are not neighbours in the graph: Redefine the tour, so that instead of jumping from u directly to v, the tour follows the shortest path from u to v in the graph defining the instance. Since $B \geq a$ this does not increase the length of the tour. This change will make the tour pass through some cities—the cities that are on the shortest path from u to v in the graph—twice. For all such cities w, do the following: Let w' be the city visited immediately before w and w'' be the city visited immediately after w. Then replace the edges (w', w) and (w, w'') by the single edge (w', w'') in the tour. By triangle inequality this procedure does not increase the length of the tour.

To ensure property 2, repeat the following for all vertices u and v that belong to the same bridge but for which there is no edge between u and v in the tour: Let u' and v' be defined as in the formulation of the lemma. If the shortest path from u to u' does not intersect the shortest path from v to v', no transformation of the tour is needed. Otherwise, the fact that u and v are on the same bridge implies that we can assume without loss of generality that the shortest path from u to u' passes v (otherwise we just exchange u and v in the argument). We then redefine the tour, so that instead of jumping from u directly to u', the tour follows the shortest path from u to u' in the graph defining the instance. As above, for every node w on the shortest path from u to u' (including v), let w' be the city visited immediately before w and w'' be the city visited immediately after w and replace the edges (w', w) and (w, w'') by the single edge (w', w'') in the tour. By triangle inequality this procedure does not increase the length of the tour.

5.5 Proof of Lemma 5.4

Consider a semitraversed edge gadget. We now argue by case analysis that it introduces an extra cost in addition to the "standard" cost of $a + b$ per bridge. For accounting purposes, we use the convention that this standard cost corresponds to a cost of $b/2$ for the incoming edge of the tour plus a cost of $b/2$ for the outgoing edge of the tour plus a cost of a for the traversal of the bridge itself. When analysing the extra cost due to semitraversals, it is important to attribute this extra cost to both edge gadgets that take part in the semitraversal. Sometimes this means two different edge gadgets that represent the same literal x (or $\bar{x}$); sometimes this means the two edge gadgets that cross at a certain bridge. For "long" jumps, i.e., cases when the tour traverses an edge (u, v) with cost $c(u, v) \neq \ell(u, v)$, a cost of $B/2$ is attributed to both of the involved bridges.

Lemma 5.6. *Given that $B \geq 2a + b - 2$, it is possible to associate a cost of at least of at least $a/2 + b/2 - 1$ with every edge gadget that becomes semitraversed because of a bridge having an undefined traversal.*

Proof. We first consider the case when the metric is not bounded; we will show later how to extend the argument to cover also bounded metrics. In the unbounded case, the distance between two vertices u and v is exactly the length of the shortest path from u to v in the graph defining the instance.

Since the bridge has an undefined traversal, there must be two adjacent cities u and v that are not neighbours in the tour. Consider the edges (u, u') and (v, v') in the tour—thanks to Lemma 5.3 we can assume that neither u' nor v' belong to the bridge.

The tour must visit all cities on the bridge. Therefore the total cost of the tour on the bridge is, according to our convention, at least $2a + 2b - 2$, which gives an extra cost of $a + b - 2$.

When the metric is bounded by some bound B, a case analysis shows, that if $B/2 \geq a + b/2 - 1$ it follows that the cost of the tour on a bridge with an undefined traversal is still at least $2a + 2b - 2$. Intuitively, this states that the case shown to the right in Fig. 28 with the dotted line replaced by a "jump" following some edge with cost B is the worst case, i.e., the case with lowest extra cost.

Since a bridge containing an undefined traversal makes both edge gadgets passing through it semitraversed, the proof of the lemma is complete. ∎

Lemma 5.7. *Given that* $B \geq \max\{a + b, 3b\}$ *it is possible to associate a cost of at least* $\min\{a/2, b\}$ *with every edge gadget that becomes semitraversed because of a bridge with a defined traversal.*

Proof. We first consider the case when the metric is not bounded and show later how to extend the argument to cover also bounded metrics. In the unbounded case, the distance between two vertices u and v is exactly the length of the shortest path from u to v in the graph defining the instance.

Consider first a bridge traversed from left to right but where the connecting edge leaving the bridge is not traversed by the tour. Hence, the tour makes a jump leaving the bridge. There are three sub-cases:

The tour goes down (Fig. 29). The earliest available free city is a distance of $2b$ away; that blocks the tour leaving the right bridge, forcing it to also make a jump of at least $2b$. The next available free city is a distance of $3b$ away. Both these cases give a total extra cost of $2b$.

The tour goes forwards (Fig. 30). The earliest available free city is a distance of $a + b$ away, giving a total extra cost of a.

The tour goes backwards (Fig. 31). The earliest available free city is a distance of $a + b$ away, giving a total extra cost of a.

Next, consider a bridge traversed from left to right where the connecting edge *entering* the bridge is not traversed by the tour. Again, there are three sub-cases.

The tour comes from above (Fig. 29). The earliest available free city is a distance of $2b$ away, but that blocks the tour entering the right bridge, forcing it to also make a jump of at least $2b$. The next available free city is a distance of $3b$ away. Both these cases give a total extra cost of $2b$.

The tour comes from the front (Fig. 30). The earliest available free city is a distance of $a + b$ away, giving a total extra cost of a.

The tour comes from behind (Fig. 31). The earliest available free city is a distance of $a + b$ away, giving a total extra cost of a.

So far, the analysis only considered unbounded metrics. Note first, however, that if $B \geq \max\{3b, a + b\}$, the above argument is valid. If the tour makes a larger jump than the shortest possible jumps stated above, the additional cost can never decrease, thanks to the triangle inequality. Next, note that if the tour leaves a bridge with a defined traversal with a "long jump", i.e., following an edge (u, v) where $c(u, v) \neq \mathscr{C}(u, v)$, that particular bridge can only cause one of the edge gadgets passing through it to be semitraversed and hence we can allocate the entire net cost of $B/2 - b/2$ to that edge gadget. If $B \geq \max\{3b, a + b\}$, then $B/2 - b/2 \geq \max\{a/2, b\}$, hence the lemma holds also in this case. ∎

Note, finally, that the above analysis is valid also for tours such that a "long jump" may start in a semitraversed gadget with no undefined traversal and end in an undefined traversal, and vice versa.

5.6 Proof of Lemma 5.5

The proof uses the same main idea as the proof that establishes existence of 6-regular 2-pushers: It uses the fact that it is possible to lower bound the size of neighbours to any given set of vertices in d-regular bipartite graphs. For a set W, let $N(W)$ denote the neighbours of W in the graph. With this notation, a recent study of Engebretsen [8] implies that there exist, for every large enough k, a 7-regular bipartite multigraph with vertex set $V_1 \uplus V_2$ ($|V_1| = |V_2| = k$) such that for every $W \subseteq V_1$ and every $W \subseteq V_1$, the following holds:

$$|W| \leq 0.15k \implies |N(W)| > 8|W|/3,$$

$$0.15k \leq |W| \leq 0.60k \implies |N(W)| > 0.25k + |W|,$$

$$|W| \geq 0.60k \implies |N(W)| > 5k/8 + 3|W|/8,$$

$$|W| \leq 0.31k \implies |N(W)| > 2|W|,$$

$$0.31k \leq |W| \leq 0.35k \implies |N(W)| > 0.31k + |W|,$$

$$|W| \geq 0.35k \implies |N(W)| > 31k/65 + 34|W|/65.$$

Our task is to prove that for every partition of the left vertices into sets T_1, U_1 and S_1 and every partition of the right vertices into sets T_2, U_2 and S_2 such that there are no edges from T_1 to T_2, and there are no edges from U_1 to U_2,

$$2\big(|S_1| + |S_2|\big) \geq \min\big\{k, |U_1| + |T_2| + |S_1| + |S_2|, |U_2| + |T_1| + |S_1| + |S_2|\big\}.$$

From now on, we use the shorthands $|T_1| = kt_1$, $|U_1| = ku_1$, $|S_1| = ks_1$, $|T_2| = kt_2$, $|U_2| = ku_2$, and $|S_2| = ks_2$. We can also assume without loss of generality that

$u_1 + t_2 \le t_1 + u_2$. Hence, we must show that

$$2s_1 + 2s_2 \ge \min\{1, u_1 + t_2 + s_1 + s_2\}. \tag{1}$$

We let $n(x)$ denote $\frac{1}{k}$ times the size of the neighbours of some set with size kx. Since there are no edges between T_1 and T_2 and there are no edges between U_1 and U_2, it follows that $s_1 \ge n(t_2) - u_1$. Similarly, $s_2 \ge n(u_1) - t_2$. Also, it is easy to see that $t_1 \le 1 - n(t_2)$ and that $u_2 \le 1 - n(u_1)$. These observations are used repeatedly in the following, somewhat overlapping, case analysis that covers all possible values of u_1 and t_2.

Case I: $u_1 \le 0.31$ and $t_2 \le 0.31$. In this case $s_1 + s_2 \ge n(t_2) - u_1 + n(u_1) - t_2 \ge u_1 + t_2$, which implies (1).

Case II: $0.15 \le u_1 \le 0.60$ and $0.15 \le t_2 \le 0.60$. Since $s_1 \ge n(t_2) - u_1 \ge t_2 + \frac{1}{4} - u_1$ and $s_2 \ge n(u_1) - t_2 \ge u_1 + \frac{1}{4} - t_2$ in this case, it follows that $s_1 + s_2 \ge \frac{1}{2}$, which implies (1).

Case III: $u_1 \ge 0.35$ and $t_2 \ge 0.35$. Using the fact that $u_1 + t_2 \le t_1 + u_2 \le 2 - n(t_2) - n(u_1) \le \frac{68}{65} - \frac{34}{65}u_1 - \frac{34}{35}t_2$, or, equivalently, that $u_1 + t_2 \le \frac{68}{99} < 0.70$, we reach a contradiction since $u_1 + t_2$ must be at least 0.70 in this case. Hence this case cannot occur.

Case IV a: $u_1 \le 0.35$ and $t_2 \ge 0.60$. In this case $s_1 \ge n(t_2) - u_1 \ge \frac{3}{8}t_2 + \frac{5}{8} - u_1 \ge \frac{3}{8} \cdot \frac{3}{5} + \frac{5}{8} - \frac{7}{20} = \frac{1}{2}$, which implies (1).

Case IV b: $u_1 \ge 0.60$ and $t_2 \le 0.35$. In this case $s_2 \ge n(u_1) - t_2 \ge \frac{3}{8}u_1 + \frac{5}{8} - t_2 \ge \frac{3}{8} \cdot \frac{3}{5} + \frac{5}{8} - \frac{7}{20} = \frac{1}{2}$, which implies (1).

Case V a: $u_1 \le 0.15$ and $t_2 \ge 0.35$. In this case $s_1 \ge n(t_2) - u_1 \ge \frac{31}{65} + \frac{34}{65}t_2 - u_1 \ge \frac{31}{65} + \frac{34}{65} \cdot \frac{35}{100} - \frac{15}{100} = \frac{51}{100} > \frac{1}{2}$, which implies (1).

Case V b: $u_1 \geq 0.35$ and $t_2 \leq 0.15$. In this case $s_2 \geq n(u_1) - t_2 \geq \frac{31}{65} + \frac{34}{65}u_1 - t_2 \geq \frac{31}{65} + \frac{34}{65} \cdot \frac{35}{100} - \frac{15}{100} = \frac{51}{100} > \frac{1}{2}$, which implies (1).

Case VI a: $u_1 \leq 0.15$ and $0.31 \leq t_2 \leq 0.35$. In this case $s_1 \geq n(t_2) - u_1 \geq t_2 + 0.31 - u_1$ and $s_2 > \max\{n(u_1) - t_2, 0\} > \max\{\frac{8}{3}u_1 - t_2, 0\}$. This gives two sub-cases that together imply (1).

$t_2 \geq \frac{8}{3}u_1$: $s_1 + s_2 \geq s_1 \geq \frac{5}{8}t_2 + 0.31 \geq \frac{5}{8} \cdot 0.31 + 0.31 = \frac{403}{800} > \frac{1}{2}$.

$t_2 \leq \frac{8}{3}u_1$: $s_1 + s_2 \geq \frac{5}{3}u_1 + 0.31 \geq \frac{5}{8}t_2 + 0.31 > \frac{1}{2}$.

Case VI b: $0.31 \leq u_1 \leq 0.35$ and $t_2 \leq 0.15$. In this case $s_1 > \max\{n(t_2) - u_1, 0\} > \max\{\frac{8}{3}t_2 - u_1, 0\}$ and $s_2 \geq n(u_1) - t_2 \geq u_1 + 0.31 - t_2$. This gives two sub-cases that together imply (1).

$u_1 \geq \frac{8}{3}t_2$: $s_1 + s_2 \geq s_2 \geq \frac{5}{8}u_1 + 0.31 \geq \frac{5}{8} \cdot 0.31 + 0.31 = \frac{403}{800} > \frac{1}{2}$.

$u_1 \leq \frac{8}{3}t_2$: $s_1 + s_2 \geq \frac{5}{3}t_2 + 0.31 \geq \frac{5}{8}u_1 + 0.31 > \frac{1}{2}$.

6 The hardness of (1,B)-TSP

To adapt the construction from the § 5 to the symmetric case we change the gadgets; on a high level both the construction and the proof of correctness are as in the asymmetric case. The equation gadget is replaced with the gadget in Fig. 32; this gadget tests odd instead of even parity.

Proposition 6.1. *The only way to traverse the equation gadget in Fig. 32 with a tour of length five—if the edge gadgets count as length one—is to traverse an odd number of edge gadgets. All other traversals have length at least six.*

To construct a symmetric edge gadget, note that already the asymmetric edge gadget is in fact almost symmetric since the bridge in the asymmetric edge gadget is an undirected path of length a. Consider the following attempt to make an undirected edge gadget: Let the edges connecting the bridge with other bridges in the asymmetric edge gadget be undirected and connect the edge gadgets as in the asymmetric case. The resulting gadget penalises many, but not all, unwanted tours. In particular, the weakness with the above construction is that a path may, without any additional penalty, enter a bridge through an edge that is directed *towards* the bridge in the asymmetric version of the gadget and leave the same bridge along the *other* edge that is directed towards the bridge. To overcome this problem, we construct a symmetric version of the asymmetric bridge by hooking up three copies of the "symmetrised asymmetric bridge" described above in parallel and then rotating the resulting package 90° (see Fig. 33). We call the resulting structure a *symmetric bridge*.

Similar to the asymmetric case, we say that a symmetric bridge has a *defined traversal* if the tour restricted to the bridge traverses all three bridges and exactly two of the horizontal edges in Fig. 33. With $a = 4$, $b = 2$ and $B = 8$, the technical lemmas from § 5.5 can be used to show that any undefined traversal of the edge gadget gives an additional local cost of four, i.e., an additional local cost of two can be attributed to each of the two edge gadgets that meet at the symmetric bridge. Defining *traversed*, *untraversed* and *semitraversed* edge gadgets as in the asymmetric case, a case analysis similar to that in the proof of Lemma 5.7 then shows that a cost of at least two can be associated with each semitraversed symmetric edge gadget. As in the asymmetric case, the individual edge gadgets corresponding to the same variable are stitched together according to the edges in a d-regular bipartite multigraph with vertex set $V_1 \uplus V_2$ (where $|V_1| = |V_2| = k$ and $2k$ is the number of occurrences of the variable) that has the property that for every partition of V_1 into sets T_1, U_1 and S_1 and every partition of V_2 into sets T_2, U_2 and S_2 such that there are no edges from T_1 to T_2, and there are no edges from U_1 to U_2, it holds that

$$2\big(|S_1| + |S_2|\big) \geq \min\big\{k, |U_1| + |T_2| + |S_1| + |S_2|, |U_2| + |T_1| + |S_1| + |S_2|\big\}.$$

To summarise, the following lemma follows in the same way as in the asymmetric case:

Lemma 6.1. *Consider a system of linear equations with the properties described in Theorem 4.1 with δ sufficiently small and an instance of (1,B)-TSP constructed from it as outlined above with $a = 4$, $b = 2$, $d = 7$, and $B = 8$. A TSP tour of length $9md(a+b)+10m+u$ in this instance can be used to construct in polynomial time an assignment satisfying all but at most u equations.*

For the symmetric analogue of Lemma 5.1, note that a "jump" past an edge gadget actually requires following an edge of length $9md(a + b) + 1$ as the construction is described above. However, by adding for every edge gadget an edge of length two that is parallel with the edge gadget in the graph defining the TSP instance, it is easy to see that the following lemma holds:

Lemma 6.2. *Consider a system of linear equations with the properties described in Theorem 4.1 and an instance of (1,B)-TSP constructed from it as outlined above. Given an assignment to the variables in the system of linear equations that satisfies all but u equations, it is possible to construct a TSP tour with length $9md(a + b) + 10m + u$.*

Given the above lemmas, our second main theorem follows in exactly the same way as in the asymmetric case.

Theorem 6.1. *For any constant $\varepsilon > 0$, it is **NP**-hard to approximate (1,8)-TSP within $389/388 - \varepsilon$.*

7 Concluding remarks

There are two main conclusions from the work presented in this paper. First, the fact that it is relatively straightforward to adapt the construction devised by Papadimitriou and Vempala [15] to the case of bounded metrics shows that this latter construction is essentially local, in spite of the fact that it uses as a critical component edges with unbounded—but constant—length. This indicates that new ideas are needed to obtain hardness within factors that are $\omega(1)$, or even hardness within an arbitrarily large constant factor.

The second main conclusion is that simpler constructions and simpler proofs of correctness are needed in order to obtain hardness results that are substantially better than the currently best known ones. Current techniques have been pushed more or less to their limits. Also, earlier versions of this paper as well as earlier versions of [15] contained errors in the accounting of penalties due to non-standard traversals. In order to achieve stronger hardness results, some kind of more structured approach is probably necessary—more complicated gadget reductions and accounting procedures are bound to be even more sensitive to errors in the analysis than the construction of Papadimitriou and Vempala [15]. We believe that a direct PCP construction is the natural next step for constructing stronger approximation hardness results for TSP with triangle inequality.

Acknowledgements

We thank Santosh Vempala for many clarifying discussions on the subject of this paper. Also, the anonymous referees contributed with many valuable comments that helped improving the presentation of our results.

References

1. Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Márió Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

2. Piotr Berman and Marek Karpinski. $8/7$-approximation algorithm for $(1, 2)$-TSP. To appear in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Miami, Florida, 22–24 January 2006.

3. Piotr Berman and Marek Karpinski. On some tighter inapproximability results. In Jiří Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Proceedings of 26th International Colloquium on Automata, Languages and Programming*, volume 1644 of *Lecture Notes in Computer Science*, pages 200–209. Springer-Verlag, Prague, 11–15 July 1999.

4. Markus Bläser and Bodo Siebert. Computing cycle covers without short cycles. In *Proceedings of the 9th Annual European Symposium on Algorithms*, Lecture Notes in Computer Science. Springer-Verlag, Århus, 28–31 August 2001.

5. Hans-Joachim Böckenhauer and Sebastian Seibert. Improved lower bounds on the approximability of the traveling salesman problem. *RAIRO Theoretical Informatics and Applications*, 34(3):213–255, 2000.

6. Janka Chlebíková and Miroslav Chlebík. Approximation hardness for small occurrence instances of NP-hard problems. Technical Report TR02-073, Electronic Colloquium on Computational Complexity, December 2002.

7. Nicos Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report CS-93-13, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, 1976.

8. Lars Engebretsen. Bipartite multigraphs with expander-like properties. Technical Report arXiv:math.CO/0412114v1, arXiv.org, 6 December 2004.

9. Lars Engebretsen. An explicit lower bound for TSP with distances one and two. *Algorithmica*, 35(4):301–319, 2003.

10. Lars Engebretsen and Marek Karpinski. Approximation hardness of TSP with bounded metrics. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Proceedings of 28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 201–212. Springer-Verlag, Crete, Greece, 8–12 July 2001.

11. Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, July 2001.

12. Heim Kaplan, Moshe Lewenstein, Nira Shafrir, and Maxim Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *Journal of the ACM*, to appear.

13. Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

14. Christos H. Papadimitriou and Santosh Vempala. On the approximability of the traveling salesman problem. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 126–133. Portland, Oregon, 21–23 May 2000.

15. Christos H. Papadimitriou and Santosh Vempala. On the approximability of the traveling salesman problem, November 2004. Manuscript.

16. Christos H. Papadimitriou and Mihalis Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, February 1993.

17. Luca Trevisan. When Hamming meets Euclid: The approximability of geometric TSP and MST. *SIAM Journal on Computing*, 30(2):475–485, 2000.

List of Figures

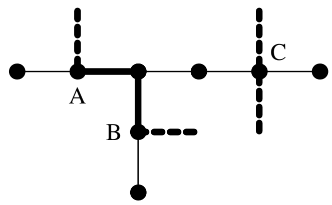


Figure 1. The above figure contains two partial tours—one entering the graph at A and leaving at B, and one both entering and leaving at C. The nodes A and B are endpoints and C is a double endpoint. The dashed parts of the tour denotes parts where the tour traverses edges with weight two.

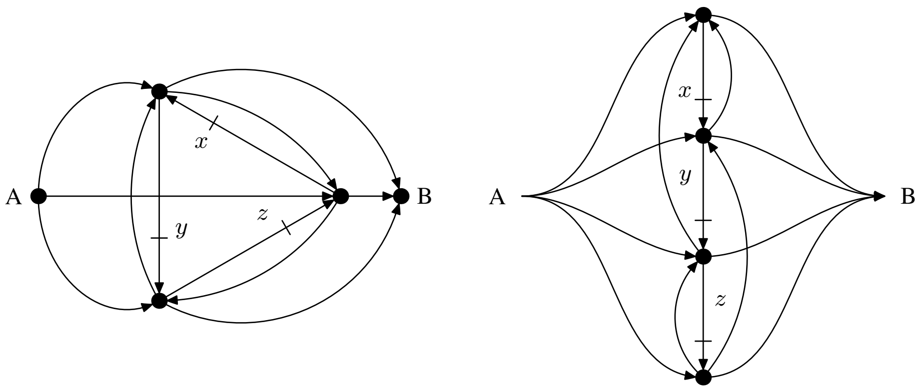


Figure 2. The gadget for equations of the form $x + y + z = 0$ (left) and $x + y + z = 1$ (right). There is a Hamiltonian path from A to B only if an even (left) or odd (right) number of the ticked edges is traversed.

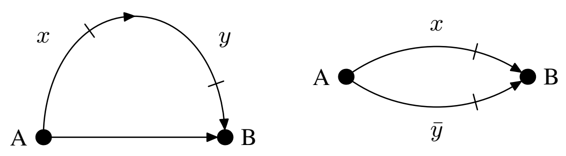


Figure 3. The gadget for equations of the form $x + y = 0$ (left) and $x + \bar{y} = 1$ (right). There is a Hamiltonian path from A to B only if an even (left) or odd (right) number of the ticked edges is traversed.

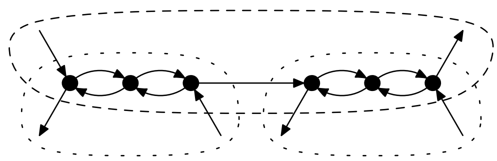


Figure 4. The gadget used to connect the ticked edges that correspond to the same variable x. The ticked edges corresponding to the two positive occurrences of x are represented by the parts enclosed in the dotted curves and the ticked edge corresponding to $\bar{x}$ is represented by the part enclosed in the dashed curve.

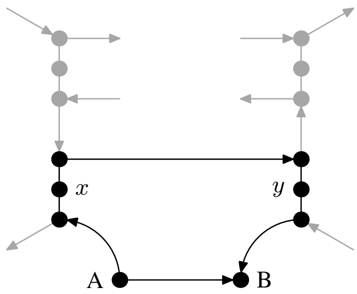
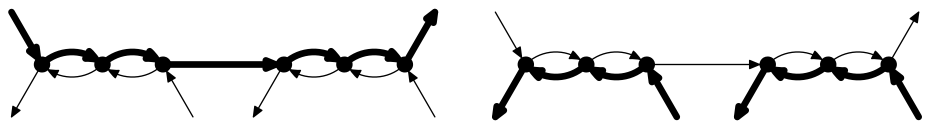


Figure 5. A more detailed view of the gadget for equations of the form $x + y = 0$. In this figure the ticked edges have been expanded to show the consistency checkers. The black edges correspond to the gadget shown in Fig. 3

Figure 6. The cost of the gadgets for equations of the form $x + y + z = 1$ is lowered by the above transformation. The figure to the left shows the connection between two such gadgets as it is obtained by joining B in one gadget as shown in Fig. 2 with A in another such gadget. The figure to the right shows how this connection is actually implemented.



(a) (b)

Figure 7. The figure above shows the “intended” traversals of the consistency checkers. The traversal (a) is to be interpreted as assigning 0 to the corresponding variable; traversal (b) as assigning 1.

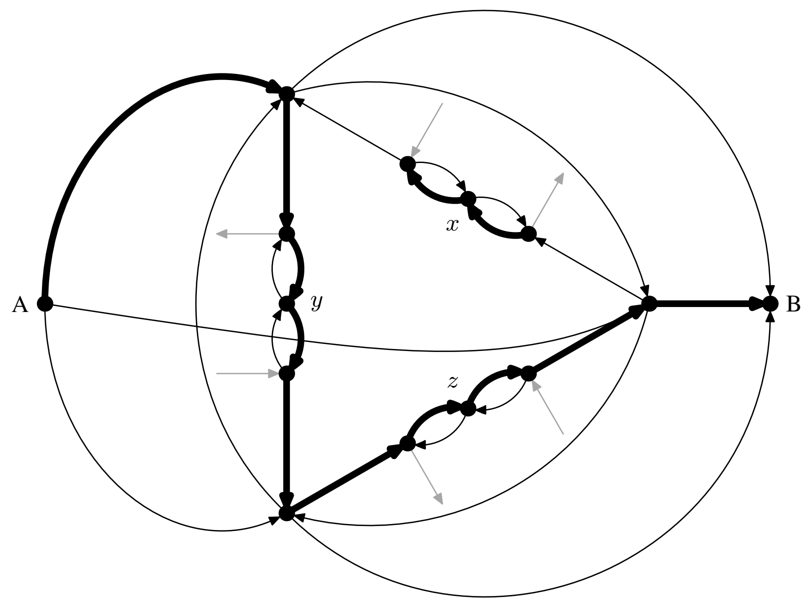


Figure 8. A more detailed view of how the tour corresponding to an assignment π such that $\pi(x) = \pi(y) = \pi(z) = 1$ traverses the gadget for equations of the form $x + y + z = 0$. In this figure the ticked edges have been expanded to show the consistency checkers. The black edges correspond to the gadget shown in Fig. 2. Note that the tour has two endpoints in the consistency checker corresponding to x.

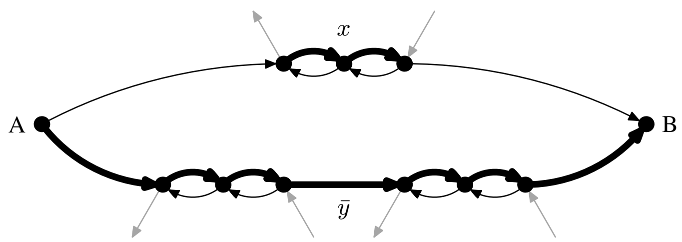


Figure 9. A more detailed view of how the tour corresponding to an assignment π such that $\pi(x) = 1$ and $\pi(y) = 0$ traverses the gadget for equations of the form $x + \bar{y} = 1$. In this figure the ticked edges have been expanded to show the consistency checkers. The black edges correspond to the gadget shown in Fig. 3. Note that the tour has two endpoints in the consistency checker corresponding to x.

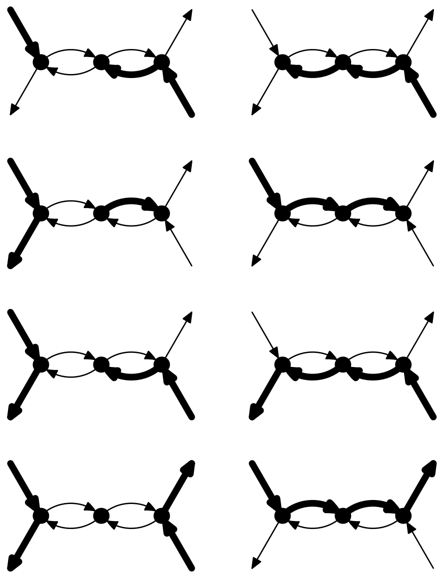


Figure 10. It is possible to change the traversals in the left column into the traversals in the right column without increasing the total number of endpoints in the graph.

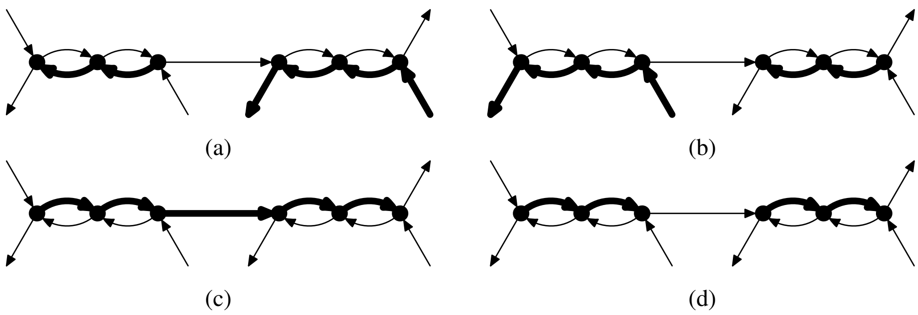


Figure 11. The traversals shown above may still be present in the tour after the "normalization" described in Lemmas 2.2–2.5.

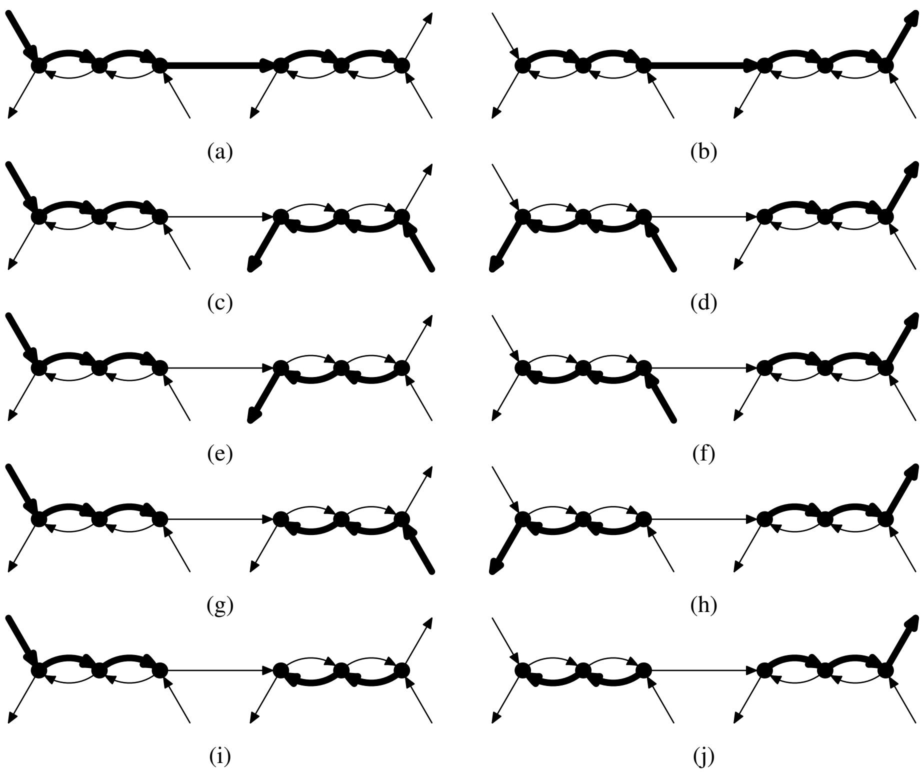


Figure 12. If the negative occurrence in the consistency checker is semitraversed, the checker has to be traversed as shown above.

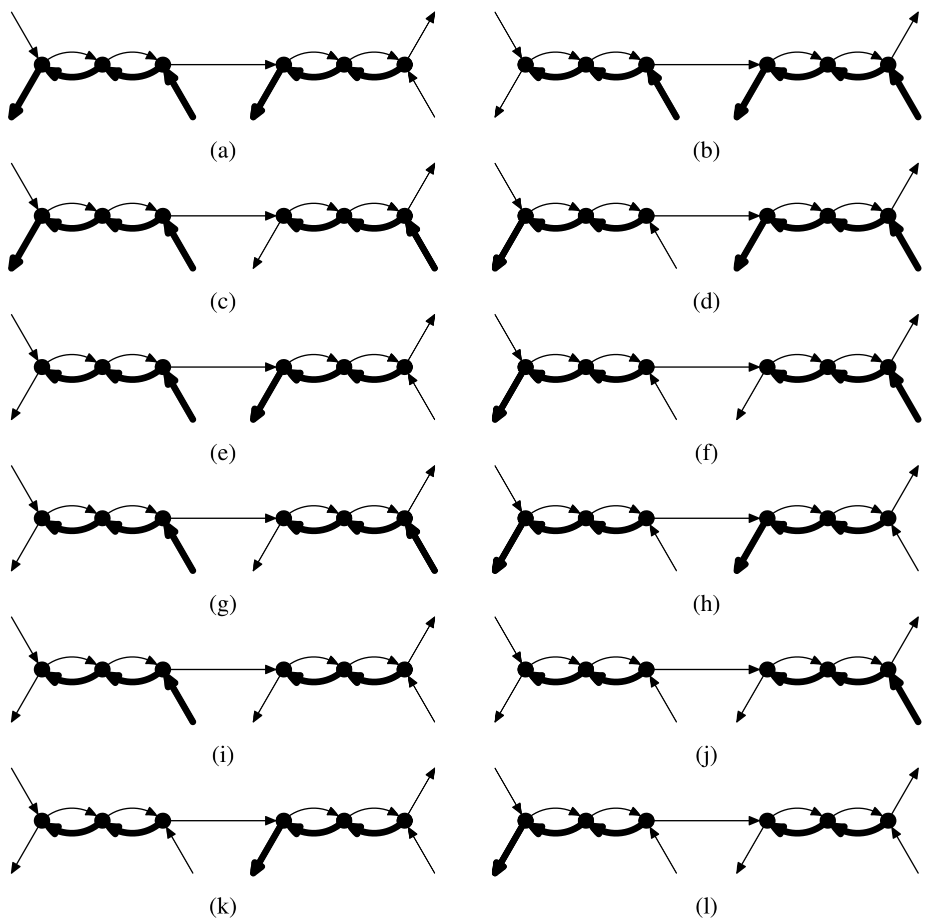
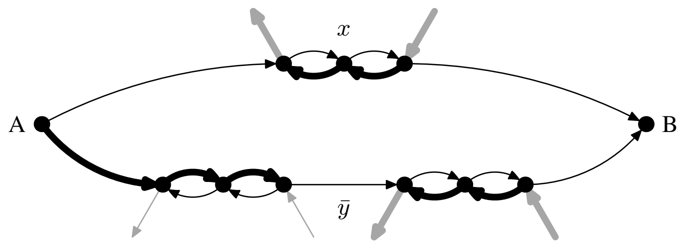
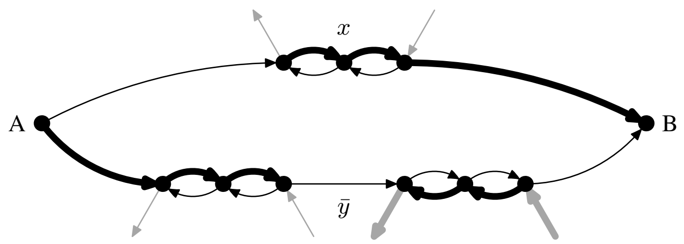


Figure 13. If there is at least one semitraversed occurrence in the consistency checker but the upper level is untraversed, the checker has to be traversed as shown above.



(a)



(b)

Figure 14. A gadget for equations of the form $x + \bar{y} = 1$ where the variable gadget corresponding to $\bar{y}$ is traversed as shown in Fig. 12c.

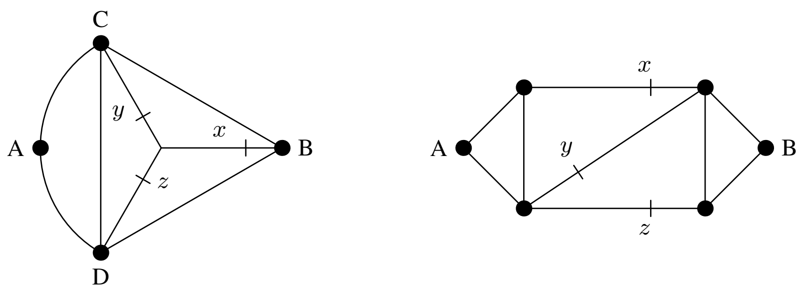


Figure 15. The gadget for equations of the form $x + y + z = 0$ (left) and $x + y + z = 1$ (right). There is a Hamiltonian path from A to B only if an even (left) or odd (right) number of ticked edges is traversed.

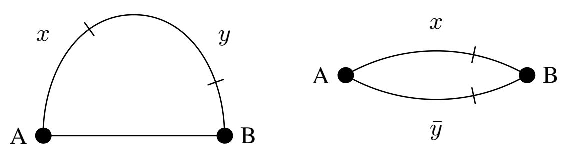


Figure 16. The gadget for equations of the form $x + y = 0$ (left) and $x + \bar{y} = 1$ (right). There is a Hamiltonian path from A to B only if an even (left) or odd (right) number of the ticked edges is traversed.

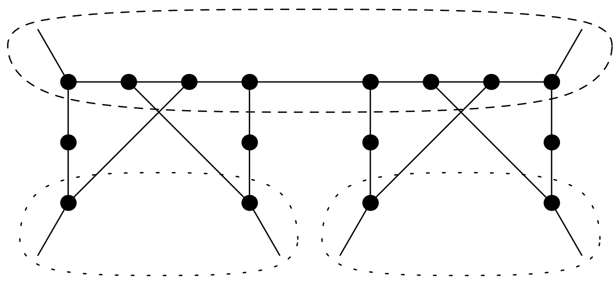


Figure 17. The gadget used to connect the ticked edges that correspond to the same variable x. The ticked edges corresponding to the two positive occurrences of x are represented by the parts enclosed in the dotted curves and the ticked edge corresponding to the negative occurrence is represented by the part enclosed in the dashed curve.

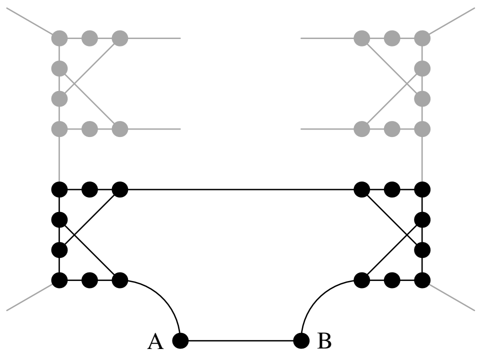


Figure 18. A more detailed view of the gadget for equations of the form $x + y = 0$. In this figure the ticked edges have been expanded to show the consistency checkers. The black edges correspond to the gadget shown in Fig. 16

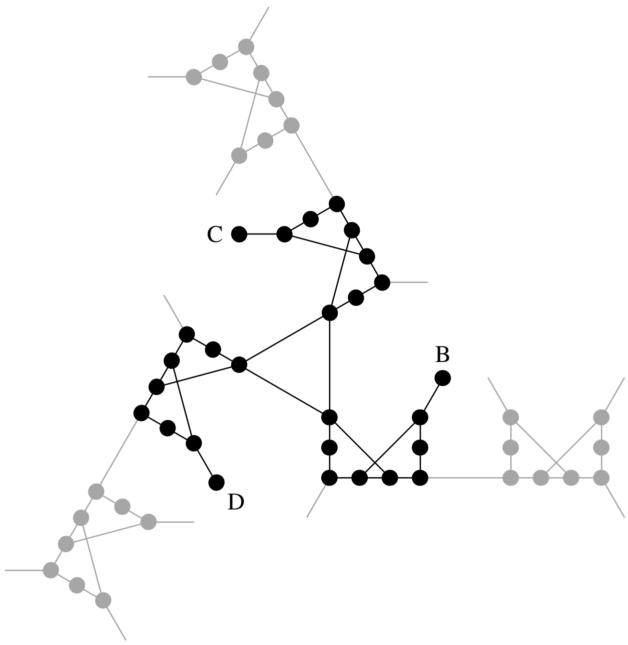
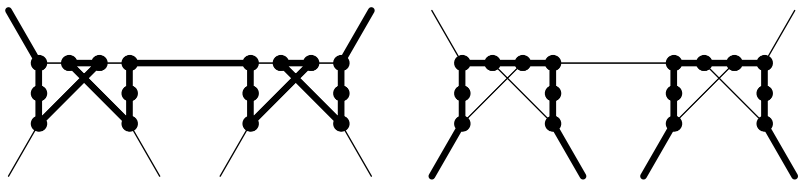


Figure 19. A more detailed view of the gadget for equations of the form $x+y+z=0$. The figure shows how the three variable gadgets meet in the center of the gadget. The black edges above correspond to the ticked edges in Fig. 15 and the three labeled nodes above are the same as the corresponding nodes in Fig. 15.



(a) (b)

Figure 20. The figure above shows the "intended" traversals of the consistency checkers. The traversal (a) is to be interpreted as assigning 0 to the corresponding variable; traversal (b) as assigning 1.

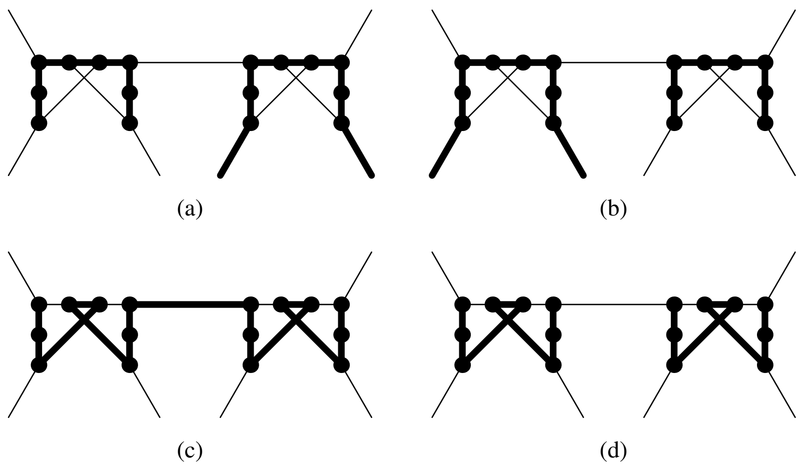


Figure 21. If there are no semitraversed occurrences in the consistency checker it may still be traversed as shown above.

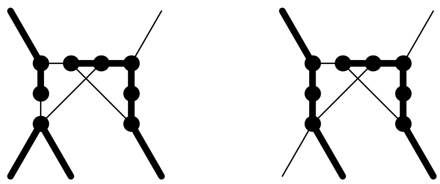


Figure 22. Some consistency checkers have double connection edges at one point, see also Fig. 19. By local transformations according to the above pattern we can assume that at most one of the double edges are traversed.

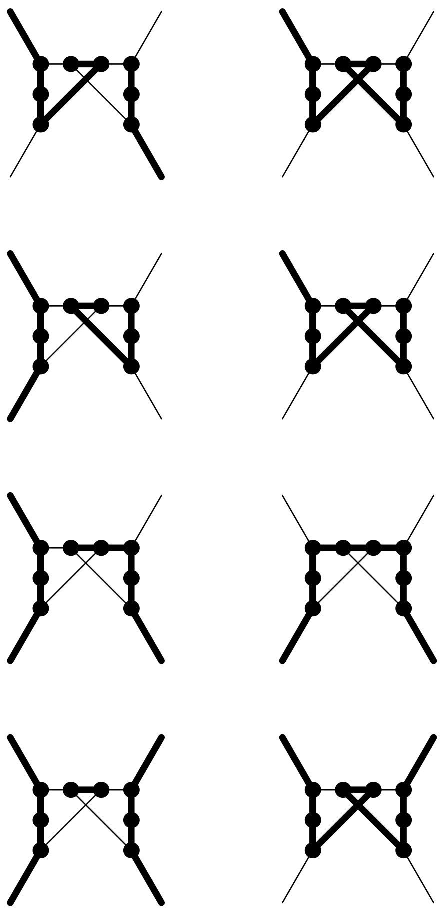


Figure 23. It is possible to change the traversals in the left column into the traversals in the right column without increasing the total number of endpoints in the graph.

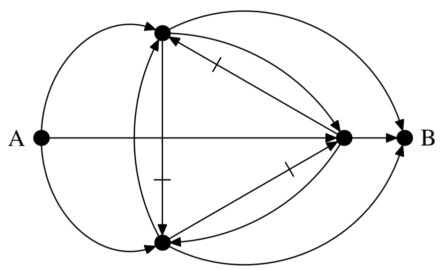


Figure 24. The gadget for equations of the form $x+y+z=0$. There is a Hamiltonian path from A to B only if zero or two of the ticked edges, which are actually gadgets themselves (Fig. 25), are traversed. The non-ticked edges have weight 1.

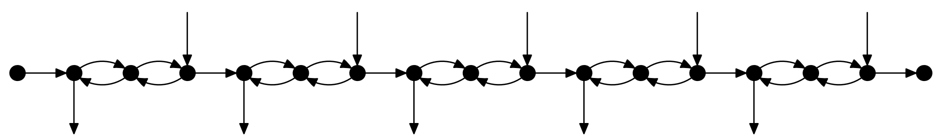


Figure 25. The edge gadget consists of d bridges. Each of the bridges are shared between two different edge gadgets. Each bridge consist of aL undirected edges of weight $1/L$ each. In the construction of Papadimitriou and Vempala [15], L is a (very) large integer constant—in our construction for bounded metrics, $L = 1$. The edges between bridges have weight b, the first horizontal edge has weight $\lfloor \frac{b+1}{2} \rfloor$, and the last horizontal edge has weight $\lceil \frac{b+1}{2} \rceil$.

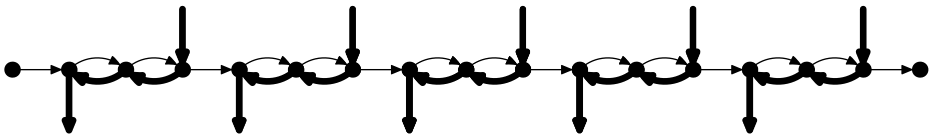


Figure 26. An untraversed edge gadget represents the value 0.

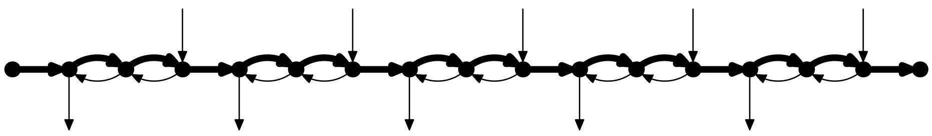


Figure 27. A traversed edge gadget represents the value 1.

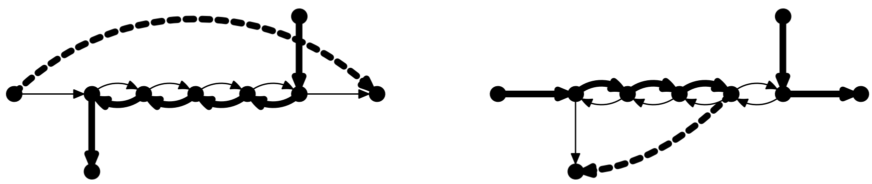


Figure 28. We can assume that traversals shown in the left figure above never occur since they can be transformed into the traversal shown in the right figure without increasing the length of the tour. A bridge with a traversal of that form gives an extra cost of at least $\min\{a + b - 2, a + b/2 - 1\}$ if $B \geq 2a + b - 2$.

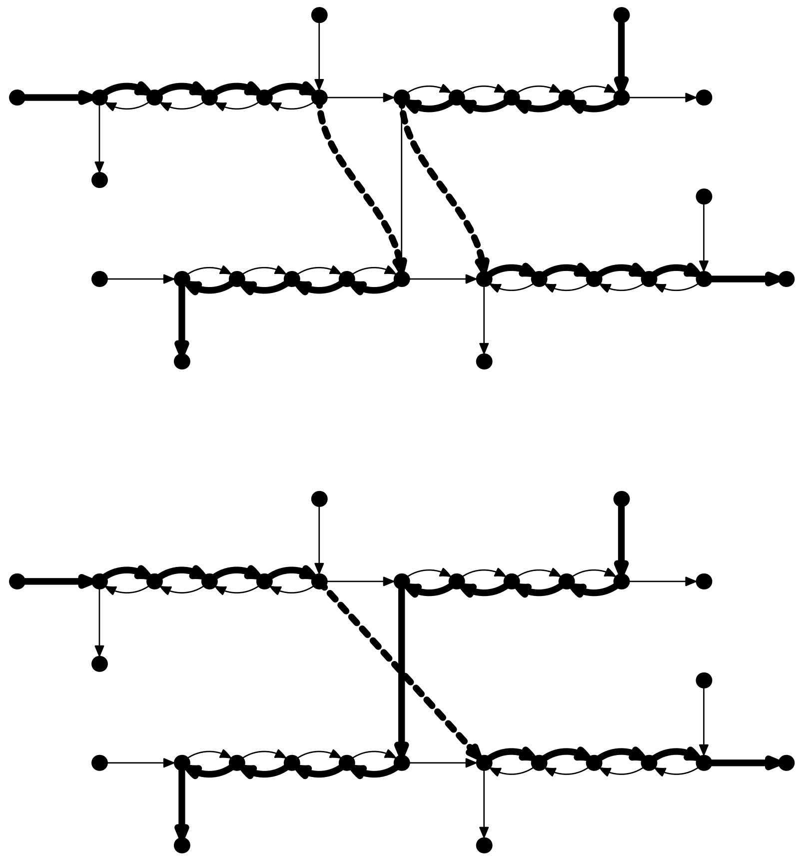


Figure 29. Switching from traversing an edge gadget representing an occurrence of x to traversing another edge gadget representing an occurrence of x gives an extra cost of at least b. The dotted edge above has length $3b$; that gives an extra cost of $2b$ which is then shared evenly among the two semitraversed edge gadgets.

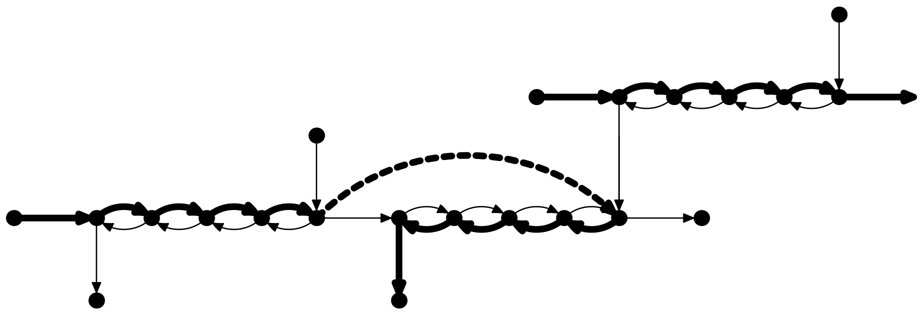


Figure 30. Switching from traversing an edge gadget representing an occurrence of x to traversing an edge gadget representing an occurrence of $\bar{x}$ gives an extra cost of at least $a/2$. The dashed edges above has length $a + b$; that gives an extra cost of a which is then shared evenly among the two semitraversed edge gadgets.

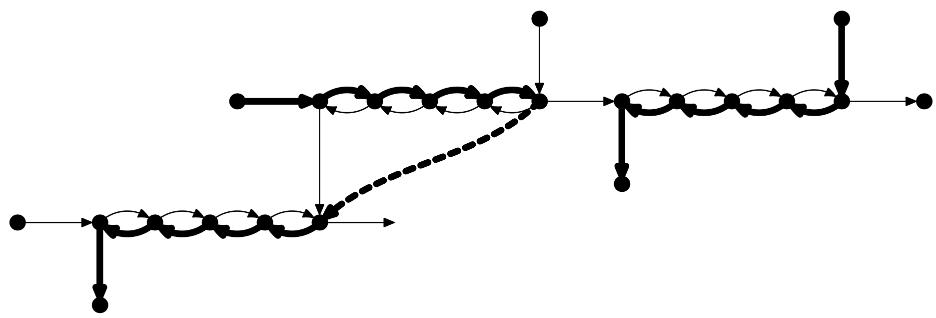


Figure 31. Switching from traversing an edge gadget representing an occurrence of x to traversing an edge gadget representing an occurrence of $\bar{x}$ gives an extra cost of at least $a/2$. The dashed edges above has length $a + b$; that gives an extra cost of a which is then shared evenly among the two semitraversed edge gadgets.

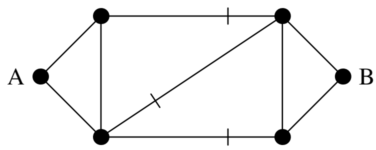


Figure 32. The symmetric gadget for equations of the form $x + y + z = 1$. There is a Hamiltonian path from A to B only if an odd number of the ticked edges are traversed.

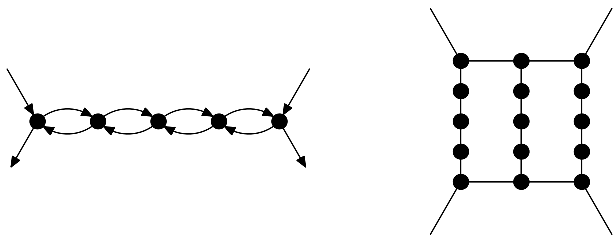


Figure 33. To transform the edge gadget from Fig. 25 into a gadget that can be used in the symmetric case, all occurrences of the structure to the left above are replaced with the structure to the right above. All vertical edges in the right figure have weight 1 and there are *a* edges in each of the three vertical paths; the other edges in the right figure have weight *b*.