

Approximately Fair Cost Allocation in Metric Traveling Salesman Games

M. Bläser · L. Shankar Ram

Published online: 21 December 2007
© Springer Science+Business Media, LLC 2007

Abstract A traveling salesman game is a cooperative game $\mathcal{G} = (N, c_D)$. Here N , the set of players, is the set of cities (or the vertices of the complete graph) and c_D is the characteristic function where D is the underlying cost matrix. For all $S \subseteq N$, define $c_D(S)$ to be the cost of a minimum cost Hamiltonian tour through the vertices of $S \cup \{0\}$ where $0 \notin N$ is called as the *home* city. Define $\text{Core}(\mathcal{G}) = \{x \in \mathfrak{R}^{|N|} : x(N) = c_D(N) \text{ and } \forall S \subseteq N, x(S) \leq c_D(S)\}$ as the core of a traveling salesman game \mathcal{G} . Okamoto (Discrete Appl. Math. 138:349–369, 2004) conjectured that for the traveling salesman game $\mathcal{G} = (N, c_D)$ with D satisfying triangle inequality, the problem of testing whether $\text{Core}(\mathcal{G})$ is empty or not is NP-hard. We prove that this conjecture is true. This result directly implies the NP-hardness for the general case when D is asymmetric. We also study approximately fair cost allocations for these games. For this, we introduce the cycle cover games and show that the core of a cycle cover game is non-empty by finding a fair cost allocation vector in polynomial time. For a traveling salesman game, let $\epsilon\text{-Core}(\mathcal{G}) = \{x \in \mathfrak{R}^{|N|} : x(N) \geq c_D(N) \text{ and } \forall S \subseteq N, x(S) \leq \epsilon \cdot c_D(S)\}$ be an ϵ -approximate core, for a given $\epsilon > 1$. By viewing an approximate fair cost allocation vector for this game as a sum of exact fair cost allocation vectors of several related cycle cover games, we provide a polynomial time algorithm demonstrating the non-emptiness of the $\log_2(|N| - 1)$ -approximate core by exhibiting a vector in this approximate core for the asymmetric traveling salesman game. We improve it further by finding a $(\frac{4}{3} \log_3(|N|) + c)$ -approximate core in

A preliminary version of the paper appeared in the third Workshop on Approximation and Online Algorithms (WAOA), 2005.

M. Bläser

FR Informatik, Universität des Saarlandes, Postfach 151150, 66041 Saarbrücken, Germany
e-mail: mblaeser@cs.uni-sb.de

L. Shankar Ram (✉)

Institut für Theoretische Informatik, ETH Zürich, 8092, Zürich, Switzerland
e-mail: lshankar@inf.ethz.ch

polynomial time for some constant c . We also show that there exists an $\epsilon_0 > 1$ such that it is NP-hard to decide whether ϵ_0 -Core(\mathcal{G}) is empty or not.

Keywords Cooperative games · Fair cost allocations · Traveling salesman game · Approximate fair cost allocation · Combinatorial optimization

1 Introduction

A cooperative game is characterized by a set of players (or agents) and a cost function that is defined for any coalition of these players. The cooperative game related with the traveling salesman problem is very well studied. In a traveling salesman game, the players are the cities which a salesman has to visit. The cost function is intuitively the cost incurred by visiting a given subset of the cities and returning to the home city.

Several problems can be posed with respect to a given combinatorial optimization game. One prominent question is to test the non-emptiness of the core of a game. Probably [18] is the first paper that studied a cooperative game, namely, the assignment game. The underlying combinatorial optimization problem is the assignment problem (or equivalently, the maximum weighted matching problem on bipartite graphs). Testing the core non-emptiness of this game is essentially the same as the polynomial solvability of the optimization problem by the Hungarian method [13]. Another example is the minimum spanning tree game where the core was shown to be non-empty by an explicit construction of a vector in the core [2, 9, 14]. In these examples and some more, a clear relationship exists between the polynomial solvability of the underlying optimization problem and testing the non-emptiness of the core of the game.

Another characterization of the core non-emptiness of a game is from linear programming. A result of Deng et al. [3] states that a necessary and sufficient condition for the core of a maximum packing game and a minimum covering game to be non-empty is that the linear programming relaxations of these problems have integral optimal solutions. Note that the underlying optimization problems in this case are NP-hard. Other characterizations in this direction are for the facility location games [12], partition games [5], and delivery games [10] to mention a few.

On the other hand, several papers deal with the intractability of the core non-emptiness of certain games. For example, Deng et al. [3], showed that testing the non-emptiness of the core of the minimum coloring game is NP-complete. The underlying combinatorial optimization problem in the case is also NP-hard. Thus, this reinstates again the relationship between these two problems. Goemans and Skutella [8] showed the NP-completeness of the core non-emptiness of a facility location game.

In this paper, we study traveling salesman games, introduced by Potters et al. [17]. More formally, a cooperative game is given by the tuple (N, f) , where $N = \{1, 2, \dots, n\}$ and $f : 2^N \rightarrow \Re$ is a characteristic function. In the case of a traveling salesman game, $N = \{1, 2, \dots, n\}$ (the cities) with a given symmetric distance matrix D (referred to as a cost function defined on all pairs of cities) on the set of cities and for a subset $S \subseteq N$, we have the characteristic function $c_D(S)$ defined to be the cost of a minimum cost Hamiltonian tour which visits all the cities in $S \cup \{0\}$

where $0 \notin N$ is called the home city or home node. Note that the cost matrix D is defined over all pairs (i, j) where $i, j \in N \cup \{0\}$. The core of a game (N, f) is defined to be the following:

$$\text{Core}(N, f) = \{x \in \mathfrak{R}^n : x(N) = f(N) \text{ and } \forall S \subseteq N, x(S) \leq f(S)\},$$

where $x(S) = \sum_{i \in S} x(i)$ with $x = (x(i))_{i=1, \dots, n}$. The interpretation of this definition for the traveling salesman game can be motivated as follows. Consider home node 0 as the home city of a professor who has to give talks at the universities located in vertices $1, \dots, n$. The total travel cost is $c_D(N)$. So, the problem is to find a fair cost allocation (a vector in the core) such that no coalition S will split off because they pay more than the actual cost of an optimal subtour through $S \cup \{0\}$ and invite the professor to visit only the universities $i \in S$.

Various aspects of traveling salesman games have already been covered in the literature. Tamir [19] showed that a metric (i.e., satisfying triangle inequality) traveling salesman game with at most four players always has a non-empty core and also the existence of a game with six players whose core is empty. Furthermore, Faigle et al. [6] designed an instance of a 2-dimensional Euclidean game with six players such that the core is empty. More recently, Okamoto [15] showed that the problem of deciding whether a general traveling salesman game has an empty core or not, is NP-hard. But for the special case of metric traveling salesman games, the same question was left open and conjectured to be NP-hard. In this paper, we show that this is indeed the case. In fact, we prove that testing the core-emptiness of a $\{1, 2\}$ traveling salesman game where the costs on any pair of cities is either one or two and the costs are symmetric (i.e., cost on a pair (i, j) is the same as that on (j, i)) is NP-hard. This also proves that it is NP-hard to decide if the core of an asymmetric traveling salesman game with triangle inequality is empty or not. Note that an asymmetric traveling salesman game is a generalization of the symmetric game.

We then consider approximately fair cost allocations, i.e., find a cost allocation vector $x \in \mathfrak{R}^{|N|}$ such that $\forall S \subseteq N, x(S) \leq \epsilon \cdot c_D(S)$ and $x(N) \geq c_D(N)$, for some $\epsilon > 1$. Our reduction also shows that it is NP-hard to find an ϵ_0 -approximate cost allocation vector for some $\epsilon_0 > 1$ for the (a)symmetric traveling salesman game, using a result of Berman et al. [1].

We introduce cycle cover games on the same underlying complete directed graph, where the characteristic function is the cost of a minimum cost cycle cover. Note that a cycle cover game on a complete directed graph is equivalent to an assignment game on a bipartite graph (complete bipartite except for a perfect matching). We show that the core is always non-empty for such a game and provide a $O(|N|^3)$ time algorithm for finding a fair cost allocation vector. This follows easily from known results on assignment games, but we present a proof, too, since we need the notations. We also show that an approximately fair cost allocation vector for an asymmetric traveling salesman game is the sum of exact fair cost allocation vectors of several related cycle cover games.

The question of finding an approximately fair cost allocation vector has already been considered for several cooperative games where testing the core non-emptiness problem is NP-hard. Faigle et al. [6] find a 1.5-approximately fair cost allocation vector for symmetric traveling salesman game. For this, they make use of Christofides'

well-known approximation algorithm for the symmetric traveling salesman optimization problem. In this paper, we provide a polynomial time algorithm that finds a $\log_2(|N| - 1)$ -approximate cost allocation vector for the asymmetric traveling salesman game. We make use of an approximation algorithm for the minimum asymmetric traveling salesman problem (Min ATSP) of Frieze et al. [7]. By adapting a polynomial time $\frac{4}{3} \log_3(n)$ -approximation algorithm for Min ATSP by Kaplan et al. [11], we find a $(\frac{4}{3} \log_3(|N|) + c)$ -approximate cost allocation vector for the asymmetric traveling salesman game for some constant c .

2 Preliminaries

Let $N = \{1, 2, \dots, n\}$. Define $D : (N \cup \{0\}) \times (N \cup \{0\}) \rightarrow \{1, 2\}$ to be an $(n + 1) \times (n + 1)$ symmetric matrix. Let $c_D : 2^N \rightarrow \mathbb{Z}$ be such that $\forall S \subseteq N$,

$$c_D(S) = \min_{\rho: S \rightarrow S} \left\{ d(0, \rho(i_1)) + \sum_{j=1}^{|S|-1} d(\rho(i_j), \rho(i_{j+1})) + d(\rho(i_{|S|}), 0) \right\}$$

over all permutations ρ on $S = \{i_1, i_2, \dots, i_{|S|}\}$. In other words, $c_D(S)$ is the cost of a minimum cost Hamiltonian tour through $S \cup \{0\}$, with $0 \notin N$ called the home node, when we consider the complete graph on $N \cup \{0\}$. The tuple (N, c_D) is the symmetric traveling salesman game. The core of the game is defined as

$$\text{Core}(N, c_D) = \{x \in \mathfrak{R}^n : x(N) = c_D(N) \text{ and } \forall S \subseteq N, x(S) \leq c_D(S)\},$$

where $x(S) = \sum_{i \in S} x(i)$ with $x = (x(i))_{i=1, \dots, n}$. Any vector $x \in \text{Core}(N, c_D)$ is called a *fair cost allocation vector*. Whenever x is a vector, $x(i)$ will refer to the corresponding value at the i th coordinate.

Consider the following decision problem: given a matrix D that fulfills the triangle inequality, is $\text{Core}(N, c_D) = \emptyset$ or not?

We denote the problem as Core- Δ TS or the problem of testing the core non-emptiness of a metric traveling salesman game. A special case of this problem is Core-(1, 2)-TS where the costs on the edges are either one or two. We show that Core- Δ TS is NP-hard by a polynomial time reduction from the following SAT problem (3SAT4) to Core-(1, 2)-TS. 3SAT4 is also called the Bounded Occurrence Satisfiability problem:

Given a boolean formula ϕ as a conjunction of disjunctive clauses with exactly three literals per clause and the number of occurrences of a literal is four, does there exist a truth assignment to the variables of the formula such that all the clauses are satisfied?

3SAT4 was shown to be NP-complete in [20]. Recently, it was shown in [1], that it is NP-hard to approximate the corresponding maximization problem to within a constant $c > 1$.

3 Papadimitriou-Yannakakis' Construction

STSP(1, 2) is the following minimization problem: given a complete graph on nodes N where the costs on the edges are either one or two, find a minimum cost Hamiltonian tour in the graph. We describe the construction of Papadimitrou and Yannakakis [16] to prove the SNP-hardness of STSP(1, 2) from 3SAT4. In the subsequent section, we elaborate our polynomial time reduction from 3SAT4 to Core-(1, 2)-TS, and highlight the differences in both the constructions.

3.1 The Basic Gadgets

The usual reductions to the traveling salesman problem make use of special components called gadgets or devices. A gadget forces an optimal Hamiltonian tour to have a special structure.

A basic gadget used in the construction is the ex-OR device, shown in Fig. 1a. All the edges in this device are of cost one. The structure of the device is so that there can be only two possible traversals of this gadget by any optimum Hamiltonian tour which uses only cost one edges since the gadget is connected to the rest of the graph only at the boundary vertices. We shall think of an ex-OR subgraph as two edges connected by an extrinsic device (Fig. 1b). This will be useful in visualizing the Hamiltonian cycle in the whole graph.

For each variable of the boolean formula, we have a device as shown in Fig. 2. It has two paths, one for each truth value of the variable. We refer to these paths as “true path” and “false path” respectively. Each path is an arrangement of 29 ex-ORs—four

Fig. 1 a The ex-OR gadget. b_1, b_2, b_3, b_4 are called boundary vertices. b Representation of an ex-OR device

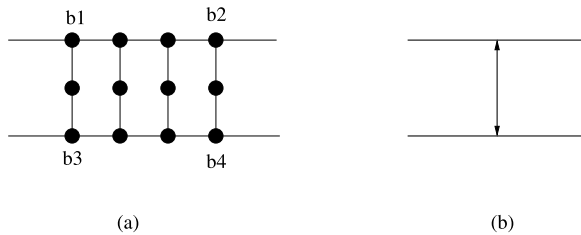


Fig. 2 The variable gadget. There are four “occurrence” edges corresponding to the four occurrences of literal x or \bar{x} , in the respective paths

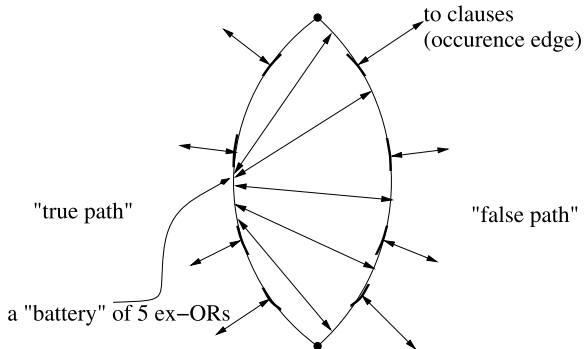


Fig. 3 The clause gadget. There are three ex-OR devices corresponding to the three literals of the clause

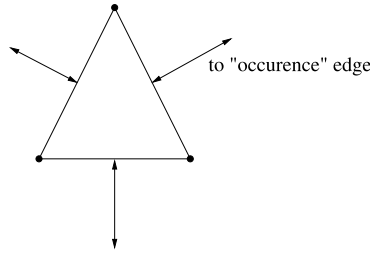
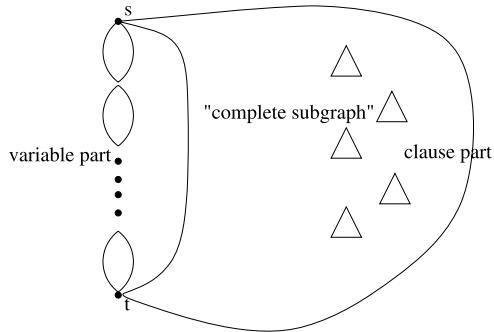


Fig. 4 The graph H . Corners of clause gadgets, and the nodes s and t' form a complete subgraph of H



of them are connected to the clause devices (one for each occurrence of a literal called *occurrence edge*), and the others (five “batteries” or “series” of five ex-ORs each) are connected within to the other path of the same variable device. The intuition behind such a construction is consistency, i.e., to ensure that an optimal tour does not traverse both paths. So, any optimal Hamiltonian tour traverses exactly one of the two paths and also all the vertices of this path appear successively on the tour.

For each clause, we have a triangle device with each edge connected to the occurrence edge of the literal in the clause via an ex-OR device. Please refer to Fig. 3.

3.2 The Construction

We now describe the actual graph that will be constructed from a given boolean formula. Let $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_m$ be the given boolean formula where each $C_i = (a_i \vee b_i \vee c_i)$. Also any variable v appears four times as the literal v and four times as the literal \bar{v} in ϕ . We construct graph the H as follows. Fix an order of the variables and connect the variable gadgets as a series, as shown in Fig. 4. The set of all m clause gadgets are connected so that the $3m$ corners are pairwise connected amongst themselves and also to the first and last vertices of the variable series.

The distance matrix D for this graph H is simply: $d(i, j) = 1$ if $(i, j) \in E$, and otherwise $d(i, j) = 2$. This means that all the edges which are mentioned in the construction are of cost one and the remaining edges (note that an instance of a TSP game is a complete graph) are of cost two.

3.3 Structural Lemmas

Definition 1 Nodes of G that are traversed by an optimal Hamiltonian tour with one edge of cost one and another of cost two are called endpoints. Those nodes that are traversed with both edges of cost two are called double endpoints and will be seen as two endpoints each.

The following two lemmata are shown by Papadimitriou and Yannakakis.

Lemma 1 Let \mathcal{A} be a truth assignment to the variables of ϕ such that maximum number of clauses are satisfied. If, under \mathcal{A} , ϕ has k unsatisfied clauses, then there exists an optimal Hamiltonian tour through the vertices of H with k or $k + 1$ endpoints, depending on k being even or odd respectively. Moreover, these endpoints are present in the clause part of H .

Lemma 2 Let n be the number of vertices in H . If ϕ is satisfiable, the cost of an optimal Hamiltonian tour in H is n . If ϕ is unsatisfiable and there exists an optimal (in the sense of satisfying maximum number of clauses) assignment with k unsatisfied clauses, then the cost of an optimal Hamiltonian tour in H is $n + \lceil \frac{k}{2} \rceil$.

4 Our Construction

Our reduction differs from that of Papadimitriou and Yannakakis [16] in that we reduce to Core-(1, 2)-TS instead of STSP(1, 2). But since the former problem differs only slightly from the latter in structure, we make use of the basic gadgets and the structural lemmas from the previous section.

Our clause gadget is shown in Fig. 5. Note that we have extra nodes as corners and that there are nine edges between the boundary points of adjacent ex-OR devices of the gadget. These edges will be referred to as boundary-boundary edges. (The gadget differs from the one presented in the conference version, the six new edges are needed in the proof of Lemma 3.) This is an important difference from the clause gadget of [16], which will be essential for the NP-hardness proof. The instance graph G of Core-(1, 2)-TS is illustrated in Fig. 6. The modifications are necessitated because of

Fig. 5 The clause gadget. There are three ex-OR devices corresponding to the three literals of the clause. b_3, b_4 vertices are the boundary vertices of the corresponding ex-ORs

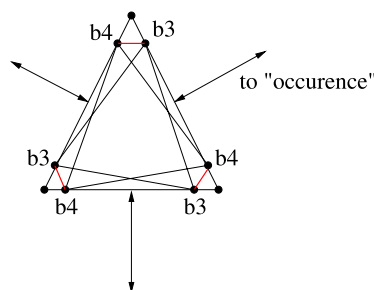


Fig. 6 The graph G . Home node “0” is not considered to be part of G . Corners of clause gadgets, the nodes s and t and “0” form a complete subgraph of $G \cup \{0\}$

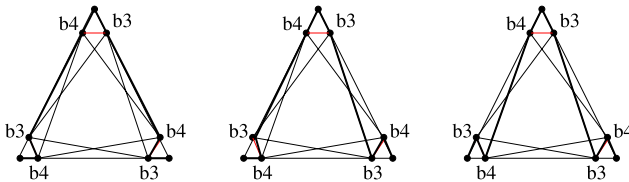
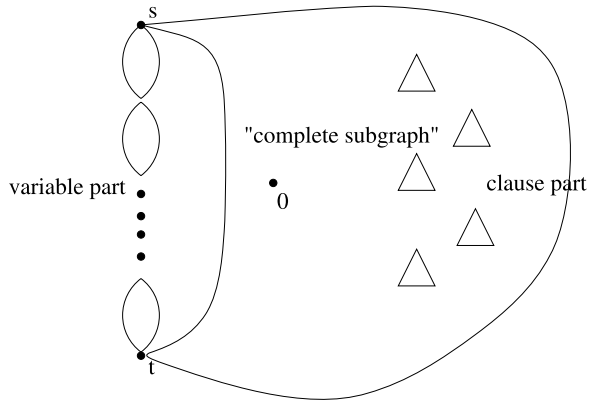


Fig. 7 The traversal of a clause gadget that corresponds to a clause satisfied by one, two, or three literals, respectively

the inclusion of the home node “0”. Note that in the construction, we only show cost one edges. It is assumed that the remaining edges of the complete graph on $N \cup \{0\}$ are of cost two.

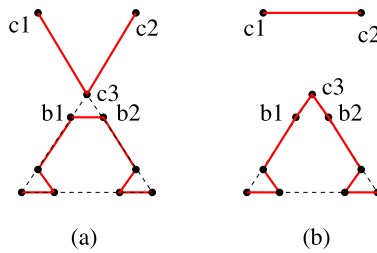
4.1 Structure of an Optimal Tour

We show the following lemmas on the structure of optimal Hamiltonian tours in G .

Lemma 3 *Let \mathcal{A} be a truth assignment to the variables of ϕ such that maximum number of clauses are satisfied. If, under \mathcal{A} , ϕ has k unsatisfied clauses, then there exists an optimal Hamiltonian tour through the vertices of G with k or $k + 1$ endpoints, depending on k being even or odd respectively. Moreover, these endpoints are present in the clause part of G .*

Proof Consider the following tour. The variable part of G is traversed according to the assignment \mathcal{A} , i.e., if $a_i = 1$ in \mathcal{A} , then we take the “true path” of the variable a_i , otherwise the “false path” of the variable. In the clause part, the tour traverses the satisfied clause gadgets first. (At most two edges of such a triangle are not covered by the variable part traversal of the tour). Figure 7 shows how the tour runs through the gadget if one, two, or three clauses are satisfied. Then, in the unsatisfied clause

Fig. 8 **a** shows an optimal Hamiltonian tour using the *boundary-boundary* edge (b_1, b_2) . Such a tour can be modified as shown in **(b)**



gadgets, one endpoint per each gadget is introduced. Since a clause gadget can only be entered through the corner points or the ex-OR gadgets, a clause gadget that corresponds to an unsatisfied clause has to have an endpoint. For parity reasons, one may have to introduce another endpoint. Thus, in this tour there are either k or $k + 1$ endpoints and all of them are introduced in the clause devices. It remains to show that such a tour is optimal. The proof of this fact is essentially the same as given in [16] which exhaustively lists the various possibilities of traversal and in each case how one can modify the tour to have the required structure without increasing costs. The clause gadget of [16] and ours differ in the introduction of new corner vertices additional cost one edges between the boundary points in our construction (the boundary-boundary edges). But externally it behaves like the gadget in [16], since we can enter it only through the corner vertices and the ex-OR-gadgets by cost one edges, so the arguments are valid here, too.

The optimal tours in our construction can enter a clause gadget several times via the corner vertices. We can modify the traversal of any optimal Hamiltonian tour through our clause gadget such that every gadget is only entered once via the corner vertices. This modification is illustrated in Fig. 8. In Fig. 8a, the optimal tour uses the boundary edge (b_1, b_2) . This can be overcome by the modification suggested in (b). Since for optimality, any clause gadget can be entered only from corner vertices, the vertices c_1, c_2 are indeed corner vertices and hence are connected by an edge of cost one. Putting it all together, the lemma follows. \square

Lemma 4 *Let n be the number of vertices in G . If ϕ is satisfiable, the cost of an optimal Hamiltonian tour in G is n . If ϕ is unsatisfiable and there exists an optimal (in the sense of satisfying maximum number of clauses) assignment with k unsatisfied clauses, then the cost of an optimal Hamiltonian tour in G is $n + \lceil \frac{k}{2} \rceil$.*

Proof Consider the optimal Hamiltonian tour constructed in Lemma 3. If ϕ is satisfiable or in other words $k = 0$, then this tour has no endpoints. So, its cost is n . Otherwise, the tour has k or $k + 1$ endpoints. Each endpoint has one edge of cost two in the tour, and hence the number of cost two edges in the tour is $\frac{k}{2}$ or $\frac{k+1}{2}$ when k is even or odd respectively, i.e., $\lceil \frac{k}{2} \rceil$ cost two edges. Thus, the cost of the tour is $(n - \lceil \frac{k}{2} \rceil) \cdot 1 + \lceil \frac{k}{2} \rceil \cdot 2 = n + \lceil \frac{k}{2} \rceil$. \square

5 Hardness Results

Let n be the number of vertices in G . Define $N = \{1, 2, \dots, n\}$, the vertices of G . Also, let $c_D : 2^N \rightarrow \mathfrak{R}$, be defined as follows. For any $S \subseteq N$, $c_D(S)$ is the cost of a minimum cost tour through the vertices of $S \cup \{0\}$. Recall that in our construction, the home node 0 is connected to the corner vertices of all clause gadgets by cost one edges.

Theorem 1 *If $N = \{1, 2, \dots, n\}$ and D is a $(n + 1) \times (n + 1)$ symmetric matrix satisfying triangle inequality, then the problem of deciding if $\text{Core}(N, c_D)$ is empty or not, is NP-hard.*

Proof We show the NP-hardness of $\text{Core-}\Delta\text{TS}$ by showing the following equivalent claim.

Claim *Let ϕ be the boolean formula instance of 3SAT4 given as a conjunction of clauses. Then, ϕ is satisfiable if and only if $\text{Core}(N, c_D)$ is non-empty.*

Suppose ϕ is satisfiable. We show a fair cost allocation in the TSP game (N, c_D) thereby proving the core to be non-empty. Since ϕ is satisfiable, by Lemma 4, the cost of an optimal Hamiltonian tour is $n + 1$. Note that this tour passes through the home node ‘0’. Let us define the vector $x \in \mathfrak{R}^n$ to be $(\frac{n+1}{n}, \frac{n+1}{n}, \dots, \frac{n+1}{n})$. We claim that $x \in \text{Core}(N, c_D)$. Clearly, $x(N) = n + 1 = c_D(N)$. Consider any $S \subseteq N$. We have, $c_D(S) \geq |S| + 1$. But, $x(S) = \sum_{i \in S} \frac{n+1}{n} = (1 + \frac{1}{n})|S| = |S| + \frac{|S|}{n} \leq |S| + 1 \leq c_D(S)$. Hence, $x \in \text{Core}(N, c_D)$.

Now, suppose ϕ is unsatisfiable. Consider an optimal truth assignment \mathcal{A} (optimal in terms of maximum number of satisfiable clauses) which satisfies all but k clauses ($k > 2$). We deal with $k = 1, 2$ cases later. Depending on the truth value of a variable in \mathcal{A} , let T denote all the vertices of G occurring in the ‘‘true paths’’ of all variables (i.e., if variable $v_i = 1$ in \mathcal{A} , then we take the path in the variable device of v_i corresponding to the literal v_i , otherwise that of \bar{v}_i), the vertices of all the ex-OR batteries on the ‘‘false paths’’ and finally all the remaining vertices of satisfied clause devices (with respect to \mathcal{A}). This means that we consider all the vertices in the variable part of G except those present in the occurrence edges on the ‘‘false paths’’ with respect to \mathcal{A} . It also implies that vertices of ex-OR devices of occurrence edges in ‘‘true paths’’ present on the satisfied clauses (with respect to \mathcal{A}) are also in T . Let C denote the corner vertices of the unsatisfied clause gadgets. Let $R := N \setminus \{T \cup C\}$. Suppose, for contradiction, that $\text{Core}(N, c_D) \neq \emptyset$. Let $x \in \text{Core}(N, c_D)$. Since $x \in \mathfrak{R}^n$ is a fair cost allocation vector, the following structural properties must hold for x .

Observation 1 *If x is a fair cost allocation vector, and T is defined as above, then $x(T) \leq |T| + 1$.*

Proof There is a Hamiltonian tour through the vertices of T which uses only cost one edges, as follows. First, traverse the vertices of T on the variable part of G according to the assignment \mathcal{A} . Now, consider the vertices of satisfied clause devices that have not yet been covered by the tour. There are three possibilities:

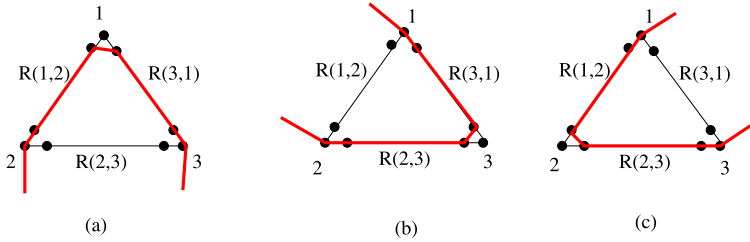


Fig. 9 a, b, c show the traversals of three tours H_1, H_2, H_3 respectively on an unsatisfied clause gadget. In H_1 , the corner vertex 1 and all the vertices of the ex-OR on the edge $R(2, 3)$ are not traversed. The complete tours can be visualized easily since all the corner vertices are pairwise connected

1. Only the corner and boundary vertices of a satisfied clause device are not traversed.
2. An edge of the satisfied clause device and the corner and some boundary vertices are not traversed.
3. Two edges of the satisfied clause device alongwith the corner vertices are not traversed.

Clearly, all these vertices can be traversed as depicted before with cost one edges by recalling the fact that all corner vertices and the home node 0 are interconnected by cost one edges. Hence, $c_D(T) = |T| + 1$. Since $x(T) \leq c_D(T)$, the claim follows. \square

Observation 2 Let C be the set of corner vertices of unsatisfied clause gadgets, and R be the set of remaining vertices in unsatisfied clause gadgets. If x is a fair cost allocation vector, then $x(C) + x(R) \geq |C| + |R| + \lceil \frac{k}{2} \rceil$.

Proof Since x is a fair cost allocation vector, $x(N) = c_D(N)$. Now, there is a Hamiltonian tour through the vertices of G of cost $n + \lceil \frac{k}{2} \rceil$, by Lemma 4. So, $c_D(N) = n + \lceil \frac{k}{2} \rceil + 1$, by recalling the fact that $c_D(N)$ is an optimal tour including the home node 0. Also, $x(N) = x(C) + x(R) + x(T)$, since the sets C, R, T are disjoint. Now, by Lemma 1, $x(T) \leq |T| + 1$, and hence the lemma follows. \square

We consider three tours H_1, H_2, H_3 through the vertices of $C \cup R$ as shown in Fig. 9. Let an unsatisfied clause gadget C_i , for $i = 1, 2, \dots, k$, be given by $(3i - 2, 3i - 1, 3i, R(3i - 2, 3i - 1), R(3i - 1, 3i), R(3i, 3i - 2))$, where the first three are the corner vertices and $R(p, q)$ denotes the vertices in the ex-OR gadget between (p, q) of C_i and the corresponding occurrence edge in the variable part of G . Thus, the tour H_1 is then $\{2, R(1, 2), R(3, 1), 3, 5, R(5, 4), R(6, 4), 6, \dots, 3k - 1, R(3k - 2, 3k - 1), R(3k, 3k - 2), 3k, 2\}$. The tours H_2 and H_3 are similarly defined. Let H be one of these tours with the maximum $x(\cdot)$ value, i.e., $x(H) := \max\{x(H_1), x(H_2), x(H_3)\}$. This implies that $x(H) \geq \frac{1}{3}\{x(H_1) + x(H_2) + x(H_3)\}$.

For all $u \in C \cup R$, $x(u)$ contributes twice in the sum $x(H_1) + x(H_2) + x(H_3)$. Therefore,

$$x(H) \geq \frac{1}{3} \{2x(C) + 2x(R)\} = \frac{2}{3} \left\{ |C| + |R| + \left\lceil \frac{k}{2} \right\rceil \right\}.$$

But, $|H| = |H_1| = |H_2| = |H_3| = \frac{2}{3}\{|C| + |R|\}$ and $x(H) \leq |H| + 1$, by the definition of the tours H_j , i.e., $\frac{2}{3}\{|C| + |R| + \lceil \frac{k}{2} \rceil\} \leq x(H) \leq \frac{2}{3}|C| + \frac{2}{3}|R| + 1$, a contradiction when $k \geq 3$.

We employ the following technique in order to overcome the difficulty in getting a contradiction for $k \leq 2$. Instead of considering the formula ϕ , we look at the formula $\phi' = \phi_1 \wedge \phi_2 \wedge \phi_3$. The formula ϕ' is a conjunction of the formulas ϕ_i , for $i = 1, 2, 3$, where each ϕ_i is a copy of the old formula ϕ but with new, distinct variables. This means that ϕ' has $3n$ variables and $3m$ clauses. It is easy to see that both ϕ and ϕ' are equivalent because the variables of each ϕ_i are distinct. Now, if there is an optimal truth assignment that satisfies all but k clauses of ϕ , then there is an optimal truth assignment that satisfies all but $3k$ clauses of ϕ' . Thus, when $k = 1$ or $k = 2$, the number of unsatisfied clauses in ϕ' is respectively 3 and 6. The above proof holds good for ϕ' . Hence, $\text{Core}(N, c_D) = \emptyset$. This proves the claim that ϕ is satisfiable if and only if $\text{Core}(N, c_D)$ is non-empty. Clearly, the construction of the graph G from ϕ can be done in polynomial time (in the size of ϕ and the number of variables). Therefore, $\text{Core-}\Delta\text{TS}$ is NP-hard. □

Theorem 2 *If $N = \{1, 2, \dots, n\}$ and D is a $(n + 1) \times (n + 1)$ matrix, not necessarily symmetric but satisfying triangle inequality, then the problem of deciding if $\text{Core}(N, c_D)$ is empty or not, is NP-hard.*

Proof Let the problem mentioned in the statement of the theorem be referred to as $\text{Core-}\Delta\text{ATS}$. But since $\text{Core-}\Delta\text{TS}$ is shown to be NP-hard by Theorem 1, and it is a special case of $\text{Core-}\Delta\text{ATS}$, the claim of the theorem follows. □

6 Approximately Fair Cost Allocation

Since the core emptiness problem is NP-hard for traveling salesman games, we turn our attention towards finding approximately fair cost allocation vectors. Define, for a (N, f) game and a given $\epsilon > 1$, the ϵ -approximate core as:

$$\epsilon\text{-Core}(N, f) = \left\{ x \in \Re^{|N|} : x(N) \geq f(N) \text{ and } \forall S \subseteq N, x(S) \leq \epsilon \cdot f(S) \right\}.$$

Towards finding such approximately fair cost allocations, we introduce new games called cycle cover games.

6.1 Cycle Cover Games

Let $G = (V, E)$ be a complete directed graph with a cost function $D : E \rightarrow \Re$. A cycle cover \mathcal{C} in G is a collection of vertex-disjoint cycles that span V . A minimum cost cycle cover is a cycle cover of minimum cost with respect to D . We define a cycle cover game to be the tuple (N, f_D) where $N = V$, and $f_D : 2^V \rightarrow \Re$ is defined for a subset $S \subseteq N$ as the cost of a minimum cost cycle cover on the vertices of S . For this game, we show that the core is non-empty by finding a fair cost allocation vector in polynomial time.

Theorem 3 For a cycle cover game (N, f_D) , the core is not empty. A fair cost allocation vector in the core can be found in $O(|N|^3)$ time.

Proof Consider the following integer program formulation for the minimum cost cycle cover problem:

$$\begin{aligned} \min \quad & \sum_{i,j \in N} d(i, j)y_{ij} \quad \text{subject to} \\ & \sum_{j \in N \setminus \{i\}} y_{ij} = 1 \quad \forall i \in N \quad \text{and} \quad \sum_{i \in N \setminus \{j\}} y_{ij} = 1 \quad \forall j \in N, \\ & \text{where } y_{ij} \in \{0, 1\}. \end{aligned}$$

We relax the final set of constraints to $y_{ij} \geq 0$, to obtain a linear program $\mathcal{L}(N)$. It is known that in fact $\mathcal{L}(N)$ has an integer optimum solution. Next, we consider the dual program of $\mathcal{L}(N)$.

$$\max \sum_{v \in N} x^+(v) + \sum_{v \in N} x^-(v) \quad \text{where } x^+(i) + x^-(j) \leq d(i, j) \quad \forall i, j \in N.$$

Let us denote the dual program by $\mathcal{D}(N)$. Let $x(v) = x^+(v) + x^-(v)$ for all $v \in N$. We claim that an optimal solution $\mathbf{x} = (\mathbf{x}(v))_{v \in N}$ of $\mathcal{D}(N)$ is a fair cost allocation vector to the cycle cover game (N, f_D) . By the duality theorem, we know that the optimal value of the objective function of $\mathcal{L}(N)$ which is $f_D(N)$ by definition, is the same as $\mathbf{x}(N)$. Consider any subset $S \subset N$. Let C_S denote a minimum cost cycle cover on S , i.e., the cost of this cycle cover is $f_D(S)$. Now, $f_D(S) = \sum_{C \in C_S} \sum_{(u,v) \in C} d(u, v) \geq \sum_{C \in C_S} \sum_{(u,v) \in C} (\mathbf{x}^+(u) + \mathbf{x}^-(v)) = \sum_{u \in S} (\mathbf{x}^+(u) + \mathbf{x}^-(u)) = \sum_{u \in S} \mathbf{x}(u) = \mathbf{x}(S)$. Here, $C \in C_S$ denotes a cycle in the cycle cover and the inequality in the middle follows because of the feasibility of \mathbf{x} in $\mathcal{D}(N)$. Thus, we have shown that $\mathbf{x} \in \mathfrak{N}^{|N|}$ is a fair cost allocation vector of the cycle cover game (N, f_D) , thereby showing the non-emptiness of the core. Also \mathbf{x} is computed in $O(|N|^3)$ time using the algorithm of [4] which is a primal-dual type algorithm. □

6.2 A Traveling Salesman Game as a Combination of Several Cycle Cover Games

We show how one can view a traveling salesman game to be a *combination* of several cycle cover games. Formally, what we prove is that a certain approximately fair cost allocation vector for a traveling salesman game can be seen as the sum of (exact) fair cost allocation vectors of several related cycle cover games. We provide an algorithm to find such an approximately fair cost allocation vector, followed by the proof of the claimed degree of approximation.

Here, the traveling salesman game refers to the asymmetric traveling salesman game where the cost matrix fulfills the triangle inequality. For the purpose of proving the main theorem of this section, we adapt the approximation algorithm of Frieze et al. [7], for the asymmetric traveling salesman optimization problem. This approximation algorithm achieves a performance guarantee of $\log_2(|V|)$, where V is the set of vertices of the underlying complete directed graph.

Algorithm 1

Input: An asymmetric game (N, c_D) with a complete directed graph on $N \cup \{0\}$, and D satisfying the triangle inequality.

Output: A vector $\mathbf{x}^* \in \Re^{|N|}$.

- 1: Set $j := 0$, $V_j := N$, and let $\mathbf{x}^* \in \Re^{|N|}$ be the all zero vector.
- 2: Compute a fair cost allocation vector $\mathbf{x}_j \in \Re^{|V_j|+1}$ for the cycle cover game on the complete graph induced on $V_j \cup \{0\}$. Let \mathcal{C} be a minimum cost cycle cover in this graph.
- 3: Set, for all $1 \leq i \leq |N|$, $\mathbf{x}^*(i) := \mathbf{x}_j(i) + \frac{\mathbf{x}_j(0)}{|N|}$ and then set $j := 1$. Let $\mathbf{z}_0 \in \Re^{|N|}$ denote the current \mathbf{x}^* .
- 4: Pick one vertex from each cycle of \mathcal{C} such that the vertex set picked, V_j , does not contain “0”.
- 5: **while** $|V_j| \geq 2$ **do**
- 6: Compute a minimum cost cycle cover \mathcal{C} in the induced complete graph on V_j .
- 7: Compute a fair cost allocation vector $\mathbf{x}_j \in \Re^{|V_j|}$ by using Theorem 3 for the cycle cover game (V_j, c_D) .
- 8: Update $\mathbf{x}^*(i) := \mathbf{x}^*(i) + x_j(i)$ if $i \in V_j$.
- 9: $j := j + 1$.
- 10: Pick exactly one vertex from each of the cycles of the cycle cover \mathcal{C} . Set V_j to be the set of such vertices.
- 11: **end while**
- 12: return the vector \mathbf{x}^* .

The algorithm to find an approximately fair cost allocation vector for an asymmetric traveling salesman game is given in Algorithm 1. Note that the home node “0” is included only in the first cycle cover game.

Theorem 4 *Let (N, c_D) be an asymmetric traveling salesman game, with D satisfying triangle inequality. If \mathbf{x}^* is the vector returned by Algorithm 1 for this game, then it is a $\log_2(|N| - 1)$ -approximately fair cost allocation vector. The running time of the algorithm is $O(|N|^3)$.*

Proof First, let us consider the following linear program for the asymmetric traveling salesman problem, $\mathcal{T}(N)$:

$$\begin{aligned}
 &\min \sum_{i,j \in N \cup \{0\}} d(i, j)y_{ij} \quad \text{subject to} \\
 &\sum_{j \in N \cup \{0\} \setminus \{i\}} y_{ij} = 1 \quad \forall i \in N \quad \text{and} \quad \sum_{i \in N \cup \{0\} \setminus \{j\}} y_{ij} = 1 \quad \forall j \in N, \\
 &\sum_{i \in S, j \in N \cup \{0\} \setminus S} y_{ij} \geq 1 \quad \forall S \subseteq N \quad \text{and} \quad \sum_{j \in S, i \in N \cup \{0\} \setminus S} y_{ij} \geq 1 \quad \forall S \subseteq N, \\
 &\text{where } y_{ij} \geq 0.
 \end{aligned}$$

The third and the fourth set of constraints together are usually referred to as *sub-tour elimination* constraints. Note the inclusion of the home node “0” in the program.

This program is the asymmetric version of the program for symmetric game given in [6]. It can be easily verified that the actual integer linear program corresponding to $\mathcal{T}(N)$ has an optimum value $c_D(N)$. When the subtour elimination constraints are dropped from $\mathcal{T}(N)$, the linear program obtained is the same as the linear program $\mathcal{L}(N \cup \{0\})$ formulated in the proof of Theorem 3. This can be seen as follows: the only issue is to verify that not having the in-degree and out-degree constraints at home node “0” is equivalent to having the constraints. Suppose “0” appears in more than one cycle of an optimal solution y (we can assume that y is integral). Let $u, v \in N$ be such that $y_{u0} = 1 = y_{0v}$ where u, v are in the same cycle in this cycle cover y . Then by changing y_{uv} from 0 to 1 and resetting y_{u0}, y_{0v} both to 0, we obtain a solution of cost at most that of y as $d_{u0} + d_{0v} \geq d_{uv}$ by triangle inequality.

From the algorithm, $\mathbf{x}^*(N) = \sum_{j=0}^k \sum_{i \in N} \mathbf{x}_j(i)$ where k is such that $|V_k| = 1$, i.e. the number of times the while-loop gets executed. By duality theorem, this means that $\mathbf{x}^*(N)$ is the sum of the costs of all k cycle covers computed in the algorithm. Now, the union of all these cycle covers is an Eulerian graph (the in-degree of any vertex is equal to its out-degree). But, any Hamiltonian tour obtained by short-cutting through such an Eulerian graph is of cost at most that of the whole Eulerian graph because of triangle inequality. Hence, $\mathbf{x}^*(N) \geq c_D(N)$ since $c_D(N)$ is the cost of an optimal Hamiltonian tour.

Consider any subset $R \subset N$. We claim that $\mathbf{x}^*(R) \leq \log_2(|R|)c_D(R)$. By definition, $\mathbf{x}^*(R) = \mathbf{z}_0(R) + \sum_{j=1}^k \mathbf{x}_j(R \cap V_j)$. First, we show that $\mathbf{z}_0(R) \leq c_D(R)$. Now, since $\mathbf{x}_0 \in \mathfrak{R}^{|N|+1}$ (refer to step 2 of the algorithm) is an exact fair cost allocation vector for the cycle cover game on $N \cup \{0\}$, we have $\mathbf{x}_0(R \cup \{0\})$ is at most the cost of a minimum cost cycle cover on $R \cup \{0\}$. But, by definition, $\mathbf{x}_0(R \cup \{0\}) = \mathbf{x}_0(R) + \mathbf{x}_0(0) = \{\mathbf{x}_0(R) + |R| \frac{\mathbf{x}_0(0)}{|N|}\} + (|N| - |R|) \frac{\mathbf{x}_0(0)}{|N|} \geq \mathbf{z}_0(R)$ since $|N| - |R| > 0$. Thus, $\mathbf{z}_0(R)$ is at most the cost of a minimum cost cycle cover on $R \cup \{0\}$ which is at most $c_D(R)$, the cost of an optimal Hamiltonian tour through $R \cup \{0\}$. Next, we show that for all $0 < j \leq k$, $\mathbf{x}_j(R \cap V_j) \leq c_D(R)$. Denote by TSP_j , the optimal value of the linear program $\mathcal{T}(V_j \cap R)$. Then, since any feasible solution to $\mathcal{T}(V_j \cap R)$ is a cycle cover on $V_j \cap R$, we have that $\sum_{j=1}^k \mathbf{x}_j(R \cap V_j)$ is bounded by $\sum_{j=1}^k TSP_j$. The only non-zero TSP_j values are those for which $|V_j \cap R| \neq 0$. By triangle inequality, we have that for all j , $TSP_j \leq TSP_0$ where TSP_0 is the cost of an optimal solution to the LP, $\mathcal{T}(N \cap R)$. As shown before, $\mathbf{z}_0(R) \leq c_D(R)$. Hence, $\mathbf{x}^*(R) \leq \mathbf{z}_0(R) + \sum_{j \geq 1: |V_j \cap R| \neq 0} TSP_j \leq c_D(R) + (\log_2(|R|) - 1)TSP_0 \leq \log_2(|R|)c_D(R)$. The last inequality is true because the linear program optimal value is a lower bound on the integer optimal value. Since, $R \subset N$, $|R| \leq |N| - 1$. So, for any $R \subset N$, we have $\mathbf{x}^*(R) \leq \log_2(|N| - 1)c_D(R)$.

From the above two paragraphs, we deduce that $\mathbf{x}^* \in \mathfrak{R}^{|N|}$ is a $\log_2(|N| - 1)$ -approximately fair cost allocation vector for the asymmetric traveling salesman game (N, c_D) .

As for the running time of the algorithm, to find a minimum cost cycle cover there is $O(|N|^3)$ algorithm due to [4]. Also, as mentioned in Theorem 3, finding a fair cost allocation vector for a cycle cover game takes $O(|N|^3)$ time. The while-loop is executed at most $\log_2(|N|) - 1$ times, where $|V_{j+1}| \leq |V_j|/2$ with $j = 0, 1, \dots, k$ where $|V_k| = 1$ and $V_0 = N$. Thus the total running time of the algorithm is $O(\sum_{i=0}^k (|N|/2^i)^3) = O(|N|^3)$. □

6.3 Improved Approximately Fair Cost Allocation

In this section, we try to adapt the $\frac{4}{3} \log_3(n)$ -approximation algorithm for the minimum asymmetric traveling salesman problem of Kaplan et al. [11] to find a better approximately fair cost allocation for the asymmetric traveling salesman game.

Lemma 5 [11] *Let $G = (V, E)$ be a complete directed graph with a cost matrix D satisfying triangle inequality. There exists a 2-regular directed graph C (obtained in polynomial time in G) such that C is the union of two directed cycle covers C_1, C_2 where the cost of C is at most $2 \cdot OPT$, OPT is the cost of an optimal Hamiltonian tour and C_1, C_2 do not share a 2-cycle.*

We adapt the above lemma to a statement concerning vectors of cycle covers.

Lemma 6 *Let (N, c_D) be an asymmetric traveling salesman game where D satisfies triangle inequality. Also let (N, f_D) be the cycle cover game on the same matrix D . There exist vectors $z_1, z_2 \in \mathfrak{R}^{|N|}$ corresponding to cycle covers C_1, C_2 of Lemma 5 such that there is a vector $z_3 \in \mathfrak{R}^{|N|}$ where $z_3 = z_1 + z_2$ that satisfies the properties:*

$$\begin{aligned} z_3(N) &\leq 2 \cdot f_D(N) \leq 2 \cdot c_D(N), \\ \forall R \subset N, \quad z_3(R) &\leq 2 \cdot f_D(R) \leq 2 \cdot c_D(R). \end{aligned}$$

These vectors can be found in polynomial time.

Proof Follows directly from Lemma 5. □

We introduce some notation for understanding the algorithm that finds an improved approximately fair cost allocation vector for the asymmetric traveling salesman game. We assume that we are given a directed graph $G = (V, E)$ with a cost matrix D . For a directed subgraph $H \subseteq G$, let $d(H) := \sum_{[u,v] \in H} d(u, v)$. Also, let $c(H)$ denote the number of connected components of H and $N(H)$ denote the number of vertices in H . Define $\delta_{i,H} := 1$ if the vertex $i \in H$ and zero otherwise.

Theorem 5 *Let (N, c_D) be an asymmetric traveling salesman game, with D satisfying triangle inequality. If \mathbf{x}^* is the vector returned by Algorithm 2 for this game, then it is a $(\frac{4}{3} \log_3(|N|) + c)$ -approximately fair cost allocation vector for some constant c . The running time of the algorithm is $O(|N|^2 \log_2(|N| d_{\max}) \log_2(|N|))$. Here d_{\max} is the cost of a maximum cost edge in the complete graph on N .*

Proof We need to show that \mathbf{x}^* satisfies the following two properties:

$$\mathbf{x}^*(N) \geq c_D(N), \tag{1}$$

$$\forall R \subset N, \quad \mathbf{x}^*(R) \leq \left(\frac{4}{3} \log_3(|N|) + c \right) \cdot c_D(R). \tag{2}$$

Algorithm 2

Input: An asymmetric game (N, c_D) with a complete directed graph on $N \cup \{0\}$, and D satisfying triangle inequality.

Output: A vector $\mathbf{x}^* \in \mathfrak{R}^{|N|}$.

- 1: Set $p := 1$. Let G_p be the induced complete graph on $N \cup \{0\}$.
- 2: Let $\mathbf{x}^* \in \mathfrak{R}^{|N|}$ be the all-zero vector.
- 3: **while** $|N(G_p)| \geq 27$ **do**
- 4: Compute the two vectors $z_1, z_2 \in \mathfrak{R}^{|N(G_p)|}$ corresponding to the cycle covers C_1, C_2 from Lemma 6. Let C be the union of $C_1 \cup C_2$. Also, $z_3 := z_1 + z_2$.
- 5: Set $C_1^p := C_1, C_2^p := C_2$ and $C^p := C$.
- 6: Amongst the directed graphs C^p, C_1^p, C_2^p choose the graph G that maximizes

$$\frac{d(G)}{\log_2(N(G)/c(G))}$$
- 7: Let $z_{j_p} \in \{z_1, z_2, z_3\}$ (for some $j_p \in \{1, 2, 3\}$) be the corresponding vector.
- 8: **if** $p = 1$ **then**
- 9: Set $z'_{j_p}(i) := z_{j_p}(i) + \frac{z_{j_p}(0)}{|N|}$ for all $i = 1, \dots, |N|$. Note that $z_{j_p} \in \mathfrak{R}^{|N|+1}$.
- 10: $\mathbf{x}^*(i) := \mathbf{x}^*(i) + z'_{j_p}(i)$.
- 11: From each connected component of G , pick exactly one vertex arbitrarily. The home node “0” should not be picked.
- 12: **else**
- 13: $\mathbf{x}^*(i) := \mathbf{x}^*(i) + \delta_{i,G} \cdot z_{j_p}(i)$.
- 14: From each connected component of G , pick exactly one vertex arbitrarily.
- 15: **end if**
- 16: Set $p := p + 1$. Set G_p to be the induced complete graph on the chosen vertices.
- 17: **end while**
- 18: Compute $c_D(N(G_p))$ by exhaustive search since $|N(G_p)| < 27$. Let $v \in \mathfrak{R}^{|N(G_p)|}$ where $v(i) = \frac{c_D(N(G_p))}{|N(G_p)|}$ for all $i \in G_p$.
- 19: $\mathbf{x}^*(i) := \mathbf{x}^*(i) + v(i)$ if $i \in G_p$.
- 20: return the vector \mathbf{x}^* .

$\mathbf{x}^*(N)$ is the sum of the costs of all the graphs G_p computed in the algorithm. Now the union of all these graphs is an Eulerian graph (since C^p is Eulerian too). But, any Hamiltonian tour obtained by short-cutting through such an Eulerian graph is of cost at most that of the whole Eulerian graph by triangle inequality. Therefore, $\mathbf{x}^*(N) \geq c_D(N)$.

To prove the second property, we use the structural lemmas from Kaplan et al. [11]. Let $R \subset N$. We have,

$$\gamma := \frac{z'_{j_1}(R) + \sum_{t=2}^{p-2} z_{j_t}(R)}{\log_2(|N(G_1)|/|N(G_{p-1})|)} = \frac{z'_{j_1}(R) + \sum_{t=2}^{p-2} z_{j_t}(R)}{\sum_{t=1}^{p-2} \log_2(|N(G_t)|/|N(G_{t+1})|)}.$$

Consider $z'_{j_1}(R)$. By definition, $z'_{j_1}(R) = z_{j_1}(R) + \frac{|R|}{|N|} \cdot z_{j_1}(0)$. But since $z_{j_1} \in \mathfrak{N}^{|N|+1}$ satisfies the properties of Lemma 6, we have that $z_{j_1}(R \cup \{0\}) = z_{j_1}(R) + z_{j_1}(0) \leq 2f_D(R) \leq 2c_D(R)$. Therefore, we can extend the above equality by,

$$\gamma \leq \max \left\{ \frac{z'_{j_1}(R)}{\log_2(|N(G_1)|/|N(G_2)|)}, \max_{t=2, \dots, p-2} \left\{ \frac{z_{j_t}(R)}{\log_2(|N(G_t)|/|N(G_{t+1})|)} \right\} \right\}.$$

Following the arguments given in [11], we obtain that the quantity on the right side of the above inequality is bounded by $\frac{4c_D(R)}{3 \log_2(3)}$. Thus, for $|N(G_p)| \geq 27$,

$$\begin{aligned} \mathbf{x}^*(R) &= z'_{j_1}(R) + \sum_{t=2}^p z_{j_t}(R) \leq z'_{j_1}(R) + \sum_{t=2}^{p-2} z_{j_t}(R) + 3c_D(R) \\ &\leq \frac{4c_D(R)}{3 \log_2(3)} \log_2(|N|) = \frac{4}{3} \log_3(|N|) \cdot c_D(R) \end{aligned}$$

since if $p \leq 2$, $z'_{j_1}(R) + \sum_{t=2}^p z_{j_t}(R) \leq 3c_D(R)$. Finally, when $|N(G_p)| < 27$, we add the vector $v \in \mathfrak{N}^{|N(G_p)|}$ to \mathbf{x}^* where $v(i) = \frac{c_D(N(G_p))}{|N(G_p)|}$ for all $i \in G_p$. We claim that v is a c -approximately fair cost allocation vector for some small constant c . It is easy to verify the claim by fixing $c = \frac{|R|}{|N(G_p)|} \cdot \frac{c_D(N(G_p))}{\min_{R \subset N(G_p)} \{c_D(R)\}}$. Therefore, \mathbf{x}^* is a $(\frac{4}{3} \log_3(|N|) + c)$ -approximate fair cost allocation vector for the asymmetric traveling salesman game. The running time of the algorithm is dominated by the time taken to compute the cycle covers of step 4 which is $O(|N|^2 \log_2(|N|d_{\max}))$, see [11]. The number of iterations p until $|N(G_p)| < 27$ is bounded by $\log_2(|N|)$. \square

6.4 Inapproximability of the Core

Our NP-hardness reduction is approximation preserving (please refer to Vazirani [21] for a definition). Thus, we have the following result.

Theorem 6 *Let (N, c_D) be an asymmetric traveling salesman game, with D satisfying triangle inequality. There exists an $\epsilon_0 > 1$ such that it is NP-hard to decide whether ϵ_0 -Core (N, c_D) is empty or not. In other words, it is NP-hard to find an ϵ_0 -approximately fair cost allocation vector for the game.*

Proof Since it is NP-hard to approximate 3SAT4 to within a certain constant $\alpha > 1$ [1] and the reduction in from the proof of Theorem 2 is approximation preserving, the result follows. \square

References

1. Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness and satisfiability of bounded occurrence instances of SAT. Technical Report TR03-022, ECCS (2003)
2. Bird, C.G.: On cost allocation for a spanning tree: a game theoretic approach. *Networks* **6**, 335–350 (1976)

3. Deng, X., Ibaraki, T., Nagamochi, H.: Algorithmic aspects of the core of combinatorial optimization games. *Math. Oper. Res.* **24**, 751–766 (1999)
4. Edmonds, J., Johnson, L.: Matching: a well-solved class of integer linear programs. In: Proceedings of Calgary International Conference on Combinatorial Structures and their Applications, pp. 89–92. Gordon and Breach, New York (1970)
5. Faigle, U., Kern, W.: On the core of ordered submodular cost games. *Math. Program.* **87**, 483–499 (2000)
6. Faigle, U., Fekete, S.P., Hochstättler, W., Kern, W.: On approximately fair cost allocation in Euclidean TSP games. *OR Spektrum* **20**, 29–37 (1998)
7. Frieze, A., Galbiati, G., Maffioli, F.: On the worst-case performance of some algorithms for the asymmetric travelling salesman problem. *Networks* **12**, 23–39 (1982)
8. Goemans, M.X., Skutella, M.: Cooperative facility location games. In: Proc. 11th SODA, pp. 76–85 (2000)
9. Granot, D., Huberman, G.: Minimum cost spanning tree games. *Math. Program.* **21**, 1–18 (1981)
10. Granot, D., Hamers, H., Tijs, S.: On some balanced, totally balanced and submodular delivery games. *Math. Program.* **86**, 355–366 (1999)
11. Kaplan, H., Lewenstein, M., Shafrir, N., Sviridenko, M.: Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *J. ACM* **52**(4), 602–626 (2005)
12. Kolen, A.: Solving covering problems and the uncapacitated plant location algorithms. *Eur. J. Oper. Res.* **12**, 266–278 (1983)
13. Kuhn, H.W.: The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**, 83–97 (1955)
14. Megiddo, N.: Computational complexity and the game theory approach to cost allocation for a tree. *Math. Oper. Res.* **3**, 189–196 (1978)
15. Okamoto, Y.: Traveling salesman games with the Monge property. *Discrete Appl. Math.* **138**, 349–369 (2004)
16. Papadimitriou, C., Yannakakis, M.: The traveling salesman problem with distances one and two. *Math. Oper. Res.* **18**, 1–11 (1993)
17. Potters, J.A.M., Curiel, I.J., Tijs, S.H.: Traveling salesman games. *Math. Program.* **53**, 199–211 (1992)
18. Shapley, L., Shubik, M.: The assignment game I: the core. *Int. J. Game Theory* **1**, 111–130 (1972)
19. Tamir, A.: On the core of a traveling salesman cost allocation game. *Oper. Res. Lett.* **8**, 31–34 (1988)
20. Tovey, C.A.: A simplified NP-complete satisfiability problem. *Discrete Appl. Math.* **8**, 85–89 (1984)
21. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2001)