Nash fairness solutions for balanced TSP

Minh Hieu Nguyen, Mourad Baiou, Viet Hung Nguyen*, Thi Quynh Trang Vo
INP Clermont Auvergne, Univ Clermont Auvergne, Mines Saint-Etienne, CNRS, UMR 6158 LIMOS,
1 Rue de la Chebarde, Aubiere Cedex, France
*Corresponding author : Viet Hung Nguyen
minh_hieu.nguyen@uca.fr,mourad.baiou@uca.fr,viet_hung.nguyen@uca.fr,thi_quynh_trang.vo@uca.fr

ABSTRACT

In this paper, we consider a variant of the Traveling Salesman Problem (TSP), called *Balanced Traveling Salesman Problem* (BTSP) [7]. The BTSP seeks to find a tour which has the smallest *max-min distance* : the difference between the maximum edge cost and the minimum one. We present a Mixed Integer Program (MIP) to find optimal solutions minimizing the max-min distance for BTSP. However, minimizing only the max-min distance may lead to a tour with an inefficient total cost in many situations. Hence, we propose a fair way based on Nash equilibrium [5], [11] to inject the total cost into the objective function of the BTSP. We consider a Nash equilibrium as it is defined in a context of fair competition based on proportional-fair scheduling. For BTSP, we are interested in solutions achieving a Nash equilibrium between two players: the first aims at minimizing the total cost and the second aims at minimizing the max-min distance. We call such solutions *Nash Fairness* (NF) solutions. We first show that NF solutions for BTSP exist and may be more than one. We show that NF solutions are Pareto-optimal [10] and can be found by optimizing a sequence of linear combinations of the two players objectives based on Weighted Sum Method [13]. We then focus on extreme NF solutions which are NF solutions having either the smallest value of total cost or the smallest max-min distance. Finally, we propose a Newton-based iterative algorithm which converges to extreme NF solutions in a polynomial number of iterations. Computational results on small-size instances from TSPLIB will be presented and commented.

1 INTRODUCTION

The Balanced Traveling Salesman Problem (BTSP) is a variation of the classical Traveling Salesman Problem (TSP) where instead of finding a Hamiltonian tour minimizing the total cost, we find a tour minimizing *the max-min distance*. The latter is the difference between the maximum edge cost and the minimum one in the tour. BTSP has been introduced by Larusic and Punnen (2011) [7] for finding Hamiltonian tours in several cases where the equitable distribution of edges are important, for example, the nozzle guide vane assembly problem [12] and the cyclic workforce scheduling problem [15].

BTSP can be formally defined as follows. Given an undirected graph $G = (V, E)$ where $V = [n] := \{1, \dots, n\}$, $|E| = m$, $c_{ij} \in \mathbb{R}_+$ is a cost associated with every edge $ij \in E$ and let $\Pi(G)$ denote the set of all Hamiltonian cycles in G, BTSP can be defined as

$$\min_{H \in \Pi(G)} \left\{ \max_{ij \in H} c_{ij} - \min_{ij \in H} c_{ij} \right\}. \quad (1)$$

BTSP is NP-hard as the problem of finding a Hamiltonian cycle in G can easily be reduced to it. In [7], the authors proposed four thresholds heuristic algorithms to solve this problem. More precisely, distinct elements of c are firstly sorted in the ascending direction, i.e. $z_1 < \dots < z_p$. The proposed algorithms find a pair (z_i, z_j) satisfying (i) a subgraph of G with the edge set $\{(i, j) \in E | z_i \leq c_{i,j} \leq z_j\}$ is Hamiltonian, and (ii) $z_i - z_j$ is as small as possible. The existence of Hamiltonian cycles in subgraphs is verified by necessary conditions for a tour and heuristic procedures instead of exact algorithms that are highly computationally expensive. As a consequence, the optimality of obtained solutions is not certified. Moreover, by minimizing only the max-min distance, the total cost of edges used in the tour is neglected and it may lead to very inefficient tours in many situations.

Another work relating to the fairness of a tour in TSP is the equitable TSP proposed by Kindable et al. [6]. In the equitable TSP, one tends to minimize the absolute difference between the total cost of two perfect matchings, which form a Hamiltonian cycle in a graph. The problem is presented with the example about the uniform distribution of distances to pedal for two people. Two integer programming formulations are proposed to solve the equitable TSP exactly. However, this problem also may not guarantee the efficiency of a tour in terms of the total cost.

In this paper, to overcome the possible inefficiency of the solutions for BTSP, we propose a fair way based on Nash equilibrium to inject the total cost into the objective function of BTSP. Nash equilibrium is the most common optimality notion for sharing resources among users [5],[11]. We consider a Nash equilibrium to be fair as it is defined in a context of fair competition based on proportional-fair scheduling that aims to provide a compromise between the utilitarian rule - which emphasizes overall system efficiency, and the egalitarian rule - which emphasizes individual fairness. For BTSP, we are interested in solutions achieving a Nash equilibrium between two players: the first aims at minimizing the total cost and the second aims at minimizing the max-min distance. We call such solutions *Nash Fairness* (NF) solutions. We first show that NF solutions for BTSP exist and may be more than one. We show that NF solutions are Pareto-optimal [10] and can be found by optimizing a sequence of linear combinations of the two players objectives based on Weighted Sum Method [13]. We then focus on extreme NF solutions which are NF solutions having either the smallest value of total cost or the smallest max-min distance. Finally, we propose a Newton-based iterative algorithm which converges to extreme NF solutions in a polynomial number of iterations. Computational results on small size instances from TSPLIB will be presented and commented.

The paper is organized as follows. In Section 2, we present a mathematical formulation for BTSP. The notion of Nash fairness solution will be discussed in Section 3. In particular, we prove the existence of NF solutions for BTSP and show that they are optimal solutions of a weighted sum objective problem. In Section 4, Newton-based iterative algorithms for finding extreme NF solutions is given. Computational results on small size instances from TSPLIB will be presented and discussed in Section 5.

2 MIP FORMULATION FOR BTSP

Although several heuristic algorithms [7] have been developed for this problem, there is no exact formulation for the BTSP mentioned in the literature to the best of our knowledge. To formulate a MIP for BTSP, we first consider the directed version of G by replacing every edge ij by two arcs (i, j) and (j, i). The costs $c_{i,j}$ and $c_{j,i}$ associated respectively with (i, j) and (j, i) are both equal to c_{ij}. A Hamiltonian tour in the original undirected graph G correspond now to a directed tour in the directed version. We propose a MIP formulation (for complete graph G) for the BTSP as follows:

$$\min t \tag{2a}$$

$$\text{s.t.} \sum_{j\in[n]} x_{j,i} = 1 \qquad \forall i \in [n] \tag{2b}$$

$$\sum_{j\in[n]} x_{i,j} = 1 \qquad \forall i \in [n] \tag{2c}$$

$$\sum_{i\in Q} \sum_{j\neq i, j\in Q} x_{i,j} \leq |Q| - 1 \qquad \forall Q \subset V \tag{2d}$$

$$t \geq u - l \tag{2e}$$

$$u \geq c_{i,j} x_{i,j} \qquad \forall i, j \in [n] \tag{2f}$$

$$l \leq \sum_{j\in[n]} c_{i,j} x_{i,j} \qquad \forall i \in [n] \tag{2g}$$

$$x_{i,j} \in \{0, 1\} \qquad \forall i, j \in [n]. \tag{2h}$$

where $x_{i,j}$ is the binary variables representing the occurrence of arc (i, j) in the solution tour. The constraints (2b), (2c) are respectively the in-degree and out-degree constraints which assure that there is exactly one incoming arc and one outgoing arc incident to every vertex. The constraints (2d) are the subtour elimination constraints. These constraints represent the classical Held-Karp (linear programming) relaxation for the Asymmetric Traveling Salesman Problem. Together with the integral constraints (2h), they assure that the solution is a tour. In order to calculate the max-min distance t, we need to determine the largest and the smallest edge costs u and l in the solution tour. Constraints (2f) obviously allow to bound u from below by the largest weight arc in the solution tour. There will be exactly one non null term in the sum in the right hand side of constraints (2g) as there is exactly one arc leaving each vertex i. This non null term represents the weight of the arc leaving i in the solution tour. Hence, constraints (2g) allow to bound l from above by the smallest arc weight in the solution tour. As t is minimized, u and t will respectively take the values of the largest and the smallest edge costs.

Table 1: The BTSP results in TSPLIB instances

Instance	Heuristic algorithms [7]	Formulation (2)
att48	192	**190**
gr48	48	**46**
berlin52	151	**149**
brazil58	1125	**1097**

We have designed a special-purpose branch-and-cut algorithm based on Formulation (2). Despite of the simplicity of the latter, the algorithm is capable to find an optimal solution for instances of BTSP up to 80 vertices within 1 hour CPU time. Our experiments on instances from TSPLIB have been able to certificate the optimality of solutions found by heuristic algorithms in [7]. Furthermore, as shown in Table 1, our exact algorithm have disproved the optimality of the solution given in [7] for several instances.

However, the purpose of this paper is not to design exact solutions for BTSP and to compare with the results in [7]. Since optimal solutions for BTSP may not be very efficient in terms of the total cost, the purpose of the paper is to inject the latter into the objective function in some fair ways allowing a trade-off between the two objectives. The rest of the paper will be devoted to this question.

3 NASH FAIRNESS SOLUTIONS FOR BTSP

3.1 Characterization of NF solutions

Nash fairness (NF) solutions for maximizing the utilities of two-player problem [5] are defined by using the Nash standard of comparison. Under the latter, a transfer of utilities between the two players is considered to be fair if the percentage increase in the utility of one player is larger than the percentage decrease in utility of the other player [1].

Proportional fairness is a generalized NF solution for multiple players. In that setting, the fair allocation should be such that, if compared to any other feasible allocation of utilities, the aggregate proportional change is less than or equal to 0 [5], [1], [11].

Let U be a set of possible *states of the world* or *alternatives* and let I be a finite set, representing a collection of individuals. For each $i \in I$, $u_i : U \longrightarrow \mathbb{R}_+$ be a utility function, describing the amount of happiness an individual i derives from each possible state such that we prefer the alternative x to the alternative y if and only if $u_i(x) \geq u_i(y)$.

Definition 3.1. [1] $x^{NF} \in U$ be a NF solution in multiple players problem if and only if

$$\sum_{j=1}^{n} \frac{u_j(x) - u_j(x^{NF})}{u_j(x^{NF})} \leq 0, \ \forall x \in U, \tag{3}$$

where $u_j(x) > 0, \ \forall j \in I, \forall x \in U$.

Let P, Q represent the total cost and the max-min distance in a feasible solution tour for BTSP. We have then $P > Q \geq 0$. We first suppose that $Q > 0$. As P, Q now are two positive utility functions, we have a two-player problem. In the usual definition of NF solutions [5], [1], the alternative assigned a greater value is preferred. However, in BTSP, we prefer the alternative assigned a smaller value for two utility functions P and Q. Thus, we need to modify the sign of each term representing the proportional change in Definition 3.1.

In the remainder of this paper, we consider (P, Q) as the solution for the total cost and the max-min distance corresponding to a feasible solution tour. Let (P^*, Q^*) be a NF solution for BTSP, condition (3) can be translated into the context of BTSP as follows

$$\frac{P^* - P}{P^*} + \frac{Q^* - Q}{Q^*} \leq 0, \ \forall (P, Q) \in S, \tag{4}$$

which is equivalent to

$$PQ^* + QP^* \geq 2P^*Q^*, \ \forall (P, Q) \in S, \tag{5}$$

where S is the set of solutions (P, Q) corresponding to all feasible solution tours for BTSP in G.

We note that in case $Q^* = 0$, the condition (5) is also satisfied. Hence, NF solution for BTSP can be generally stated as follows

Lemma 3.2. *$(P^*, Q^*) \in S$ be a NF solution for BTSP if and only if $PQ^* + QP^* \geq 2P^*Q^*, \ \forall (P, Q) \in S$.*

Remark 3.3. $(P, 0)$ is always a NF solution (i.e a solution tour with all equal edge costs).

3.2 Existence of NF solutions

In this section, we first show the existence of NF solutions for BTSP. Let us recall that in the classical multiple players problem mentioned in Section 3.1 where we prefer the alternative assigned a greater value, the NF solution can be obtained with the objective function

$$\max \sum_{j=1}^{n} \log \ u_j,$$

provided that U is convex. The necessary and sufficient first-order optimality condition for this problem is exactly the Nash standard of comparison principle for n players. Notice that the above NF solution is the one maximizing the product of the utilities over U.

On the contrary in BTSP, we prefer the alternative assigned a smaller value for two utility functions P, Q. Thus, there exists a NF solution which can be obtained by minimizing instead of maximizing the product of the utilities.

Theorem 3.4. *$(P^*, Q^*) = \text{argmin}_{(P,Q) \in S} PQ$ is a NF solution.*

Proof. Obviously, there always exists a solution $(P^*, Q^*) \in S$ such that

$$(P^*, Q^*) = \underset{(P,Q) \in S}{\text{argmin}} \, PQ.$$

Now $\forall (P', Q') \in S$ we have $P'Q' \geq P^*Q^*$. Then

$$P'Q^* + Q'P^* \geq 2\sqrt{P'Q'P^*Q^*} \geq 2P^*Q^*,$$

The first inequality holds by the Cauchy-Schwarz inequality.
Hence, (P^*, Q^*) is a NF solution. □

Theorem 3.4 proves the existence of one NF solution for BTSP that minimizes PQ, or equivalently minimizes $(\log P + \log Q)$. However, finding such a solution may be difficult as it requires to minimize a concave function. In the following, we show that all NF solutions can be found by minimizing an appropriate linear combination of P and Q based on the Weighted Sum Method [8]. More precisely, all NF solutions can be obtained by solving the following optimization problem

$$\mathcal{P}(\alpha) = \min \ \alpha P + Q \text{ s.t } (P, Q) \in S,$$

where $\alpha \in [0, 1]$ is the coefficient to be determined. For solving $\mathcal{P}(\alpha)$, we can solve the MIP (2) in Section 2 with $\alpha P + Q$ as the objective function instead of Q.

Let $\alpha \in \mathbb{R}_+$ and (P_α, Q_α) be an optimal solution of $\mathcal{P}(\alpha)$. Denote $T_\alpha := \alpha P_\alpha - Q_\alpha$ and $C_0 := \{\alpha \in \mathbb{R}_+ | T_\alpha = 0\}$. Due to the definitions of P and Q : P, Q respectively represent the total cost and the max-min distance in a solution tour, we always have $P > Q \geq 0$. Hence, if $\alpha \in C_0$ then $\alpha < 1$, if not $T_\alpha \geq P_\alpha - Q_\alpha > 0$.

Theorem 3.5. *$(P^*, Q^*) \in S$ is a NF solution if and only if there exists a coefficient $\alpha^* \in C_0$ such that (P^*, Q^*) is an optimal solution obtained by solving $\mathcal{P}(\alpha^*)$.*

Proof. Firstly, let (P^*, Q^*) be a NF solution and $\alpha^* = Q^*/P^*$. We will show that (P^*, Q^*) is an optimal solution of $\mathcal{P}(\alpha^*)$.

Since (P^*, Q^*) is a NF solution, we have

$$P'Q^* + Q'P^* \geq 2P^*Q^*, \ \forall (P', Q') \in S, \tag{6}$$

Since $\alpha^* = \frac{Q^*}{P^*}$, we have $\alpha^* P^* + Q^* = 2Q^*$.

Dividing two sides of (6) by $P^* > 0$ we obtain

$$2Q^* \leq \frac{Q^*}{P^*} P' + Q', \ \forall (P', Q') \in S, \tag{7}$$

So we deduce from (7)

$$\alpha^* P^* + Q^* \leq \alpha^* P' + Q', \ \forall (P', Q') \in S,$$

Hence, (P^*, Q^*) is an optimal solution of $\mathcal{P}(\alpha^*)$ and then $\alpha^* \in C_0$.

Now suppose $\alpha^* \in C_0$, we show that (P^*, Q^*) is a NF solution.

Since $T^* = \alpha^* P^* - Q^* = 0$, we have

$$\alpha^* = \frac{Q^*}{P^*}.$$

If (P^*, Q^*) is not a NF solution, there exists a solution $(P', Q') \in S$ such that

$$P'Q^* + Q'P^* < 2P^*Q^*,$$

We have then

$$\alpha P' + Q' = \frac{P'Q^* + Q'P^*}{P^*} < \frac{2P^*Q^*}{P^*} = \alpha^* P^* + Q^*,$$

which contradicts the optimality of (P^*, Q^*). □

Corollary 3.5.1. *NF solutions are Pareto-optimal solutions over S.*

Proof. Base on Theorem (3.5), all NF solutions can be obtained by solving $\mathcal{P}(\alpha)$. Now let (P^*, Q^*) be the corresponding NF solution of $\mathcal{P}(\alpha^*)$, we will show that (P^*, Q^*) is Pareto-optimal solution over S by contradiction.

Let us assume that there exists another solution $(P', Q') \in S$ such that $P' < P$ and $Q' < Q$. We have then

$$\alpha^* P' + Q' < \alpha^* P^* + Q^*,$$

which contradicts the optimality of (P^*, Q^*).

Hence, NF solutions are Pareto-optimal solutions over S. □

The following remark asserts that there may be more than one NF solution for BTSP.

Remark 3.6. Let (P, Q) and (P', Q') be two different feasible solutions in BTSP. The two inequalities

$$P'Q + Q'P \geq 2PQ \text{ and } P'Q + Q'P \geq 2P'Q'.$$

may be satisfied simultaneously.

The main question now is how to determine a coefficient α^* allowing to find a NF solution according to Theorem 3.5. In the next section, we present an iterative algorithm converging to α^* in a polynomial number of iterations. The value of α^* found by this algorithm corresponds to the NF solutions with the smallest total cost or the smallest max-min distance.

4 ALGORITHMS FOR FINDING EXTREME NASH FAIRNESS SOLUTIONS

As shown by Remark 3.6, there may be many NF solutions for an instance of BTSP. Among these solutions, two solutions may naturally be preferred to the others: the one with the smallest P and the one with the smallest Q. Let us call the first *Efficient Nash Fairness (ENF)* solution and the second *Balanced Nash Fairness (BNF)* solution. We call both ENF solution and BNF solution *extreme Nash Fairness* solution. In the following, we will focus first on ENF solution. As we will argue at the end of the section, all the subsequent results applied to ENF solution can be also applied to BNF solution with slight changes.

Theorem 4.1. *The ENF solution is unique.*

Proof. Suppose (P, Q) and (P', Q') are two ENF solutions. By the definition of ENF solution, we have $P \leq P'$ and $P' \leq P$ that imply $P = P'$.

Furthermore, (P, Q) and (P', Q') also are NF solutions. Hence

$$P'Q + Q'P \geq 2PQ \text{ and } P'Q + Q'P \geq 2P'Q'.$$

Since $P = P' > 0$, we have

$$Q + Q' \geq 2Q \text{ and } Q + Q' \geq 2Q'.$$

These equations lead to $Q = Q'$. □

We propose now an algorithm to find the coefficient α^* such that the optimal solution (P^*, Q^*) obtained by solving $\mathcal{P}(\alpha^*)$ is the ENF solution. This algorithm is inspired from the application of Newton method (or the Newton-Raphson method) to linear fractional programs that was first discussed by Isbell and Marlow [4] and then generalized to nonlinear fractional programs by Dinkelbach [3]. It is often called the Dinkelbach method. The algorithm can be stated as follows.

Algorithm 1

Input: An undirected graph G with n vertices, m edges and a positive cost vector $c \in \mathbb{R}^m_+$.

Output: A Hamiltonian tour corresponding to the ENF solution.

```
α_0 ← 1, i ← 0
repeat
    solve P(α_i) to obtain (P_i, Q_i) and X_i
    T_i ← α_i P_i − Q_i
    α_{i+1} ← Q_i / P_i
    i ← i + 1
until T_i = 0
return (P_i, Q_i, X_i).
```

where X_i represents the solution tour correspond to (P_i, Q_i).

Denote $\{\alpha_i\}$ as the sequence constructed by Algorithm 1. We will prove that Algorithm 1 terminates in a polynomial number of iterations and the obtained solution (P_i, Q_i) is the ENF solution. Our proof will use the following lemmas.

Lemma 4.2. *Let $\alpha, \alpha' \in \mathbb{R}_+$ and (P_α, Q_α), $(P_{\alpha'}, Q_{\alpha'})$ be the optimal solutions of $\mathcal{P}(\alpha)$ and $\mathcal{P}(\alpha')$ respectively, if $\alpha \leq \alpha'$ then $P_\alpha \geq P_{\alpha'}$ and $Q_\alpha \leq Q_{\alpha'}$.*

Proof. The optimality of (P_α, Q_α) and $(P_{\alpha'}, Q_{\alpha'})$ gives

$$\alpha P_\alpha + Q_\alpha \leq \alpha P_{\alpha'} + Q_{\alpha'}, \text{ and} \tag{8a}$$

$$\alpha' P_{\alpha'} + Q_{\alpha'} \leq \alpha' P_\alpha + Q_\alpha \tag{8b}$$

By adding both sides of (8a) and (8b), we obtain $(\alpha - \alpha')(P_\alpha - P_{\alpha'}) \leq 0$. Since $\alpha \leq \alpha'$, it follows that $P_\alpha \geq P_{\alpha'}$.

On the other hand, inequality (8a) implies $Q_{\alpha'} - Q_\alpha \geq \alpha(P_\alpha - P_{\alpha'}) \geq 0$ that leads to $Q_\alpha \leq Q_{\alpha'}$. □

Lemma 4.3. *During the execution of Algorithm 1, the sequence $\{\alpha_i\}$ is always non-negative and non-increasing. Moreover, $T_i \geq 0$, $\forall i \geq 0$.*

Proof. We have $\alpha_0 = 1 \geq 0$. Since $P > Q \geq 0$ $\forall (P, Q) \in S$, it follows that $\alpha_{i+1} = Q_i/P_i \geq 0$, $\forall i \geq 0$.

Our proof is given by induction on i. If $i = 0$, then $T_0 = \alpha_0 P_0 - Q_0 = P_0 - Q_0 \geq 0$ and $\alpha_0 = 1 \geq Q_0/P_0 = \alpha_1$. Suppose that our hypothesis is true until $i = k \geq 0$, we will prove that it is also true with $i = k + 1$.

Indeed, we have

$$T_{k+1} = \alpha_{k+1} P_{k+1} - Q_{k+1} = \frac{Q_k P_{k+1} - P_k Q_{k+1}}{P_k}. \tag{9}$$

The inductive hypothesis gives $\alpha_k \geq \alpha_{k+1}$ that implies $P_{k+1} \geq P_k > 0$ and $Q_k \geq Q_{k+1} \geq 0$ according to Lemma 4.2. It leads to $Q_k P_{k+1} - P_k Q_{k+1} \geq 0$ and then $T_{k+1} \geq 0$.

On the other hand, from the definition of α_{k+2}, we get

$$\alpha_{k+1} - \alpha_{k+2} = \frac{\alpha_{k+1} P_{k+1} - Q_{k+1}}{P_{k+1}} = \frac{T_{k+1}}{P_{k+1}} \geq 0. \tag{10}$$

That concludes the proof. □

Lemma 4.4. *Algorithm 1 terminates in a polynomial number of iterations.*

Proof. By contradiction, we first show that if $T_{i+1} > 0$ then $P_i < P_{i+1}$ and $Q_i > Q_{i+1}$.

Let assume that $P_i \geq P_{i+1}$. According to Lemma 4.3, $\alpha_i \geq \alpha_{i+1}$ that implies $P_i \leq P_{i+1}$ as the result of Lemma 4.2. Thus, $P_i = P_{i+1}$. From (9), if $T_{i+1} > 0$ then $Q_i P_{i+1} > Q_{i+1} P_i$. Since $P_i = P_{i+1} > 0$, we get $Q_i > Q_{i+1}$.

On the other hand, as (P_i, Q_i) is the optimal solution $\mathcal{P}(\alpha_i)$, it shows that $\alpha_i P_i + Q_i \leq \alpha_i P_{i+1} + Q_{i+1}$. Using $P_i = P_{i+1}$, we obtain $Q_i \leq Q_{i+1}$ which leads to a contradiction.

By repeating the same argument for $Q_i \leq Q_{i+1}$, we also have a contradiction.

Consequently, while $T_k > 0$, each i^{th} $(i \leq k)$ iteration of Algorithm 1 explores a Pareto-optimal solution (P_i, Q_i) with the distinct value of Q_i.

Now let c_i^{max} and c_i^{min} be the maximum edge cost and the minimum one in the solution (P_i, Q_i). We have $Q_i = c_i^{max} - c_i^{min}$ and due to the distinctness of Q_i, we obtain $c_i^{max} - c_i^{min} \neq c_j^{max} - c_j^{min}, \ \forall i \neq j$. We have then

$$(c_i^{max}, c_i^{min}) \not\equiv (c_j^{max}, c_j^{min}), \ \forall i \neq j,$$

Thus, each Pareto-optimal solution obtained by an iteration of Algorithm 1 has distinct pair of edges corresponding to the maximum edge cost and the minimum one. For graph G with n vertices, we have at most $O(n^2)$ edges and then the maximum number of distinct pairs of edges is $O(n^4)$.

Hence, the number of iterations in worst case is also $O(n^4)$. Consequently, Algorithm 1 terminates in a polynomial number of iterations. □

Theorem 4.5. *We obtain the ENF solution by Algorithm 1.*

Proof. Let α_n be the solution obtained by Algorithm 1 and (P_n, Q_n) be the optimal solution of $\mathcal{P}(\alpha_n)$. By the stopping criteria, $T_n = 0$ and $\alpha_n \in C_0$. Hence, $T_i > 0 \ \forall i < n$ and then $\alpha_i < \alpha_{i-1} \ \forall i \leq n$. According to Theorem 3.5, (P_n, Q_n) is a NF solution. We will prove that (P_n, Q_n) is a NF solution with the smallest total cost.

Assume that (P, Q) is another NF solution such that $P < P_n$. By Theorem 3.5, there exists $\alpha \in C_0$ such that (P, Q) is the optimal solution of $\mathcal{P}(\alpha)$. Furthermore, $\alpha \in (\alpha_n, \alpha_0]$ (if not, $P_\alpha \geq P_n$). Then there exists $0 < i \leq n$ such that $\alpha \in (\alpha_i, \alpha_{i-1}]$. Since $\alpha \leq \alpha_{i-1}$, $P \geq P_{i-1}$ and $Q \leq Q_{i-1}$ due to Lemma 4.2. Thus, we get

$$\frac{Q}{P} \leq \frac{Q_{i-1}}{P_{i-1}} \tag{11}$$

By the definitions of α and α_i, inequality (11) is equivalent to $\alpha \leq \alpha_i$ which leads to a contradiction. Hence, (P_n, Q_n) is the ENF solution. □

For finding the BNF solution, we use a similar algorithm starting from $\alpha_0 = 0$ instead of 1. In this case, the sequence $\{\alpha_i\}$ is always non-negative, non-decreasing and $T_i \leq 0 \ \forall i \geq 0$. After a polynomial number of iterations, we also obtain a coefficient $\alpha_n \in C_0$ and then we can prove that the optimal solution (P_n, Q_n) obtained by solving $\mathcal{P}(\alpha_n)$ is exactly the BNF solution. More precisely, we state this algorithm as follows.

Algorithm 2

Input: An undirected graph G with n vertices, m edges and a positive cost vector $c \in \mathbb{R}_+^m$.
Output: A Hamiltonian tour corresponding to the BNF solution.

$\alpha_0 \leftarrow 0, \ i \leftarrow 0$
repeat
  solve $\mathcal{P}(\alpha_i)$ to obtain (P_i, Q_i) and X_i
  $T_i \leftarrow \alpha_i P_i - Q_i$
  $\alpha_{i+1} \leftarrow Q_i / P_i$
  $i \leftarrow i + 1$
until $T_i = 0$
return (P_i, Q_i, X_i).

Then, we also state some lemmas and theorems to prove that we obtain BNF solution by using Algorithm 2.

Lemma 4.6. *During the execution of Algorithm 2, the sequence $\{\alpha_i\}$ is always non-negative and non-decreasing. Moreover, $T_i \leq 0, \ \forall i \geq 0$.*

Lemma 4.7. *Algorithm 2 terminates in a polynomial number of iterations.*

Theorem 4.8. *We obtain the BNF solution by Algorithm 2.*

Remark 4.9. In case $Q = 0$ (i.e a solution tour with all equal edge costs), the optimal solution of BTSP is also the BNF solution.

5 NUMERICAL RESULTS

Let us denote NFBTSP for Nash Fairness Balanced TSP, the problem of finding NF extreme solutions for BTSP. In this section, we conduct several experiments aiming at solving NFBTSP with Algorithms 1 and 2 on rather small size instances from TSPLIB [14]. We also solve the classical TSP and the BTSP [7] on the same instances. The obtained solutions for three problems will be then compared and commented.

For solving the three problems TSP, BTSP and NFBTSP, we design a simple branch-and-cut algorithm devoted to minimize a linear objective function over the MIP program in Section 2 (of course, for TSP the constraints (2f) and (2g) are exluded). We use CPLEX 12.10 to implement our branch-and-cut algorithm. The constraints (2d) are set as lazy cuts which are generated only when being violated by some integer solution. For BTSP and NFBTSP, we also have some specific branching rules for variable l inspired from the threshold algorithm [9], [7]. For NFBTSP, this branch-and-cut algorithm is used in each iteration of Algorithms 1 and 2 to solve the subproblem $P(\alpha)$. All the experiments are conducted on a PC Intel Core i5-9500 3.00GHz with 6 cores and 6 threads.

Table 2 presents the results of three problems TSP, BTSP and NFBTSP in several instances of TSPLIB with a range of nodes from 14 to 29. For NFBTSP, we also provide the number of iterations for finding respectively the ENF solution and the BNF solution (column "Iters") and its corresponding final value of α. We can see by the values of P and Q in this table that the ENF and BNF solutions for NFBTSP strike a better trade-off between two objectives: the total cost and the max-min distance comparing with those for the classical TSP and the BTSP. In particular, when the solutions for classical TSP and for BTSP are too different: small P and big Q in the

Table 2: TSP, BTSP and NFBTSP results on TSPLIB problems

Instance	TSP			BTSP			ENFTSP					BNFTSP				
	P	Q	Time	P	Q	Time	P	Q	Time	Iters	alpha	P	Q	Time	Iters	alpha
burma14	3323	472	0.10	4986	134	0.15	4986	134	1.40	4	0.027	4986	134	0.70	2	0.027
ulysses16	6859	1452	0.51	14032	868	0.70	7047	1399	4.42	3	0.199	13670	868	17.56	3	0.063
gr17	2085	311	0.21	4138	119	0.35	2227	234	5.53	3	0.105	3346	139	8.63	4	0.042
gr21	2707	328	0.01	8630	115	0.68	2989	278	1.24	3	0.093	5945	120	18.60	3	0.020
ulysses22	7013	1490	141.46	19168	868	1.68	7070	1471	356.97	3	0.208	7070	1471	651.38	4	0.208
gr24	1272	83	0.07	3886	33	1.85	1282	81	1.17	3	0.063	3847	33	49.17	3	0.009
fri26	937	118	0.13	2458	21	2.72	980	82	24.39	3	0.084	2447	21	55.06	3	0.009
bays29	2020	140	0.55	6757	38	7.95	3449	59	2,033.04	5	0.017	4558	44	2471.15	3	0.010
bayg29	1610	86	0.31	4252	29	4.12	1817	63	44.92	3	0.035	3246	35	2567.73	4	0.011

solution for TSP and big P and small Q in the one of BTSP, the two extreme NF solutions for NFTSP give two interesting alternatives. More precisely, the ENF solution (respectively BNF solution) offers a better alternative than the solution of TSP (respectively BTSP) with a significant drop on the value of Q (respectively P) and a slight growth on the value of P (respectively Q). Table 2 also indicates that Algorithms 1 and 2 seem to converge very quickly after only maximum 5 iterations. One important issue is the CPU time for solving NFBTSP is quite huge comparing with the CPU time spent for solving TSP and BTSP. A deeper analysis on the iterations of Algorithms 1 and 2 tells us that the smaller is the value of α, the more difficult is $P(\alpha)$. Especially, the CPU time spent for solving $P(\alpha)$ in the last iteration occupies a very big part of the overall CPU time. Hence, a special-purpose algorithm for solving $P(\alpha)$ may be more interesting than simply optimizing a linear function over the MIP given in Section 2.

6 CONCLUSION

In this paper, we have made use of Nash fairness equilibrium to achieve a trade-off between the efficiency estimated by the total cost and the balancedness estimated by the max-min distance in solutions for balanced TSP (BTSP). We have proven first the existence of Nash Fairness (NF) equilibrium solutions for BTSP. Second, we have designed Newton-based iterative algorithms to find the two extreme NF solutions: the one with minimum total cost and the one with minimum max-min distance. Numerical results conducted on small size instances from TSPLIB have shown that comparing with the optimal solutions of the original BTSP, the NF solutions found by our algorithm have much smaller total cost with a reasonable augmentation of the max-min distance. An important notice is that the results in this paper can be also applied to various balanced combinatorial optimization [9] problems such as balanced assignment problem , balanced spanning tree problem [2],... Another future development of our work is the improvement of the CPU time for solving the problem $P(\alpha)$ especially when α is very small. One of the possible direction is to study the possibility of stopping the solving of $P(\alpha)$ once a improved solution of (P, Q) is obtained instead of stopping at optimality.

REFERENCES

[1] Dimitris Bertsimas, Vivek F.Farias, and Nikolaos Trichakis. January-February 2011. The Price of Fairness. *Operations Research* 59-1 (January-February 2011), 17–31.

[2] Paolo M. Camerini, Francesco Maffioli, Silvano Martello, and Paolo Toth. 1986. Most and least uniform spanning trees. *Discrete Applied Mathematics* 15, 2 (1986), 181–197. https://doi.org/10.1016/0166-218X(86)90041-7

[3] W. Dinkelbach. 1967. On nonlinear fractional programming. *Management Sci* 13 (1967), 492–498.

[4] J.R. Isbell and W.H. Marlow. 1956. Attrition games. *Naval Res. Logist. Quart 3* (1956), 71–94.

[5] FP Kelly, AK Maullo, and DKH Tan. November 1997. Rate control for communication networks : shadow prices, proportional fairness and stability. (November 1997).

[6] Joris Kinable, Bart Smeulders, Eline Delcour, and Frits CR Spieksma. 2017. Exact algorithms for the equitable traveling salesman problem. *European Journal of Operational Research* 261, 2 (2017), 475–485.

[7] John Larusic and Abraham P Punnen. 2011. The balanced traveling salesman-problem. *Computers & Operations Research* 38, 5 (2011), 868–875.

[8] R. Timothy Marler and Jasbir Singh Arora. 2010. The weighted sum method for multi-objective optimization: New insights. *Structural and Multidisciplinary Optimization 41(6):853-862* (2010).

[9] S Martello, W.R Pulleyblank, P Toth, and D de Werra. 1984. Balanced optimization problems. *Operations Research Letters* 3, 5 (1984), 275–278. https://doi.org/10.1016/0167-6377(84)90061-0

[10] Fiorenzo Mornati. 2013. Pareto Optimality in the work of Pareto. *European Journal of Social Sciences* 51-2 (2013), 65–82.

[11] W. Ogryczak, Hanan Luss, Michal Pioro, Dritan Nace, and Artur Tomaszewski. 2014. Fair Optimization and Networks: A survey. *Journal of Applied Mathematics* (2014), 1–26.

[12] Robert D Plante. 1988. The nozzle guide vane problem. *Operations research* 36, 1 (1988), 18–33.

[13] Jasbir Singh Arora R. Timothy Marler. June 2010. The weighted sum method for multi-objective optimization: New insights. 41(6) (June 2010), 853–862.

[14] Gerhard Reinelt. 1991. TSPLIB A traveling salesman problem library. *ORSA journal on computing* 3, 4 (1991), 376–384.

[15] Zeev Zeitlin. 1981. Minimization of maximum absolute deviation in integers. *Discrete Applied Mathematics* 3, 3 (1981), 203–220.