Nash balanced assignment problem

Minh Hieu Nguyen, Mourad Baiou, Viet Hung Nguyen

Nash balanced assignment problem

Minh Hieu Nguyen[1] , Mourad Baiou[1] , Viet Hung Nguyen[1,2]

[1] INP Clermont Auvergne, Univ Clermont Auvergne, Mines Saint-Etienne, CNRS, UMR 6158 LIMOS, 1 Rue de la Chebarde, Aubiere Cedex, France
[2] Corresponding author : `viet_hung.nguyen@uca.fr`
`{minh_hieu.nguyen,mourad.baiou}@uca.fr`

Abstract. In this paper, we consider a variant of the classic *Assignment Problem* (AP), called the *Balanced Assignment Problem* (BAP) [2]. The BAP seeks to find an assignment solution with the smallest value of *max-min distance*: the difference between the maximum assignment cost and the minimum one. However, by minimizing only the max-min distance, the total cost of the BAP solution is neglected, and it may lead to a very inefficient solution in terms of the total cost. Hence, we propose a fair way based on Nash equilibrium [1] [3], [4] to inject the total cost into the objective function of the BAP for finding assignment solutions having a better trade-off between the two objectives: the first aims at minimizing the total cost and the second aims at minimizing the max-min distance. For this purpose, we introduce the concept of *Nash Fairness* (NF) solutions based on the definition of proportional-fair scheduling adapted in the context of the AP: a transfer of utilities between the total cost and the max-min distance is considered to be fair if the percentage increase in the total cost is smaller than the percentage decrease in the max-min distance and vice versa.
We first show the existence of an NF solution for the AP, which is exactly the optimal solution minimizing the product of the total cost and the max-min distance. However, finding such a solution may be difficult as it requires minimizing a concave function. The main result of this paper is to show that finding all NF solutions can be done in polynomial time. For that, we propose a Newton-based iterative algorithm converging to NF solutions in polynomial time. It consists in optimizing a sequence of linear combinations of the two objectives based on the Weighted Sum Method [5]. Computational results on various instances of the AP are presented and commented.

Keywords: Combinatorial optimization · Balanced assignment problem · Proportional fairness · Proportional-fair scheduling · Weighted Sum Method

1 Introduction

The *Assignment Problem* (AP) is a fundamental combinatorial optimization problem. It can be formally defined as follows. Given a set of n workers, a set of n jobs, and a $n \times n$ cost matrix whose elements are positive, representing the assignment of any worker to any job, the AP aims at finding a one-to-one worker-job assignment (i.e., a bipartite perfect matching) that minimizes certain objective functions.

In the classic AP, we seek to find an assignment solution minimizing the total cost. It is a well-known optimization problem that can be solved by the Hungarian algorithm in $O(n^3)$ [7]. The *Balanced Assignment Problem* (BAP) is a variant of the classic AP where instead of minimizing the total cost, we minimize *the max-min distance*, which is the difference between the maximum assignment cost and the minimum one in the assignment solution. In [2], the authors proposed an efficient threshold-based algorithm to solve the BAP in $O(n^4)$. However, by minimizing only the max-min distance, the total cost of the BAP solution is neglected, and it may lead to a very inefficient solution in terms of the total cost.

In this paper, to overcome the possible inefficiency of the solutions for the BAP, we propose a fair way based on Nash equilibrium to inject the total cost into the objective function of the BAP. Nash equilibrium is the most common optimality notion for sharing resources among users [1] [3],[4]. We are interested in assignment solutions for the AP achieving a Nash equilibrium between two players: the first aims to minimize the total cost, and the second aims to minimize the max-min distance. For this purpose, we introduce the *Nash Fairness* (NF) solutions based on the definition of proportional-fair scheduling adapted in the context of the AP: a transfer of utilities between the total cost and the max-min distance is considered to be fair if the percentage increase in the total cost is smaller than the percentage decrease in the max-min distance and vice versa.

We have introduced the concept of NF solutions for the Balanced Traveling Salesman Problem (BTSP) in a recent paper [12]. In [12], we proposed an algorithm converging to particular NF solutions called *extreme NF solutions* having respectively smallest value of total cost and max-min distance. Similar to [12], in this current paper, we also introduce the concept of NF solutions in the context of the AP. But, the main contribution of our work in this paper is a stronger result than in [12]: we provide an algorithm for finding all NF solutions in polynomial time.

The paper is organized as follows. In Section 2, we introduce a linear programming (LP) formulation for the BAP. The concept of NF solutions is presented in Section 3. In particular, we prove the existence of NF solutions for the AP and show that they are optimal solutions of a weighted sum objective problem. In Section 4, an algorithm for finding all NF solutions and computational results on various instances of the AP is given. Finally, this paper's conclusion and future works are discussed in Section 5.

2 LP formulation for the BAP

We consider an AP with a $n \times n$ cost matrix whose elements $c_{i,j}$ are positive, and they represent the cost assignments between worker i and job j. We first present the linear programming (LP) formulation for the BAP as follows

$$\min Q = u - l \tag{1a}$$

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

$$u \geq \sum_{j\in[n]} c_{i,j}x_{i,j} \qquad \forall i \in [n] \tag{1c}$$

$$l \leq \sum_{j\in[n]} c_{i,j}x_{i,j} \qquad \forall i \in [n] \tag{1d}$$

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

where $[n] = \{1, ..., n\}$ and $x_{i,j}$ represents the assignment between worker i and job j corresponding to the cost assignment $c_{i,j}$. To calculate the max-min distance Q, we need to determine the maximum and the minimum assignment costs u and l in the assignment solution. Constraints (1c) obviously allow bounding u from below by the maximum assignment cost in the assignment solution. Similarly, constraints (1d) allow bounding l from above by the minimum assignment cost in the assignment solution. As $Q = u - l$ is minimized, u and l will respectively take the values of the maximum and the minimum assignment costs. We will show that this LP formulation has an integral optimal solution corresponding to an assignment solution (i.e., bipartite perfect matching).

Theorem 1. *This LP formulation always has an optimal solution where the variables take integer values.*

Proof. The objective function assures that u and l will be equal, respectively, to the maximum and the minimum assignment costs in the optimal solution. Consequently, the optimal solution of this LP is always integral because the constraints matrix of (1b) is totally unimodular (e.g., see [6]), and the constraints (1c) and (1d) are simply bound constraints. □

In the following, we solve the classic AP as well as the BAP for several instances of the AP where we generate random uniform $c_{i,j}$ in $[1, 10^2]$. Optimal solutions of these instances are shown in Table 1 where *assignx* represents an instance of the AP with a cost matrix of dimension $x \times x$ and P, Q represent respectively the total cost and the max-min distance in the optimal solution. We use CPLEX 12.10 on a PC Intel Core i5-9500 3.00GHz with 6 cores and 6 threads for solving the classic AP and the BAP. We can see in each instance of the AP that the optimal solutions for the classic AP may be undesirable with respect to those for the BAP and vice versa: inefficient values of Q in the optimal solutions for the classic AP compared with those in the optimal solutions for the BAP and inefficient values of P in the optimal solutions for the BAP comparing with those in the optimal solutions for the classic AP.

Hence, this paper aims to use a fair way to inject the total cost into the objective function for finding assignment solutions having a better trade-off between the two objectives.

Table 1. Optimal solutions for the classic AP and the BAP

Instance	classic AP			BAP		
	P	Q	Time	P	Q	Time
assign3	100	6	0.01	200	3	0.01
assign6	114	15	0.01	173	10	0.01
assign17	68	10	0.01	242	3	2.43
assign25	189	27	0.03	2004	11	2.44
assign30	157	18	0.04	643	7	32.4

3 Nash fairness solutions for the AP

We have introduced the concept of Nash fairness (NF) solution for the Balanced Traveling Salesman Problem (BTSP) [12]. In this section, we restate the concept of NF solutions in the context of the AP, and we put the proofs of theorems in the Appendix.

3.1 Proportional fairness

NF solutions for the AP are closely related to the concept of proportional fairness for multiple players problem [1]. In the context of multiple players problem, 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 $u_i(x) \geq u_i(y), \forall i \in I$.

NF solutions for two-player problem [3] are defined by using the Nash standard of comparison: 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 introduced by Bertsimas et al. [1] 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 [3], [1], [4].

Definition 1. *[1] $x^{NF} \in U$ is an NF solution for multiple players problem if and only if*

$$\sum_{j=1}^{n} \frac{u_j(x) - u_j(x^{NF})}{u_j(x^{NF})} \leq 0, \ \forall x \in U, \tag{2}$$

where n is the number of players and $u_j(x) > 0, \ \forall j \in I, \forall x \in U$.

3.2 Characterization of NF solutions for the AP

Let P, Q now represent the total cost and the max-min distance in a feasible assignment solution for the AP. From the definitions of P and Q we have $P > Q \geq 0$. We first suppose that $Q > 0$. As P, Q now are two strictly positive utility functions, we have a two-player problem. In the usual definition of NF solutions [3], [1], an alternative assigned a greater value is preferred. However, in the context of the AP, we prefer the alternative assigned a smaller value for two utility functions P and Q. Thus, the aggregate proportional change should be greater than or equal to 0 in the definition of NF solutions for the AP. That is to say, the sum of relative gains when switching from NF solutions to another feasible solution is not negative in the context of the AP.

We denote the value solution for the total cost and the max-min distance corresponding to a feasible assignment solution by (P, Q). Let (P^*, Q^*) be an NF solution for the AP, condition (2) can be translated into the context of the AP as follows

$$\frac{P - P^*}{P^*} + \frac{Q - Q^*}{Q^*} \geq 0,\ \forall (P, Q) \in \mathcal{S}, \tag{3}$$

which is equivalent to

$$PQ^* + QP^* \geq 2P^*Q^*,\ \forall (P, Q) \in \mathcal{S}, \tag{4}$$

where $\mathcal{S}$ is the set of solutions (P, Q) corresponding to all feasible assignment solutions for the AP.

Note that in case $Q^* = 0$, the condition (4) is also satisfied. Hence, NF solution for the AP can be generally stated as follows

Lemma 1. *[12] $(P^*, Q^*) \in \mathcal{S}$ is a NF solution for the AP if and only if $PQ^* + QP^* \geq 2P^*Q^*,\ \forall (P, Q) \in \mathcal{S}$.*

Remark 1. An assignment solution with equal assignment costs (i.e., $Q = 0$) is a NF solution.

3.3 Existence of NF solutions

In this section, we first show the existence of NF solutions for the AP. Let us recall that in the multiple players problem mentioned in Section 3.1 where we prefer an alternative assigned a greater value, NF solutions can be obtained equivalently as the optimal solution of the problem

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

provided that U is convex. Notice that the above NF solution equivalently maximizes the product of the utilities over U [1].

On the contrary, in the AP, there exist NF solutions that can be obtained by minimizing instead of maximizing the product of the utilities.

Theorem 2. *[12]* $(P^*, Q^*) = \arg\min_{(P,Q)\in\mathcal{S}} PQ$ *is a NF solution.*

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

$$(P^*, Q^*) = \underset{(P,Q)\in\mathcal{S}}{\arg\min}\, PQ.$$

Now $\forall (P', Q') \in \mathcal{S}$ 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 is held by the Cauchy-Schwarz inequality.
Hence, (P^*, Q^*) is a NF solution. □

Theorem 2 proves the existence of NF solutions for the AP that minimize PQ, or equivalently minimize $(\log P + \log Q)$. We call such solutions *Product Nash Fairness* (PNF) solutions. However, finding PNF solutions may be difficult as it requires minimizing a concave function. In the following, we show that all NF solutions can be obtained by solving the following optimization problem

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

where $\alpha \in [0, 1]$ is a coefficient to be determined. For solving $\mathcal{P}(\alpha)$, we solve the LP formulation 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 $\mathcal{C}_0 := \{\alpha \in \mathbb{R}_+ | \alpha P_\alpha - Q_\alpha = 0\}$. Hence, if $\alpha \in \mathcal{C}_0$ (i.e., $T_\alpha = 0$) then $\alpha < 1$, otherwise $\alpha P_\alpha - Q_\alpha \geq P_\alpha - Q_\alpha > 0$.

Notice that we assume the existence of the algorithms for solving the problem $\mathcal{P}(\alpha)$ with $\alpha \in [0, 1]$. The solution of $\mathcal{P}(\alpha)$ will be characterized only by the solution (P, Q) and not by the decision vector of the solution. Thus, two solutions having the exact value of (P, Q) will be considered the same. In addition, by solving the problem $\mathcal{P}(\alpha)$ when $\alpha = 0$, i.e., the problem minimizing Q, we can determine the particular NF solutions with $Q = 0$ (if they exist). In the rest of this paper, we only consider the case $Q > 0$.

Theorem 3. *[12]* $(P^*, Q^*) \in \mathcal{S}$ *is a NF solution if and only if* (P^*, Q^*) *is an optimal solution of* $\mathcal{P}(\alpha^*)$ *where* $\alpha^* = \frac{Q^*}{P^*}$.

Proof. See Appendix. □

Theorem 3 states a necessary and sufficient condition for the NF solutions. We are interested now in the following question: Given a feasible solution $(P', Q') \in \mathcal{S}$, how to assert that (P', Q') is an NF solution? We give the answer to this question in the next proposition.

Proposition 1. *Given a feasible solution* $(P', Q') \in \mathcal{S}$. *Let* $\alpha' = \frac{Q'}{P'}$ *and* (P^*, Q^*) *be an optimal solution of* $\mathcal{P}(\alpha')$, *then* (P', Q') *is a NF solution* $\iff$ $\alpha'P^* + Q^* - \alpha'P' - Q' = 0$.

Proof. $\Longrightarrow$ If $\alpha' P^* + Q^* - \alpha' P' - Q' = 0$ then (P', Q') is also an optimal solution of $\mathcal{P}(\alpha')$. Since $\alpha' = \frac{Q'}{P'}$, (P', Q') is a NF solution due to Theorem 3.

$\Longleftarrow$ If (P', Q') is a NF solution then (P', Q') is also an optimal solution of $\mathcal{P}(\alpha')$ due to Theorem 3. Thus, $\alpha' P^* + Q^* = \alpha' P' + Q'$ which leads to $\alpha' P^* + Q^* - \alpha' P' - Q' = 0$.

We call (P, Q) a Pareto-optimal solution for the AP if (P, Q) is an optimal solution of $\mathcal{P}(\alpha)$ where $\alpha \in [0, 1]$. By Theorem 3, an NF solution is necessarily a Pareto-optimal solution but not vice versa.

Proposition 1. *There may be more than one NF solution for the AP.*

Proof. Let us illustrate this by an instance of the AP having the following cost matrix :

$$A = \begin{bmatrix} 30 & 48 & 68 \\ 44 & 65 & 34 \\ 67 & 36 & 48 \end{bmatrix}$$

By verifying all feasible assignment solutions in this instance, we easily obtain three assignment solutions $(1-1, 2-3, 3-2), (1-2, 2-1, 3-3)$ and $(1-3, 2-2, 3-1)$ corresponding to three NF solutions $(100, 6)$, $(140, 4)$ and $(200, 3)$. Note that $i - j$ where $1 \leq i, j \leq 3$ represents the assignment between worker i and job j in the solution of this instance. □

The main question now is how to determine the coefficients of $\mathcal{C}_0$ corresponding one-to-one to all the NF solutions according to Theorem 3. The next section presents an algorithm for finding all NF solutions in polynomial time.

4 Finding all NF solutions for the AP

In [12], we proposed an algorithm converging to *extreme NF solutions* having respectively the smallest value of P and Q. This section introduces another one to find all NF solutions in polynomial time. Obviously, they include the PNF solutions minimizing PQ.

4.1 Upper bound for the number of NF solutions

We call $(P, Q), (P', Q')$ two distinct solutions if $(P, Q) \not\equiv (P', Q')$. We will show that the number of NF solutions for the AP is at most $C^2_{n^2} + n$ where $C^2_{n^2} = \frac{n^2(n^2-1)}{2}$ by the following lemma and theorem.

Lemma 2. *If* $(P, Q) \not\equiv (P', Q')$ *are two distinct NF solutions having* $Q, Q' > 0$ *then* $Q \neq Q'$.

Proof. Suppose that $Q = Q' > 0$. Using the definition of NF solution, we have

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

which is equivalent to

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

Hence we obtain $P = P'$, which leads to a contradiction. □

Theorem 4. *The number of NF solutions for the AP is at most $C^2_{n^2} + n$.*

Proof. If (P, Q) is an NF solution and $Q = 0$, the corresponding assignment solution has n equal assignment costs. For the AP with $n \times n$ cost matrix, we have n^2 assignments, and consequently, there are at most n distinct NF solutions having the same value $Q = 0$.

We now consider the NF solutions with $Q > 0$. We will show that the number of NF solutions having $Q > 0$ is at most $C^2_{n^2}$.

Let c_i^{max} and c_i^{min} be the maximum and the minimum assignment cost in the assignment solution corresponding to (P_i, Q_i) then $Q_i = c_i^{max} - c_i^{min}$. As shown in Lemma 2, for two distinct NF solutions $(P_i, Q_i), (P_j, Q_j)$ with both Q_i and Q_j strictly positive we obtain $Q_i \neq Q_j$ which is equivalent to $c_i^{max} - c_i^{min} \neq c_j^{max} - c_j^{min}$. We have then $(c_i^{max}, c_i^{min}) \not\equiv (c_j^{max}, c_j^{min})$. Thus, the assignment solutions corresponding to (P_i, Q_i), (P_j, Q_j) have distinct pairs of assignments representing the maximum and the minimum assignment cost. As we have at most n^2 distinct assignments, the number of distinct pairs of assignments is at most $C^2_{n^2}$. Consequently, the number of NF solutions having $Q > 0$ is at most $C^2_{n^2}$. Hence, the total number of NF solutions for the AP is at most $C^2_{n^2} + n$. □

By Theorem 4, the number of Pareto-optimal solutions having distinct values of Q is at most $C^2_{n^2}$.

4.2 Algorithm for finding all NF solutions

As shown in Theorem 3, each element $\alpha^* \in \mathcal{C}_0$ corresponds to a NF solution and vice versa. For all NF solutions, we aim to find all elements of $\mathcal{C}_0$. Our main idea is that from each $\alpha_0 \in [0, 1]$, we first use a procedure called *Find*() to find $\alpha_k \in \mathcal{C}_0$ satisfying α_k is the unique element $\in \mathcal{C}_0$ between α_0 and α_k.

Thus, let $\mathcal{I}$ be the set containing the intervals $[\alpha_i, \alpha_j]$ corresponding to distinct Pareto-optimal solutions $(P_i, Q_i), (P_j, Q_j)$. We use another procedure called *Test()* for verifying whether there exists an NF solution corresponding to $c_k \in [\alpha_i, \alpha_j]$ or not. Using these procedures, the algorithm for finding all NF solutions can be stated as follows.

Algorithm 1 Finding all NF solutions

Input: An AP with positive values in a $n \times n$ cost matrix.
Output: Set $\mathcal{C}_0$ whose elements correspond to all NF solutions for this AP.

$c_0 \leftarrow Find(0)$
$c_1 \leftarrow Find(1)$
if $c_1 = c_0$ **then**
  $\mathcal{C}_0 = \{c_0\}$
else
  $\mathcal{I} = \{[c_0, c_1]\}$, $\mathcal{C}_0 = \{c_0, c_1\}$
  for $[c_i, c_j] \in \mathcal{I}$ **do**
    $Test([c_i, c_j])$
  end for
end if
procedure FIND(α_0)
  solve $\mathcal{P}(\alpha_0)$ to obtain an optimal solution (P_0, Q_0)
  $i \leftarrow 0$
  repeat
    $\alpha_{i+1} \leftarrow Q_i/P_i$
    solve $\mathcal{P}(\alpha_{i+1})$ to obtain an optimal solution (P_{i+1}, Q_{i+1})
    $T_i \leftarrow \alpha_{i+1}P_{i+1} + Q_{i+1} - \alpha_{i+1}P_i - Q_i$
    $i \leftarrow i + 1$
  until $T_i = 0$
  Return α_{i+1}.
end procedure
procedure TEST($[c_i, c_j]$)
  $\alpha_k = \frac{Q_i - Q_j}{P_j - P_i}$
  if $\mathcal{P}(\alpha_k)$ has an optimal solution different to (P_i, Q_i) and (P_j, Q_j) **then**
    $c_k \leftarrow Find(\alpha_k)$
    if $c_i = c_k$ **then**
      $[c_i, c_j] \leftarrow [\alpha_k, c_j]$ ▷ Update the elements of $\mathcal{I}$
      $Test([\alpha_k, c_j])$
    else if $c_k = c_j$ **then**
      $[c_i, c_j] \leftarrow [c_i, \alpha_k]$
      $Test([c_i, \alpha_k])$
    else ▷ c_k is a new element of $\mathcal{C}_0$
      $[c_i, c_j] \leftarrow [c_i, c_k], [c_k, c_j]$
      $\mathcal{C}_0 \leftarrow \mathcal{C}_0 \cup c_k$
      $Test([c_i, c_k])$, $Test([c_k, c_j])$
    end if
  end if
end procedure

Let α_0 be the initial point, $T_i = \alpha_{i+1}P_{i+1} + Q_{i+1} - \alpha_{i+1}P_i - Q_i$ and $\{\alpha_i\}_{i \geq 0}$ denote the sequence constructed by Procedure $Find(\alpha_0)$. We show that Algorithm 1 explores all NF solutions in polynomial time by the following lemmas and theorem. Due to lack of space, we put some proofs in Appendix.

Lemma 3. *[12] 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 $\alpha < \alpha'$ then $P_\alpha \geq P_{\alpha'}$ and $Q_\alpha \leq Q_{\alpha'}$.*

Proof. See Appendix. □

As a consequence of Lemma 3, if (P_α, Q_α), $(P_{\alpha'}, Q_{\alpha'})$ are optimal solutions of $\mathcal{P}(\alpha)$ and $\mathcal{P}(\alpha')$ and $P_\alpha < P_{\alpha'}$ (or $Q_\alpha > Q_{\alpha'}$) then we obtain $\alpha \geq \alpha'$.

Lemma 4. *[12] During the execution of Procedure Find(α_0) in Algorithm 1, $\alpha_{i+1} \in]0,1[$ and $T_i \leq 0, \forall i \geq 0$. Moreover, if $\alpha_0 P_0 - Q_0 > 0$ then the sequence $\{\alpha_i\}_{i\geq 0}$ is strictly decreasing. Otherwise, if $\alpha_0 P_0 - Q_0 < 0$ then the sequence $\{\alpha_i\}_{i\geq 0}$ is strictly increasing.*

Proof. See Appendix. □

Lemma 5. *[12] From each $\alpha_0 \in [0,1]$, Procedure Find(α_0) converges to a coefficient $\alpha_k \in C_0$ satisfying α_k is the unique element $\in \mathcal{C}_0$ between α_0 and α_k.*

Proof. See Appendix. □

Lemma 6. *Procedure Find(α_0) terminates in polynomial time.*

Proof. If $\alpha_0 P_0 - Q_0 = 0$ then $\alpha_0 = Q_0/P_0 = \alpha_1$. Thus, $T_0 = 0$ because both (P_0, Q_0) and (P_1, Q_1) are the optimal solutions of $\mathcal{P}(\alpha_0)$. Consequently, Procedure Find(α_0) returns the value α_0.

Suppose that Procedure *Find*(α_0) converges to $\alpha_k \in \mathcal{C}_0$ in $k+1$ iterations. We have $T_i < 0, \forall 0 \leq i \leq k-1$. We only consider the nontrivial case where $k > 0$ (i.e., $\alpha_0 P_0 - Q_0 \neq 0$).

Without loss of generality, we suppose that $\alpha_0 P_0 - Q_0 > 0$ that leads to $\{\alpha_i\}_{i\geq 0}$ is strictly decreasing. Since $\alpha_i > \alpha_{i+1}$, we have $Q_i \geq Q_{i+1}$ and $P_i \leq P_{i+1}, \forall i \geq 0$ due to Lemma 3.

We first show that if $T_i < 0$ then $Q_i > Q_{i+1}, \forall i \geq 0$.

Let us assume that $Q_i = Q_{i+1}$. The optimality of (P_{i+1}, Q_{i+1}) gives

$$\alpha_{i+1} P_{i+1} + Q_{i+1} \leq \alpha_{i+1} P_i + Q_i,$$

Using $Q_i = Q_{i+1}$ and $\alpha_{i+1} > 0$, we obtain $P_{i+1} \leq P_i$. Thus, $P_i = P_{i+1}$.

Since $P_i = P_{i+1}$ and $Q_i = Q_{i+1}$, it implies $T_i = 0$ which leads to a contradiction.

Consequently, the execution of Procedure *Find*(α_0) explores k Pareto-optimal solutions having distinct values of Q. As the number of Pareto-optimal solutions having distinct value of Q is at most $C^2_{n^2}$, Procedure *Find*(α_0) terminates after a polynomial number of iterations. Hence, Procedure *Find*(α_0) terminates in polynomial time cause the LP formulation in Section 2 for $\mathcal{P}(\alpha)$ can be solved in polynomial time. □

Now by using the following lemma, we show that Procedure *Test*() can be used for verifying the existence of a Pareto-optimal solution (as well as NF solution) in each interval $[\alpha_i, \alpha_j]$.

Lemma 7. *Given an interval* $[\alpha_i, \alpha_j]$ *defined by* $0 \leq \alpha_i < \alpha_j \leq 1$ *corresponding to two distinct Pareto-optimal solutions* (P_i, Q_i) *and* (P_j, Q_j)*. Let* $\alpha^* = \frac{Q_j - Q_i}{P_i - P_j}$*, if* $\mathcal{P}(\alpha^*)$ *has no Pareto-optimal solution which is different to* (P_i, Q_i) *and* (P_j, Q_j)*, there does not have another one in* $[\alpha_i, \alpha_j]$*.*

Proof. Using Lemma 3 with $\alpha_i < \alpha_j$ and $(P_i, Q_i), (P_j, Q_j)$ be the two distinct Pareto-optimal solutions, we have $Q_i < Q_j, P_i > P_j$.

We first show that $\alpha^* \in [\alpha_i, \alpha_j]$.

Due to the optimality of (P_i, Q_i) and (P_j, Q_j) we obtain

$$\alpha_i P_i + Q_i \leq \alpha_i P_j + Q_j,$$
$$\alpha_j P_j + Q_j \leq \alpha_j P_i + Q_i,$$

Hence, $\alpha_i \leq \frac{Q_j - Q_i}{P_i - P_j} \leq \alpha_j$ which leads to $\alpha_i \leq \alpha^* \leq \alpha_j$.

Now suppose that we do not obtain any Pareto-optimal solution by solving $\mathcal{P}(\alpha^*)$ which is different to (P_i, Q_i) and (P_j, Q_j), we will show that there does not have another one in $[\alpha_i, \alpha_j]$.

Since $\alpha^* = \frac{Q_j - Q_i}{P_i - P_j}$, we have $\alpha^* P_i + Q_i = \alpha^* P_j + Q_j$. That means in this case (P_i, Q_i) and (P_j, Q_j) are two optimal solutions of $\mathcal{P}(\alpha^*)$. If there exists another Pareto-optimal solution (P, Q) of $\mathcal{P}(\alpha)$ where $\alpha \in [\alpha_i, \alpha_j]$, we have then $Q_i < Q < Q_j$ and $P_i > P > P_j$. Applying the consequence of Lemma 3 with $Q < Q_j$ we obtain $\alpha \leq \alpha^*$. Similarly, from $Q > Q_i$ we obtain $\alpha \geq \alpha^*$.

Hence, $\alpha = \alpha^*$ and then $\mathcal{P}(\alpha^*)$ has the Pareto-optimal solution (P, Q) which is different to (P_i, Q_i) and (P_j, Q_j). It leads to a contradiction. □

Theorem 5. *Algorithm 1 explores all NF solutions in polynomial time.*

Proof. As a consequence of Lemma 5, the interval $[c_0, c_1]$ contains all elements of $\mathcal{C}_0$.

We know that the number of Pareto-optimal solutions having distinct values of Q is at most $C^2_{n^2}$. Consequently, $[c_0, c_1]$ can be separated by at most $C^2_{n^2} - 1$ intervals $[c_i, c_j]$ such that $c_i < c_j$ correspond to two Pareto-optimal solutions having distinct values of Q and there intervals have no common points except the endpoints. By using Procedure *Test*(), each recursive call give us a Pareto-optimal solution or show that we have explored an interval having no Pareto-optimal solution and consequently no NF solution inside. As we use Procedure *Find*(α) in each recursive call, Procedure *Test*() also terminates in polynomial time. Moreover, we obtain an NF solution from each Pareto-solution found with a corresponding coefficient $\in [c_0, c_1]$. Since Algorithm 1 terminated as the interval $[c_0, c_1]$ is totally explored, it found all NF solutions in polynomial time. The PNF solutions minimizing PQ can be easily determined by comparing the products of all NF solutions. □

4.3 Numerical results

Let us denote NFAP, and PAF as the problems of finding all NF solutions and finding the NF solutions minimizing PQ for the AP. In this section, we conduct several experiments aimed at solving the NFAP with CPLEX 12.10 on some instances from the data sets of the AP [11] as well as on various instances in which we generate a random uniform cost matrix. We also solve the classic AP and the BAP in the same instances. All the experiments are conducted on a PC Intel Core i5-9500 3.00GHz with 6 cores and 6 threads.

Table 2. Numerical results for the classic AP, BAP, and NFBAP

Instance	**classic AP**			**BAP**			**PNF**				**NFAP**
	P	**Q**	**Time**	**P**	**Q**	**Time**	**P**	**Q**	**Time**	α	**all NF solutions**
assign3	100	6	0.01	200	3	0.05	140	4	0.11	0.028	(100,6), (140,4), (200,3)
assign4	70	9	0.01	196	3	0.25	196	3	1.20	0.015	(70,9), (80,8) (120,5), (196,3)
assign6	114	15	0.01	173	10	0.34	118	12	2.54	0.101	(118,12)
assign17	68	10	0.01	262	3	2.43	71	8	14.8	0.112	(71,8), (80,7), (130,4)
assign25	189	27	0.14	2004	11	7.12	189	27	41.2	0.142	(189,27), (452,14)
assign30	157	18	0.04	643	7	32.4	158	16	74.1	0.101	(158,16), (473,8)
assign45	6212	200	0.15	40937	54	85.0	6240	185	574	0.029	(6240,185), (7133,160) (9394,112), (12766,75)
assign75	8828	65	0.28	63860	36	122	9741	49	336	0.005	(9741,49)
assign100	305	6	0.58	661	3	34.3	310	3	85.6	0.009	(310,3)

Table 3. Sum of relatives gains when switching from PNF solutions to the optimal solutions of the classic AP and the BAP

Instance	**Sum of relative gains**	
	PNF vs classic AP	**PNF vs BAP**
assign3	0.214	0.178
assign4	1.357	0.000
assign6	0.216	0.299
assign17	0.207	2.065
assign25	0.000	9.010
assign30	0.118	2.507
assign45	0.076	4.852
assign75	0.232	5.290
assign100	0.983	1.132

Table 2 presents the numerical results in several instances with a range of dimensions of the cost matrix from 3×3 to 100×100. We also provide the PNF solution minimizing PQ and its corresponding value of α. We can see by the values of P and Q in this table that the PNF solution strikes a better trade-off between the total cost and the max-min distance compared with those for the classic AP and the BAP. In particular, when the solutions for the classical AP and the BAP are quite different: inefficient values of Q in the optimal solutions for the classic AP compared with those in the optimal solutions for the BAP and inefficient values of P in the optimal solutions for the BAP comparing with those in the optimal solutions for the classic AP, the PNF solution offers almost a better alternative than the solution for the classic AP (respectively for the BAP) with a significant drop on the values of Q (respectively P) and a slight growth on the values of P (respectively Q). More precisely, Table 3 presents the sums of relative gains when switching from PNF solutions to the optimal solutions for the classic AP and the BAP. Note that their values are not negative as we mentioned in (3), Section 3.2, and values further from 0 are preferable for the PNF solutions because they have then a much better trade-off between P and Q. Table 2 also indicates that we only have several NF solutions, and in most cases, the PNF solution is one of the extreme NF solutions having the smallest value of P or Q. One important issue is the CPU time for solving the NFAP (approximately for finding the PNF solution) is quite huge compared with the CPU time spent for solving the classic AP and the BAP. A deeper analysis of the iterations of Procedure $\mathit{Find}(\alpha)$ tells us that the CPU time spent for solving $P(\alpha)$ with a small value of α 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 LP given in Section 2.

5 Conclusion

In this paper, we have used 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 the *Assignment Problem* (AP). We have proven first the existence of *Nash Fairness* (NF) solutions for the AP. Second, we have designed an algorithm to find all NF solutions, including the PNF solutions minimizing the product of total cost and max-min distance. Numerical results conducted on instances of the AP have shown that compared with the optimal solutions for the BAP, the NF solutions found by our algorithm have almost much smaller total cost with a reasonable augmentation of the max-min distance and vice versa compared with the optimal solutions for the classic AP. An important notice is that this paper's results can also be applied to various balanced combinatorial optimization problems such as the balanced traveling salesman problem [12], the balanced spanning tree problem [10], etc. The future developments of our work are improving time complexity for Algorithm 1 by developing a special-purpose algorithm for solving $\mathcal{P}(\alpha)$. Moreover, we are also interested in finding a better upper bound for the number of NF solutions and generating the concept of NF solutions for bi-objective optimization problems with positive objective functions.

References

1. D. Bertsimas, V.F. Farias, N. Trichakis, The Price of Fairness, *Operations Research*, January-February 2011, 59(1), pp 17-31.
2. S. Martello, W.R. Pulleyblank, P. Toth, D. de Werra, Balanced optimization problems, *Operations Research Letters* Volume 3, Number 5, August 1984.
3. FP Kelly, AK Maullo, DKH Tan, Rate control for communication networks : shadow prices, proportional fairness and stability, *Journal of the Operational Research Society* 49(3), November 1997.
4. W. Ogryczak, Hanan Luss, Michab Pioro, Dritan Nace, Artur Tomaszewski, Fair Optimization and Networks: A survey, *Journal of Applied Mathematics*, *Hindawi Publishing Coporation* (2014). pp 1-26.
5. R. Timothy Marler, Jasbir Singh Arora, The weighted sum method for multi-objective optimization: New insights, *Structural and Multidisciplinary Optimization* 41(6), June 2010. pp 853-862.
6. I. Heller, C.B. Tompkins, An Extension of a Theorem of Dantzig's, *Annals of Mathematics Studies*, *Princeton University Press* (1956). 247-254(38).
7. H. W. Kuhn, The Hungarian method for assignment problem, *Naval Research Logistics Quarterly* 2 (1–2): 83–97, March 1955.
8. W. Dinkelbach, On nonlinear fractional programming, *Management Sci*, (1967) 492-498(13).
9. J.R. Isbell, W.H. Marlow, Attrition games, *Naval Res. Logist. Quart 3*, (1956), pp 71-94.
10. Martello, Most and least uniform spanning trees, *Discrete Applied Mathematics* 15(2) 1986, pp 181-197.
11. J.E. Beasley, Linear programming on Cray supercomputer, *Journal of the Operational Research Society*, 41 (1990), pp 133-139.
12. M.H. Nguyen, Mourad Baiou, V.H. Nguyen and T.Q.T. Vo, Nash fairness solutions for balanced TSP, *Proceeding of International Network Optimization Conference, June 2022 (INOC2022).*

APPENDIX

Theorem 3. [12] *$(P^*, Q^*) \in \mathcal{S}$ is a NF solution if and only if (P^*, Q^*) is an optimal solution of $\mathcal{P}(\alpha^*)$ where $\alpha^* = \frac{Q^*}{P^*}$.*

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

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

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

Since $\alpha^* = \frac{Q^*}{P^*}$, we have $\alpha^* P^* + Q^* = 2Q^*$.
Dividing two sides of (6) by $P^* > 0$ we obtain

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

So we deduce from (7)

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

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

$\Longleftarrow$ Now suppose $\alpha^* = \frac{Q^*}{P^*}$ and (P^*, Q^*) is an optimal solution of $\mathcal{P}(\alpha^*)$, we show that (P^*, Q^*) is a NF solution.

If (P^*, Q^*) is not a NF solution, there exists a solution $(P', Q') \in \mathcal{S}$ 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^*). □

Lemma 3. [12] *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 $\alpha < \alpha'$ 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 $\alpha < \alpha'$, 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. [12] *During the execution of Procedure Find(α_0) in Algorithm 1, $\alpha_{i+1} \in]0,1[$ and $T_i \leq 0, \forall i \geq 0$. Moreover, if $\alpha_0 P_0 - Q_0 > 0$ then the sequence $\{\alpha_i\}_{i\geq 0}$ is strictly decreasing. Otherwise, if $\alpha_0 P_0 - Q_0 < 0$ then the sequence $\{\alpha_i\}_{i\geq 0}$ is strictly increasing.*

Proof. Since $P > Q > 0$, $\forall (P,Q) \in \mathcal{S}$, it follows that $\alpha_{i+1} = \frac{Q_i}{P_i} \in]0,1[$, $\forall i \geq 0$.

The optimality of (P_{i+1}, Q_{i+1}) gives

$$\alpha_{i+1} P_{i+1} + Q_{i+1} \leq \alpha_{i+1} P_i + Q_i,$$

Thus, $T_i := \alpha_{i+1} P_{i+1} + Q_{i+1} - \alpha_{i+1} P_i - Q_i \leq 0, \forall i \geq 0$.

We first consider $\alpha_0 P_0 - Q_0 > 0$. We proof $\alpha_i > \alpha_{i+1}$, $\forall i \geq 0$ by induction on i. For $i = 0$, we have $\alpha_0 > Q_0/P_0 = \alpha_1$. Suppose that our hypothesis is true until $i = k \geq 0$, we will prove that it is also true with $i = k+1$.

The inductive hypothesis gives $\alpha_k > \alpha_{k+1}$ that implies $P_{k+1} \geq P_k > 0$ and $Q_k \geq Q_{k+1} > 0$ according to Lemma 3. It leads to $Q_k P_{k+1} - P_k Q_{k+1} \geq 0$ and $Q_k P_{k+1} - P_k Q_{k+1} = 0 \iff (P_i, Q_i) \equiv (P_{i+1}, Q_{i+1})$.

If $T_i = 0$ then Procedure Find(α_0) returns the value α_{k+1}. In this case, $\{\alpha_i\}_{0\leq i\leq k+1}$ is strictly decreasing.

If $T_i \neq 0$ then $(P_i, Q_i) \not\equiv (P_{i+1}, Q_{i+1})$ that leads to $Q_k P_{k+1} - P_k Q_{k+1} > 0$. We get

$$\alpha_{k+1} - \alpha_{k+2} = \frac{Q_k}{P_k} - \frac{Q_{k+1}}{P_{k+1}} = \frac{Q_k P_{k+1} - P_k Q_{k+1}}{P_k P_{k+1}} > 0,$$

Thus, in this case our hypothesis is also true with $i = k+1$. Consequently, $\{\alpha_i\}$ is strictly decreasing, $\forall i \geq 0$.

Similarly, if $\alpha_0 P_0 - Q_0 < 0$ we obtain that the sequence $\{\alpha_i\}_{i\geq 0}$ is strictly increasing. That concludes the proof. □

Lemma 5. [12] *From each $\alpha_0 \in [0,1]$, Procedure Find(α_0) converges to a coefficient $\alpha_k \in \mathcal{C}_0$ satisfying α_k is the unique element $\in \mathcal{C}_0$ between α_0 and α_k.*

Proof. If $\alpha_0 P_0 - Q_0 = 0$ then Procedure *Find*(α_0) returns the value α_0 that leads to the conclusion. Without loss of generality, we suppose that $\alpha_0 P_0 - Q_0 > 0$ and Procedure *Find*(α_0) converges to a coefficient $\alpha_{n+1} \in]0,1[, \forall \alpha_0 \in [0,1]$ where $n \geq 0$. Due to Lemma 4, $\{\alpha_i\}_{i\geq 0}$ is strictly decreasing.

By the stopping criteria of Procedure *Find*(α_0), when $T_n = \alpha_{n+1} P_{n+1} + Q_{n+1} - \alpha_{n+1} P_n - Q_n = 0$ we obtain $\alpha_{n+1} \in C_0$ and (P_n, Q_n) is a NF solution (Proposition 1). If $n = 0$ then (P_0, Q_0) is a NF solution which is an optimal solution of both $\mathcal{P}(\alpha_0)$ and $\mathcal{P}(\alpha_1)$. Due to Lemma 3, for all $\alpha \in (\alpha_1, \alpha_0)$, (P_0, Q_0) is the optimal solution of $\mathcal{P}(\alpha)$ that leads to the conclusion.

We consider $n \geq 1$. We have $T_n = 0$ and $T_i < 0$, $\forall 0 \leq i \leq n-1$.

Suppose that there exists $\alpha^* = (\alpha_{n+1}, \alpha_0) \cap \mathcal{C}_0$. According to Theorem 3, there exists a NF solution (P^*, Q^*) which is an optimal solution of $\mathcal{P}(\alpha^*)$ and $\alpha^* P^* = Q^*$. Since the sequence $\{\alpha\}$ is strictly decreasing, there exists $0 \leq i \leq n$ such that $\alpha^* \in [\alpha_{i+1}, \alpha_i)$.

We first show that $\alpha^* \neq \alpha_{i+1}$ by contradiction. Let assume that $\alpha^* = \alpha_{i+1}$. Since $\alpha^* > \alpha_{n+1}$, we have $i \leq n-1$. It leads to $\alpha_{i+1}P^* = Q^*$. Thus, we get

$$\frac{Q^*}{P^*} = \alpha_{i+1} = \frac{Q_i}{P_i} \implies \frac{P^*}{P_i} = \frac{Q^*}{Q_i},$$

If $i = 0$ then $\frac{P^*}{P_0} = \frac{Q^*}{Q_0}$. Since $\alpha_1 < \alpha_0$, we have $P^* \geq P_0$ that implies $Q^* \geq Q_0$. Thus, $\alpha_1 P^* + Q^* \geq \alpha_1 P_0 + Q_0$.

If $i \geq 1$ then $P^* = P_i, Q^* = Q_i$ since they are both Pareto-optimal solutions and $\frac{P^*}{P_i} = \frac{Q^*}{Q_i}$. Thus, $\alpha_{i+1}P^* + Q^* = \alpha_{i+1}P_i + Q_i$.

Consequently, we always have $\alpha_{i+1}P^* + Q^* \geq \alpha_{i+1}P_i + Q_i$. Moreover, $\alpha_{i+1}P_{i+1} + Q_{i+1} = \alpha_{i+1}P^* + Q^*$ because both (P_{i+1}, Q_{i+1}) and (P^*, Q^*) are the optimal solutions of $\mathcal{P}(\alpha_{i+1})$. Thus, $T_i = \alpha_{i+1}P_{i+1} + Q_{i+1} - \alpha_{i+1}P_i - Q_i \geq 0$ which leads to a contradiction due to $T_i < 0$, $\forall 0 \leq i \leq n-1$. Hence, we have $\alpha^* \neq \alpha_{i+1}$.

It follows that $\alpha^* \in (\alpha_{i+1}, \alpha_i)$. Since $\alpha^* < \alpha_i$, we have $P^* \geq P_i$ and $Q^* \leq Q_i$ due to Lemma 3. Thus, we get

$$\alpha^* = \frac{Q^*}{P^*} \leq \frac{Q_i}{P_i} = \alpha_{i+1},$$

which leads to a contradiction due to the fact that $\alpha^* > \alpha_{i+1}$.

Consequently, Procedure $\mathit{Find}(\alpha_0)$ converges to a coefficient $\alpha_k \in \mathcal{C}_0$ satisfying α_k is the unique element $\in \mathcal{C}_0$ between α_0 and α_k. □