Proportional Fairness for Combinatorial Optimization

Minh Hieu Nguyen, Mourad Baiou, Viet Hung Nguyen, Thi Quynh Trang Vo

Proportional Fairness for Combinatorial Optimization

Minh Hieu Nguyen[1], Mourad Baiou[1], Viet Hung Nguyen[1], and Thi Quynh Trang Vo[1]

INP Clermont Auvergne, Univ Clermont Auvergne, Mines Saint-Etienne, CNRS, UMR 6158 LIMOS, 1 Rue de la Chebarde, Aubiere Cedex, France
{minh_hieu.nguyen,mourad.baiou,viet_hung.nguyen,thi_quynh_trang.vo}@uca.fr

Abstract. Proportional fairness (PF) is a concept widely studied in the literature, particularly in the fields of telecommunications, network design, resource allocation, and social choice. It aims to distribute utilities in a way that ensures fairness and equity among users while optimizing system performance. Under convexity, PF is equivalent to the *Nash bargaining solution*, a well-known notion from cooperative game theory and it can be obtained by maximizing the product of the utilities. In contrast, when dealing with non-convex optimization, PF is not guaranteed to exist and when exists, it is also the Nash bargaining solution. Consequently, finding PF under non-convexity remains challenging since it is not equivalent to any known optimization problem.
This paper deals with PF in the context of combinatorial optimization where the feasible set is discrete, finite, and non-convex. For this purpose, we consider a general *Max-Max Bi-Objective Combinatorial Optimization* (Max-Max BOCO) problem where its two objectives to be simultaneously maximized take only positive values. Then, we seek to find the solution to achieving PF between two objectives which is referred to as *proportional fair solution* (PF solution).
We first show that the PF solution, when exists, can be obtained by maximizing a suitable linear combination of two objectives. Subsequently, our main contribution lies in presenting an exact algorithm that converges within a logarithmic number of iterations to determine the PF solution. Finally, we provide computational results conducted on the bi-objective spanning tree problem which is a specific example of Max-Max BOCO.

Keywords: Bi-Objective Combinatorial Optimization · Nash Bargaining Solution · Proportional Fairness · Binary Search Algorithm · Bi-Objective Spanning Tree Problem

1 Introduction

Proportional fairness (PF) is a concept widely studied in the literature, particularly in the fields of telecommunications, network design, resource allocation, and social choice. The goal of PF is to provide a compromise between the *utilitarian rule* - which emphasizes overall system efficiency, and the *egalitarian rule*

- which emphasizes individual fairness. For example, in wireless communication systems, PF is often used in the allocation of transmission power, bandwidth, and data rates among mobile users to maximize overall system throughput while ensuring fair access for all users [1]. In network scheduling and traffic management, PF plays a significant role in packet scheduling algorithms. It helps ensure that packets from different flows or users are treated fairly, preventing any single user from dominating network resources [2]. In resource allocation, PF principles are applied to prevent resource starvation and improve system efficiency by dynamically distributing resources based on the individual demands of users, thereby enhancing overall system performance and ensuring a fair and balanced utilization of available resources [3].

It has been demonstrated that if the feasible set is convex, PF is equivalent to the *Nash bargaining solution* which always exists and can be obtained by maximizing the product of the objectives (or equivalently, by maximizing a logarithmic sum problem) [1], [4], [5]. In this case, finding the optimal solution is computationally expensive since the objective remains nonlinear. Thus, in practice, heuristic algorithms are often employed to find approximate solutions that achieve near-PF in some special scenarios (see, e.g., [2], [7]). In contrast, when dealing with non-convex optimization, the existence of PF is not guaranteed and if it exists, it is also the Nash bargaining solution [1], [9] . To address such a case, popular approaches involve considering certain non-convex sets, which are convex after a logarithmic transformation [9]. An alternative approach is to introduce the concept of *local proportional fairness*, which is always achievable, and then analyze its properties [8].

This paper deals with PF in the context of combinatorial optimization where the feasible set is discrete, finite, and non-convex. For this purpose, we consider a general *Max-Max Bi-Objective Combinatorial Optimization* (Max-Max BOCO) problem where its two objectives to be simultaneously maximized take only positive values. Then, we seek to find the solution achieving PF between two objectives which is referred to as *proportional fair solution* (PF solution).

Notice that some approaches using PF as a criterion for solving the bi-objective minimization problems have been introduced in [10], [11], [12]. In these scenarios, the solution achieved PF, which is referred to as the Nash Fairness (NF) solution, always exists and can be obtained by an iterative algorithm based on the application of the Newton-Raphson method. The NF solution is also generalized in the context of bi-objective discrete optimization where the objectives can be either maximized or minimized. As a result, the PF solution described in this paper represents a specific instance of the NF solution for Max-Max BOCO, as mentioned in [13]. However, it is important to note that in [13], the authors primarily introduced the concept of NF solution and focused on calculating the NF solution set for the cases when it always exists and there may be many NF solutions (i.e., for Max-Min BOCO and Min-Min BOCO). Generally, determining the PF solution in the context of combinatorial optimization remains a challenging question, especially when it is not guaranteed to exist and cannot be directly obtained through the application of the Newton-Raphson method.

In this paper, we first show that the PF solution, when exists, can be found by maximizing a suitable linear combination of two objectives. Then, our main contribution lies in presenting an exact algorithm that converges within a logarithmic number of iterations to determine efficiently the PF solution for Max-Max BOCO. Finally, we provide computational results conducted on the Bi-Objective Spanning Tree Problem (BOSTP) which is a specific example of Max-Max BOCO.

The paper is organized as follows. In Section 2, we discuss the characterization of the PF solution for Max-Max BOCO. In Section 3, we propose an exact algorithm for finding the PF solution. Computational results on some instances of the BOSTP will be presented in Section 4. Finally, in Section 5, we give some conclusions and future works.

2 Characterization of the PF solution for Max-Max BOCO

Since the objectives of Max-Max BOCO is to be simultaneously maximized, it can be formulated as

$$\max_{x \in \mathcal{X}} (P(x), Q(x)),$$

where $\mathcal{X}$ denotes the set of all feasible decision vectors x. For Max-Max BOCO, we suppose that $\mathcal{X}$ is a finite set and $P(x), Q(x) > 0, \forall x \in \mathcal{X}$.

Let $(P, Q) = (P(x), Q(x))$ denote the objective values corresponding to a decision vector $x \in \mathcal{X}$. Let $\mathcal{S}$ represent the set of pairs (P, Q) corresponding to all feasible decision vector solutions. Throughout this paper, we will characterize the feasible solutions for Max-Max BOCO using pairs (P, Q) instead of explicitly listing the decision vector solutions. Thus, two feasible solutions having the same values of (P, Q) will be considered equivalent. Throughout this paper, we use the notation "$\equiv$" to denote equivalent solutions. Since $\mathcal{X}$ is finite, the number of feasible solutions is also finite, implying that $\mathcal{S}$ is a finite set. According to the definition of proportional fairness (see, e.g., [1], [4]), $(P^{PF}, Q^{PF}) \in \mathcal{S}$ is the PF solution provided that

$$\frac{P - P^{PF}}{P^{PF}} + \frac{Q - Q^{PF}}{Q^{PF}} \leq 0 \iff \frac{P}{P^{PF}} + \frac{Q}{Q^{PF}} \leq 2, \ \forall (P, Q) \in \mathcal{S}, \qquad (1)$$

Since the feasible set of Max-Max BOCO is discrete and non-convex, finding the PF solution for Max-Max BOCO remains challenging since it does not always exist and consequently, there is not any known optimization problem that is equivalent to the condition (1).

In the following, we show that the PF solution, when exists, can be obtained by maximizing a suitable linear combination of P and Q by considering the optimization problem:

$$\mathcal{F}(\alpha) = \max_{(P,Q) \in \mathcal{S}} f_\alpha(P, Q) := P + \alpha Q,$$

where $\alpha \geq 0$ is an appropriate coefficient to be determined.

Notice that we assume the existence of the algorithms for maximizing the linear combinations of P and Q, including for maximizing P and maximizing Q.

Theorem 1. *[13]* $(P^{PF}, Q^{PF}) \in \mathcal{S}$ *is the PF solution if and only if* (P^{PF}, Q^{PF}) *is a solution of* $\mathcal{F}(\alpha^{PF})$ *with* $\alpha^{PF} = P^{PF}/Q^{PF}$.

Proof. See Appendix. □

Notice that the generalized version of Theorem 1 for the NF solution has been presented in [13]. As Theorem 1 provides a necessary and sufficient condition for the PF solution, the main question now is how to propose an exact algorithm for determining the PF solution based on Theorem 1. We give the answer to this question in the next section.

3 Algorithm for determining the PF solution

3.1 Algorithm construction

In this section, we outline the idea of constructing an exact algorithm to determine the PF solution for Max-Max BOCO.

For a given $\alpha_k \geq 0$, let $T(\alpha_k) := P_k - \alpha_k Q_k$ where (P_k, Q_k) is a solution of $\mathcal{F}(\alpha_k)$. Throughout this paper, let (P^{PF}, Q^{PF}) denote the PF solution and α^{PF} denote the PF coefficient, a coefficient such that (P^{PF}, Q^{PF}) is a solution of $\mathcal{F}(\alpha^{PF})$ and $P^{PF} = \alpha^{PF} Q^{PF}$. We have $T(\alpha^{PF}) = P^{PF} - \alpha^{PF} Q^{PF} = 0$. According to Theorem 1 and the uniqueness of the PF solution when exists, determining the PF solution is equivalent to determining the PF coefficient α^{PF}.

We first show the monotonic relationship between $\alpha \geq 0$ and the solution of $\mathcal{F}(\alpha)$ with respect to the values of P and Q. As a consequence, we also deduce the monotonic relationship between α and $T(\alpha)$.

Lemma 1. *Given* $0 \leq \alpha' < \alpha''$ *and let* (P', Q'), $(P'', Q'') \in \mathcal{S}$ *be respectively the solutions of* $\mathcal{F}(\alpha')$ *and* $\mathcal{F}(\alpha'')$. *Then* $P' \geq P''$ *and* $Q' \leq Q''$. *Moreover,* $T(\alpha') > T(\alpha'')$.

Proof. The optimality of (P', Q') and (P'', Q'') gives

$$P' + \alpha' Q' \geq P'' + \alpha' Q'', \text{ and} \tag{2a}$$

$$P'' + \alpha'' Q'' \geq P' + \alpha'' Q' \tag{2b}$$

Adding (2a) and (2b) gives $(\alpha' - \alpha'')(Q' - Q'') \geq 0$. Since $\alpha' < \alpha''$, $Q' \leq Q''$.

On the other hand, the inequality (2a) implies $P' - P'' \geq \alpha'(Q'' - Q') \geq 0$.

Since $P' \geq P''$, $Q' \leq Q''$ and $\alpha' < \alpha''$, we obtain $T(\alpha') = P' - \alpha' Q' \geq P'' - \alpha' Q'' > P'' - \alpha'' Q'' = T(\alpha'')$. □

As a result of Lemma 1, if $\alpha' < \alpha''$ and (P', Q') is the solution of both $\mathcal{F}(\alpha')$ and $\mathcal{F}(\alpha'')$ then (P', Q') is the solution of $\mathcal{F}(\alpha)$ for all $\alpha' < \alpha < \alpha''$. Moreover, based on the monotonic relationship between α and $T(\alpha)$, for given $0 \leq \alpha_i < \alpha_j$ and $T(\alpha_i)T(\alpha_j) > 0$, we have $\alpha^{PF} \notin (\alpha_i, \alpha_j)$ because for $\alpha' \in (\alpha_i, \alpha_j)$ and an arbitrary solution (P', Q') of $\mathcal{F}(\alpha')$, $T(\alpha')$ has the same sign as $T(\alpha_i)$ and $T(\alpha_j)$ which implies $T(\alpha') \neq 0$ and then $\alpha' \neq \alpha^{PF}$.

Let α^{sup} be an upper bound of α^{PF} such that $\alpha^{PF} < \alpha^{sup}$ (we will provide a detailed definition for α^{sup} in our algorithm). According to the results of Theorem 1 and Lemma 1, the main idea of our algorithm is based on the binary search algorithm in the interval $[0, \alpha^{sup}]$. More precisely, we use Procedure *SEARCH*() to identify the PF solution and the PF coefficient α^{PF} in such interval, ensuring that $T(\alpha^{PF}) = 0$. Starting from an interval $[\alpha_i, \alpha_j] \subseteq [0, \alpha^{sup}]$ with $T(\alpha_i) > 0$ and $T(\alpha_j) < 0$, Procedure *SEARCH*() selects α_s as the midpoint of the interval $[\alpha_i, \alpha_j]$ and solve $\mathcal{F}(\alpha_s)$ to obtain a solution (P_s, Q_s). Then, we use respectively Procedure *Verify_PF_sol*() and Procedure *Verify_PF_coeff*() to verify whether (P_s, Q_s) is the PF solution and whether α_s is the PF coefficient. If the verification is unsuccessful, the half-interval in which the PF coefficient cannot exist is eliminated and we retain only one half-interval for further exploration within Procedure *SEARCH*(). The choice is made between $[\alpha_i, \alpha_s]$ and $[\alpha_s, \alpha_j]$, depending on the sign of $T(\alpha_s)$. We continue these steps until we obtain an interval with a length smaller than a positive parameter ϵ defined by the input of the Max-Max BOCO problem. The selection method of ϵ guarantees the absence of the PF coefficient in such an interval. Consequently, our algorithm always converges in a logarithmic number of iterations in terms of ϵ and α^{sup}.

In the following, we present the statement and the proofs for our algorithm.

3.2 Algorithm statement and proofs

In this section, we first introduce Procedure *Verify_PF_sol*(α_0, P_0, Q_0) to verify whether a solution (P_0, Q_0) of $\mathcal{F}(\alpha_0)$ is the PF solution. The proof of this procedure will be stated in the next lemma.

Algorithm 1 Verify whether a solution (P_0, Q_0) of $\mathcal{F}(\alpha_0)$ is the PF solution

Input: $\alpha_0 \geq 0$, $(P_0, Q_0) \in \mathcal{S}$ is a solution of $\mathcal{F}(\alpha_0)$.
Output: True if (P_0, Q_0) is the PF solution or False otherwise.

procedure *Verify_PF_sol*(α_0, P_0, Q_0)
  if $P_0 - \alpha_0 Q_0 = 0$ **then** return True
  else
    $\alpha' \leftarrow P_0/Q_0$
    solving $\mathcal{F}(\alpha')$ to obtain the solution (P', Q').
    if $f_{\alpha'}(P', Q') = f_{\alpha'}(P_0, Q_0)$ **then** return True
    else return False
    end if
  end if
end procedure

Lemma 2. *Given $\alpha_0 \geq 0$ and $(P_0, Q_0) \in \mathcal{S}$ as a solution of $\mathcal{F}(\alpha_0)$. Let $\alpha' = P_0/Q_0$ and (P', Q') be a solution of $\mathcal{F}(\alpha')$. If $T(\alpha_0) = P_0 - \alpha_0 Q_0 \neq 0$ then (P_0, Q_0) is the PF solution if and only if $f_{\alpha'}(P', Q') = f_{\alpha'}(P_0, Q_0)$.*

Proof. $\Longrightarrow$ If (P_0, Q_0) is the PF solution then (P_0, Q_0) is also a solution of $\mathcal{F}(\alpha')$ due to Theorem 1. Thus, $f_{\alpha'}(P', Q') = f_{\alpha'}(P_0, Q_0)$.

$\Longleftarrow$ If $f_{\alpha'}(P', Q') = f_{\alpha'}(P_0, Q_0)$ then (P_0, Q_0) is also a solution of $\mathcal{F}(\alpha')$. Since $\alpha' = P_0/Q_0$, (P_0, Q_0) is the PF solution due to Theorem 1. □

Then, from a given $\alpha_0 \geq 0$ and a solution (P_0, Q_0) of $\mathcal{F}(\alpha_0)$, we discuss how to construct Procedure *Verify_PF_coeff*(α_0, P_0, Q_0) which aims to verify whether α_0 is the PF coefficient and return the PF solution if the verification is successful. It is important to remind that if $T(\alpha_0) = P_0 - \alpha_0 Q_0 = 0$ then $\alpha_0 = \alpha^{PF}$ and (P_0, Q_0) is necessarily the PF solution due to Theorem 1. However, if $T(\alpha_0) \neq 0$, we may not assert that $\alpha_0 \neq \alpha^{PF}$ as well as (P_0, Q_0) is not the PF solution. In general, although the PF solution is necessary a solution of $\mathcal{F}(\alpha^{PF})$, we might not obtain the PF solution by solving $\mathcal{F}(\alpha^{PF})$. The fact is that the problem $\mathcal{F}(\alpha^{PF})$ may have multiple solutions, and we obtain one solution, which might not be the PF solution. More precisely, we state the following proposition.

Proposition 1. *We might not obtain the PF solution by solving $\mathcal{F}(\alpha)$, $\forall \alpha \geq 0$.*

Proof. To prove this conclusion, we consider an example of the Bi-Objective Spanning Tree Problem (BOSTP) which is also a Max-Max BOCO problem. Let G be an undirected, connected graph, and each edge of G is associated with two positive values: profit and reliability. The BOSTP consists of finding a spanning tree of G maximizing both the total profit and the minimum edge reliability.

This example of BOSTP with two values on each edge is illustrated in Figure 1. For example, the profit and reliability associated with edge (14) are respectively 20 and 9.

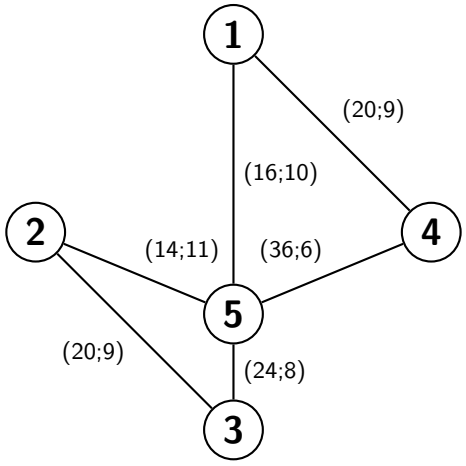

Fig. 1: An example of the BOSTP

Let (P,Q) denote the solution for the total profit and the minimum edge reliability corresponding to a spanning tree solution. We show each distinct spanning tree by listing its edges and the corresponding solution (P, Q) as follows.

- (14) (15) (23) (25) and $(P_1, Q_1) = (70, 9)$
- (14) (15) (23) (35) and $(P_2, Q_2) = (80, 8)$
- (14) (15) (25) (35) and $(P_3, Q_3) = (74, 8)$
- (14) (45) (23) (25) and $(P_4, Q_4) = (90, 6)$
- (14) (45) (23) (35) and $(P_5, Q_5) = (100, 6)$
- (14) (45) (25) (35) and $(P_6, Q_6) = (94, 6)$
- (15) (45) (23) (25) and $(P_7, Q_7) = (86, 6)$
- (15) (45) (23) (35) and $(P_8, Q_8) = (96, 6)$
- (15) (45) (25) (35) and $(P_9, Q_9) = (90, 6)$

Then, we can easily verify that (P_2, Q_2) is the PF solution since

$$\frac{P_i}{P_2} + \frac{Q_i}{Q_2} \leq 2, \forall 1 \leq i \leq 9,$$

Thus, we get $\alpha^{PF} = P_2/Q_2 = 10$. However, if $0 \leq \alpha < 10$ (resp. $\alpha > 10$) then (P_5, Q_5) (resp. (P_1, Q_1)) is the solution of $\mathcal{F}(\alpha)$ and if $\alpha = 10$, solving $\mathcal{F}(\alpha)$ may return (P_1, Q_1) or (P_5, Q_5) instead of the PF solution (P_2, Q_2) because they are simultaneously the solutions of $\mathcal{F}(10)$ due to $P_1 + 10Q_1 = P_2 + 10Q_2 = P_5 + 10Q_5 = 160$. In this case, $T(10) \neq 0$ despite $\alpha^{PF} = 10$.

Generally, if $\mathcal{F}(\alpha^{PF})$ has multiple (distinct) solutions including the PF solution, we might not obtain the PF solution by solving $\mathcal{F}(\alpha)$. □

For Procedure *Verify_PF_coeff*(), we present the following optimization problem

$$\mathcal{G}(\alpha) = \max_{(P,Q)\in\mathcal{S}} g_\alpha(P, Q) := P + \alpha Q - |P - \alpha Q|,$$

where $|.|$ denotes the absolute function.

Algorithm 2 Verify whether α_0 is the PF coefficient

Input: $\alpha_0 \geq 0$, (P_0, Q_0) is a solution of $\mathcal{F}(\alpha_0)$ and $T(\alpha_0) = P_0 - \alpha_0 Q_0 \neq 0$.
Output: The PF solution if α_0 is the PF coefficient or (Null, Null) otherwise.

```
1: procedure Verify_PF_coeff(α0, P0, Q0)
2:     solving G(α0) to obtain the solutions (P1, Q1)
3:     if g_α0(P1, Q1) = f_α0(P0, Q0) then return (P1, Q1)
4:     else return (Null, Null)
5:     end if
6: end procedure
```

Lemma 3. *For a given $\alpha_0 \geq 0$, let $(P_0, Q_0), (P_1, Q_1) \in \mathcal{S}$ be respectively the solutions of $\mathcal{F}(\alpha_0)$ and $\mathcal{G}(\alpha_0)$. If $T(\alpha_0) = P_0 - \alpha_0 Q_0 \neq 0$ then α_0 is the PF coefficient if and only if $g_{\alpha_0}(P_1, Q_1) = f_{\alpha_0}(P_0, Q_0)$.*

Proof. $\Longrightarrow$ Suppose that $\alpha_0 = \alpha^{PF}$. According to Theorem 1, there exists the PF solution $(P^{PF}, Q^{PF}) \in \mathcal{S}$ such that (P^{PF}, Q^{PF}) is a solution of $\mathcal{F}(\alpha_0)$ and $P^{PF} = \alpha_0 Q^{PF}$. Since both (P_0, Q_0) and (P^{PF}, Q^{PF}) are the solutions of $\mathcal{F}(\alpha_0)$ and $P^{PF} - \alpha_0 Q^{PF} = 0$, we have

$$P_0 + \alpha_0 Q_0 = P^{PF} + \alpha_0 Q^{PF} - |P^{PF} - \alpha_0 Q^{PF}|,$$

The optimality of (P^{PF}, Q^{PF}) gives

$$P^{PF} + \alpha_0 Q^{PF} \geq P_1 + \alpha_0 Q_1,$$

Since $|P_1 - \alpha_0 Q_1| \geq 0$, we deduce $P^{PF} + \alpha_0 Q^{PF} \geq P_1 + \alpha_0 Q_1 - |P_1 - \alpha_0 Q_1|$. Thus,

$$P^{PF} + \alpha_0 Q^{PF} - |P^{PF} - \alpha_0 Q^{PF}| \geq P_1 + \alpha_0 Q_1 - |P_1 - \alpha_0 Q_1|, \tag{3}$$

Since (P_1, Q_1) is a solution of $\mathcal{G}(\alpha_0)$, we have

$$P_1 + \alpha_0 Q_1 - |P_1 - \alpha_0 Q_1| \geq P^{PF} + \alpha_0 Q^{PF} - |P^{PF} - \alpha_0 Q^{PF}|, \tag{4}$$

From (3) and (4), we get

$$P_1 + \alpha_0 Q_1 - |P_1 - \alpha_0 Q_1| = P^{PF} + \alpha_0 Q^{PF} - |P^{PF} - \alpha_0 Q^{PF}| = P_0 + \alpha_0 Q_0,$$

which implies $g_{\alpha_0}(P_1, Q_1) = f_{\alpha_0}(P_0, Q_0)$.

$\Longleftarrow$ Suppose that $g_{\alpha_0}(P_1, Q_1) = f_{\alpha_0}(P_0, Q_0)$. We obtain $P_1 + \alpha_0 Q_1 - |P_1 - \alpha_0 Q_1| = P_0 + \alpha_0 Q_0$. Since $P_1 + \alpha_0 Q_1 - |P_1 - \alpha_0 Q_1| \leq P_1 + \alpha_0 Q_1 \leq P_0 + \alpha_0 Q_0$, we must have $|P_1 - \alpha_0 Q_1| = 0$ and $P_1 + \alpha_0 Q_1 = P_0 + \alpha_0 Q_0$. Consequently, (P_1, Q_1) is a solution of $\mathcal{F}(\alpha_0)$ and $P_1 = \alpha_0 Q_1$. Thus, (P_1, Q_1) is the PF solution and $\alpha_0 = \alpha^{PF}$ due to Theorem 1. □

Notice that we can obtain the PF solution by solving $\mathcal{F}(10)$ and $\mathcal{G}(10)$ for the instance of the BOSTP mentioned in Proposition 1. Subsequently, from $0 \leq \alpha_i < \alpha_j$ and (P_i, Q_i), (P_j, Q_j) as the solutions of $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$, we present Procedure $SEARCH(\alpha_i, P_i, Q_i, \alpha_j, P_j, Q_j, \epsilon)$ for determining the PF solution where the PF coefficient α^{PF} is in the interval $[\alpha_i, \alpha_j]$. We recall that the parameter ϵ is presented for the stopping condition of Procedure $SEARCH()$ as mentioned in Section 3.1.

For Max-Max BOCO, ϵ can be determined as

$$\epsilon = min\ \{|\frac{P' - P''}{Q'' - Q'} - \frac{P'' - P'''}{Q''' - Q''}|\} \tag{5}$$

where $|.|$ denotes the absolute function, $(P', Q'), (P'', Q''), (P''', Q''') \in \mathcal{S}$ are respectively the solutions of $\mathcal{F}(\alpha')$, $\mathcal{F}(\alpha'')$, $\mathcal{F}(\alpha''')$ for which $0 \leq \alpha' < \alpha'' < \alpha'''$, $P' \geq P'' \geq P''', Q' > Q'' > Q'''$ and $\frac{P'-P''}{Q''-Q'} \neq \frac{P''-P'''}{Q'''-Q''}$.

Algorithm 3 Determine the PF solution where the corresponding PF coefficient is in the interval $[\alpha_i, \alpha_j]$

Input: (α_i, P_i, Q_i) and (α_j, P_j, Q_j) satisfying the following conditions:
- $0 \leq \alpha_i < \alpha_j$ such that α_i, α_j are not PF coefficients.
- (P_i, Q_i) and (P_j, Q_j) are respectively solutions of $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$.
- $(P_i, Q_i) \not\equiv (P_j, Q_j)$, (P_i, Q_i) and (P_j, Q_j) are not PF solutions.
- a parameter ϵ as defined in (5).

Output: The PF solution if it exists or Null otherwise.

procedure *SEARCH*$(\alpha_i, P_i, Q_i, \alpha_j, P_j, Q_j, \epsilon)$
  $\alpha_k \leftarrow \frac{P_i - P_j}{Q_j - Q_i}$
  if $\alpha_k = \alpha_i$ **or** $\alpha_k = \alpha_j$ **then** return Null
  end if
  solving $\mathcal{F}(\alpha_k)$ to obtain a solution (P_k, Q_k)
  if *Verify_PF_sol*$(\alpha_k, P_k, Q_k) ==$ True **then** return (P_k, Q_k)
  end if
  $(P', Q') \leftarrow$ *Verify_PF_coeff*(α_k, P_k, Q_k)
  if $(P', Q') \not\equiv$ (Null, Null) **then** return (P', Q')
  else if $(P_k, Q_k) \equiv (P_i, Q_i)$ **or** $(P_k, Q_k) \equiv (P_j, Q_j)$ **then** return Null
  else
    if $\alpha_j - \alpha_i \geq \epsilon$ **then**
      $\alpha_s \leftarrow \frac{\alpha_i + \alpha_j}{2}$
      solving $\mathcal{F}(\alpha_s)$ to obtain a solution (P_s, Q_s)
      if *Verify_PF_sol*$(\alpha_s, P_s, Q_s) ==$ True **then** return (P_s, Q_s)
      end if
      $(P'', Q'') \leftarrow$ *Verify_PF_coeff*(α_s, P_s, Q_s)
      if $(P'', Q'') \not\equiv$ (Null, Null) **then** return (P'', Q'')
      else
        $T(\alpha_s) \leftarrow P_s - \alpha_s Q_s$
        if $T(\alpha_s) > 0$ **then** return *SEARCH*$(\alpha_s, P_s, Q_s, \alpha_j, P_j, Q_j, \epsilon)$
        else if $T(\alpha_s) < 0$ **then** return *SEARCH*$(\alpha_i, P_i, Q_i, \alpha_s, P_s, Q_s, \epsilon)$
        end if
      end if
    else return Null
    end if
  end if
end procedure

It is necessary to select α_k before selecting the midpoint α_s at each iteration. The fact is that we might not obtain the PF solution by only selecting the midpoints of the intervals. For example, in the instance of the BOSTP mentioned in Proposition 1, the PF solution can only be obtained by solving $\mathcal{F}(\alpha)$ and $\mathcal{G}(\alpha)$

with $\alpha = 10$. However, by repeating choosing the midpoints of the intervals, we might not reach $\alpha_s = 10$ (in contrast, for α_k, we can obtain $\alpha_k = 10$).

By the following lemma, we show that our choosing method for α_k at each iteration and the parameter ϵ offer some specific criteria to promptly verify the existence of α^{PF} in the interval $[\alpha_i, \alpha_j]$.

Lemma 4. *Let $[\alpha_i, \alpha_j]$ be an interval defined by $0 \leq \alpha_i < \alpha_j$, α_i, α_j are not PF coefficients. Let $(P_i, Q_i) \not\equiv (P_j, Q_j)$ be respectively the solutions of $\mathcal{F}(\alpha_i)$, $\mathcal{F}(\alpha_j)$ and they are not PF solutions. Let $\alpha_k = \frac{P_i - P_j}{Q_j - Q_i}$ and (P_k, Q_k) be a solution of $\mathcal{F}(\alpha_k)$. If one of the following conditions is satisfied, then $\alpha^{PF} \notin [\alpha_i, \alpha_j]$.*

1. *Either $\alpha_k = \alpha_i$ or $\alpha_k = \alpha_j$.*
2. *$\alpha_k \neq \alpha^{PF}$ and either $(P_k, Q_k) \equiv (P_i, Q_i)$ or $(P_k, Q_k) \equiv (P_j, Q_j)$.*
3. *$\alpha_k \neq \alpha^{PF}$, $(P_k, Q_k) \not\equiv (P_i, Q_i)$, $(P_k, Q_k) \not\equiv (P_j, Q_j)$ and $\alpha_j - \alpha_i < \epsilon$.*

Proof. Due to lack of space, the proof will be shown in Appendix. □

Notice that if the objectives P, Q are positive integers (this hypothesis is also verified for most combinatorial optimization problems), from (5) we have

$$\left|\frac{P' - P''}{Q'' - Q'} - \frac{P'' - P'''}{Q''' - Q''}\right| = \frac{|(P' - P'')(Q''' - Q'') - (P'' - P''')(Q'' - Q')|}{(Q'' - Q')(Q''' - Q'')} \geq \frac{1}{Q_{max}^2},$$

since $|(P' - P'')(Q''' - Q'') - (P'' - P''')(Q'' - Q')| \in \mathbb{Z}_+$, $0 < Q'' - Q', Q''' - Q'' \leq Q_{max}$ where Q_{max} is the maximum value of Q that can be obtained by solving the problem of maximizing Q. Consequently, when the objective values of Max-Max BOCO are positive integers, we can select the parameter ϵ as

$$\epsilon = \frac{1}{Q_{max}^2}, \tag{6}$$

Finally, by combining these three procedures, our algorithm to determine the PF solution for Max-Max BOCO can be stated as follows.

Algorithm 4 Determine the PF solution for Max-Max BOCO

Input: An instance of Max-Max BOCO, ϵ defined as (5).
Output: PF solution if it exists or Null otherwise.
1: solving $\mathcal{F}(0)$ to obtain a solution (P_0, Q_0)
2: **if** $Verify\_PF\_sol(0, P_0, Q_0) ==$ True **then** return (P_0, Q_0)
3: **end if**
4: $\alpha^{sup} \leftarrow P_0/Q_0 + 1$
5: solving $\mathcal{F}(\alpha^{sup})$ to obtain a solution (P^{sup}, Q^{sup})
6: **if** $Verify\_PF\_sol(\alpha^{sup}, P^{sup}, Q^{sup}) ==$ True **then** return (P^{sup}, Q^{sup})
7: **else** return $SEARCH(0, P_0, Q_0, \alpha^{sup}, P^{sup}, Q^{sup}, \epsilon)$
8: **end if**

By the following lemma and theorem, we show that all the conditions as mentioned in the input of Algorithm 3 are satisfied for running Procedure

$SEARCH(0, P_0, Q_0, \alpha^{sup}, P^{sup}, Q^{sup}, \epsilon)$. Then, Algorithm 4 can determine the PF solution for Max-Max BOCO in a logarithmic number of iterations in terms of ϵ and α^{sup}.

Lemma 5. *In the setting of Algorithm 4, if (P_0, Q_0) and (P^{sup}, Q^{sup}) are not PF solutions then $(P_0, Q_0) \not\equiv (P^{sup}, Q^{sup})$ and $0 < \alpha^{PF} < \alpha^{sup}$. Moreover, we have $T(0) > 0$ and $T(\alpha^{sup}) < 0$.*

Proof. If $(P_0, Q_0) \equiv (P^{sup}, Q^{sup})$ then (P_0, Q_0) is a solution of both $\mathcal{F}(0)$ and $\mathcal{F}(\alpha^{sup})$. Thus, for all $\alpha \in (0, \alpha^{sup})$, (P^0, Q^0) is the solution of $\mathcal{F}(\alpha)$. By taking $\alpha = P_0/Q_0 \in (0, \alpha^{sup})$, we deduce (P_0, Q_0) is the PF solution due to Theorem 1 which leads to a contradiction.

Since $\alpha^{PF} > 0$, $P^{PF} \leq P_0$ and $Q^{PF} \geq Q_0$ due to Lemma 1. Thus

$$\alpha^{PF} = \frac{P^{PF}}{Q^{PF}} \leq \frac{P_0}{Q_0} < \frac{P_0}{Q_0} + 1 = \alpha^{sup},$$

Obviously, $T(0) = P_0 - 0Q_0 = P_0 > 0$.

Since $\alpha^{sup} > 0$, we also have $P^{sup} \leq P_0$ and $Q^{sup} \geq Q_0$. Hence,

$$T(\alpha^{sup}) = P^{sup} - \alpha^{sup} Q^{sup} < P^{sup} - \frac{P_0}{Q_0} Q^{sup} = \frac{P^{sup} Q_0 - P_0 Q^{sup}}{Q_0} \leq 0$$

□

Theorem 2. *Algorithm 4 can determine the PF solution in a logarithmic number of iterations in terms of ϵ and α^{sup}.*

Proof. The execution of Algorithm 4 is based on the binary search algorithm for the interval $[0, \alpha^{sup}]$ with a length equals α^{sup}. At each iteration, we divided an interval into two half-intervals with equal length. Then, the half in which the PF coefficient cannot exist is eliminated and the search continues on the remaining half. Since Algorithm 4 terminated in the worst case when it found an interval with a length smaller than ϵ, the number of iterations for Algorithm 4 is $O(\log_2 \frac{\alpha^{sup}}{\epsilon})$. Consequently, Algorithm 4 can determine the PF solution in a logarithmic number of iterations in terms of ϵ and α^{sup}. □

Due to Theorem 2, notice that if solving $\mathcal{F}(\alpha)$ and $\mathcal{G}(\alpha)$ can be done in polynomial time, then the PF solution can be determined in polynomial time.

4 Experimental study on the BOSTP

4.1 Definition and modeling

In this section, we first restate the BOSTP that we utilized in Section 3.2. The BOSTP is a variant of the spanning tree problem that merges the Maximum STP, which involves maximizing the total profit, and the Max-Min STP, where the goal is to maximize the minimum edge reliability. Notice that the Maximum

STP is algorithmically equivalent to the *Minimum STP* which is a fundamental optimization problem that can be solved efficiently in polynomial time [14]. The Max-Min STP - which is also algorithmically equivalent to a variant called *Min-Max STP* mentioned in the prior literature [15] - aims at constructing solutions having a good performance in the worst case. For instance, in network design and optimization, the Max-Min STP can help ensure that the weakest link (edge with minimum reliability) in a communication or transportation network is as strong as possible, minimizing the risk of failure or congestion.

For the BOSTP, we find a spanning tree achieving proportional fairness between two objectives: the total profit representing the overall efficiency and the minimum of the edge reliability representing the individual fairness. Notice that profit and reliability are two important criteria in the various applications of the spanning tree problem [16]. Furthermore, for the simplicity of calculation, we suppose that the values of profit and reliability are positive integers. Thus, the objective values of the BOSTP are also positive integers, and then the parameter ϵ can be selected as mentioned in (6).

We consider a finite, connected, undirected graph $G = (V, E)$ where $V = [n] := \{1, ..., n\}$ with $n \geq 2$, $|E| = m$ and $p_e, r_e \in \mathbb{Z}_+$ are two weights associated with edge $e \in E$ representing respectively profit and reliability on this edge. Let $\mathcal{T}(G)$ denote the set of all spanning trees in G. Let $P, Q > 0$ denote respectively the total profit and the minimum edge reliability in a spanning tree of G. The BOSTP can be formally formulated as

$$\max_{T \in \mathcal{T}(G)} \quad P = \sum_{e \in T} p_e \tag{7a}$$

$$\max_{T \in \mathcal{T}(G)} \quad Q = \min_{e \in T} r_e \tag{7b}$$

As shown in Section 3, for determining the PF solution, we aim to solve $\mathcal{F}(\alpha)$ and $\mathcal{G}(\alpha)$ for some $\alpha \in [0, \alpha^{sup}]$. According to Section 2, we present the formulation for $\mathcal{F}(\alpha)$.

$$\mathcal{F}(\alpha): \max \quad P + \alpha Q \tag{8a}$$

$$\text{s.t.} \quad P = \sum_{e \in E} p_e x_e \tag{8b}$$

$$Q \leq r_e x_e + (1 - x_e)M \qquad \forall e \in E \tag{8c}$$

$$\sum_{e \in E} x_e = n - 1 \tag{8d}$$

$$\sum_{e \in \delta(V')} x_e \geq 1 \qquad \forall V' \subseteq V, \emptyset \neq V' \neq V \tag{8e}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E \tag{8f}$$

where x_e is the binary variables representing the occurrence of edge e in the spanning tree solution. Constraint (8d) is the degree constraint which assures

that there are exactly $n-1$ edges in the spanning tree solution. Constraints (8e) are the subtour elimination constraints: $\delta(V')$ is the set of edges crossing the cut (one endpoint in V' and one in $V - V'$).

Constraints (8c) allow bounding Q by the minimum edge reliability in the spanning tree solution. Indeed, in case $x_e = 1$, Constraints (8c) guarantee that Q is smaller than all the edge reliabilities in the tour. Otherwise, when x_e equals 0, the largest edge reliability M assures the validity of Constraints (8c). As $P + \alpha Q$ is maximized, Q will take the minimum edge reliability values.

We present the following formulation which contains all the constraints from (8b) to (8f) for $\mathcal{G}(\alpha)$. However, to prevent redundancy, these constraints have been omitted.

$$\mathcal{G}(\alpha) : \max \ P + \alpha Q - t \tag{9a}$$
$$\text{s.t. } t \geq P - \alpha Q \tag{9b}$$
$$t \geq \alpha Q - P \tag{9c}$$

Using two constraints (9b) and (9c), the parameter t represents the absolute value of $P - \alpha Q$.

It is important to note that a special-purpose algorithm can be used for solving $\mathcal{F}(\alpha)$ as well as $\mathcal{G}(\alpha)$ in polynomial time which is similar to the one for solving the Min-Max STP [15]. It is based on there are at most $O(n^2)$ different values of Q and the Maximum STP can be solved in polynomial time. However, in this section, we show the computational results by solving directly the MIP formulations of $\mathcal{F}(\alpha)$ and $\mathcal{G}(\alpha)$ due to its simple setting and better running time.

4.2 Computational results on the instances of the BOSTP

We investigate the performance of the presented algorithm for the BOSTP on random NetworkX graph. It returns a $G_{n,pro}$ random graph, also known as an Erdos-Renyi graph or a binomial graph [17] where n is the number of nodes and pro is the probability for edge creation. For this paper, we selected the number of nodes from the interval $[15, 40]$ with probability $pro = 0.5$. Moreover, the edge profit and the edge reliability are generated uniformly randomly respectively in the intervals $[100, 900]$ and $[10, 90]$.

The optimal solutions for the Maximum STP, Max-Min STP, and BOSTP are shown in Table 1. We also provided the time calculation and the number of iterations in the columns "Time" and "Iters". For each number of nodes n, we have generated two distinct graphs "GNn_1" and "GNn_2". The values of P, Q in case the PF solution does not exist are denoted as "Null". For solving these MIP formulations, we use CPLEX 12.10 on a PC Intel Core i5-9500 3.00GHz with 6 cores and 6 threads.

According to Table 1, we obtained the PF solutions for most instances and they are different from the solutions of the Maximum STP and the Max-Min STP. For the instance "GN20_1", we see that the PF solution has the same value of Q compared to the solution of the Max-Min STP but the value of P

Table 1: Computational results of Maximum STP, Max-Min STP and BOSTP

Instance	Maximum STP			Max-Min STP			BOSTP			
	P	Q	Time	P	Q	Time	P	Q	Time	Iters
GN15_1	10809	18	0.01	7727	59	0.01	9837	57	0.26	2
GN15_2	10812	10	0.02	7186	61	0.01	8587	57	0.48	3
GN20_1	15554	12	0.01	9390	54	0.03	11860	54	0.24	1
GN20_2	15152	14	0.02	9058	62	0.18	13179	61	0.30	2
GN25_1	20300	13	0.05	11046	66	0.12	Null	Null	1.68	3
GN25_2	20334	10	0.04	12052	73	0.23	15743	68	3.29	2
GN30_1	24259	12	0.10	14062	74	0.30	21633	67	3.28	2
GN30_2	24272	16	0.07	13359	74	0.08	18651	71	1.96	2
GN35_1	28329	11	0.08	19314	77	0.24	Null	Null	5.24	4
GN35_2	28554	17	0.05	17944	69	0.25	25138	68	4.42	2
GN40_1	33531	10	0.14	20432	73	0.24	28358	72	6.45	3
GN40_2	33681	12	0.14	17789	79	0.65	29171	68	5.07	2

is much better. Generally, the PF solutions offer a more favorable compromise between two objectives than the solutions of the Maximum STP (resp. Max-Min STP): the significant increase in the values of Q (resp. P) compared to the slight drop in the values of P (resp. Q) in percentage. Table 1 also indicates that our algorithm seems to converge quickly in terms of time calculation and number of iterations. It is worth noting that the upper bound on the number of iterations, as specified in Theorem 2, may theoretically be higher due to the determinations of α^{sup} and ϵ. However, in practice, adding the selecting method of α_k helps us quickly verify the existence of the PF solution rather than only using the binary search algorithm, especially when there is no PF solution and we need to wait for an interval to reach a length smaller than ϵ. Another important remark is that the existence of the PF solution seems to be much related to the edge weights and the structure of the graph rather than to the size of the graph. Although we selected randomly the values of profit and reliability, the PF solutions appeared with a high frequency, approximately 85% over the total tested instances.

5 Conclusion

In this paper, we have applied *proportional fairness* in the context of *Max-Max Bi-Objective Combinatorial Optimization* (Max-Max BOCO) where the two objectives to be maximized take only positive values and the feasible set is discrete, finite and non-convex. We considered a general Max-Max BOCO problem where we looked for a solution achieving proportional fairness between two objectives - which is referred to as *proportional fair solution* (PF solution). We first presented the characterization of the PF solution for Max-Max BOCO. Then, we designed an exact algorithm that converges within a logarithmic number of iterations to determine the PF solution. Finally, computational experiments on some in-

stances of the Bi-Objective Spanning Tree Problem have shown the effectiveness of our algorithm, indicating its rapid convergence.

Future work should focus on enhancing and assessing the algorithm's complexity in some special scenarios. For example, we may deal with Max-Max BOCO having linear objectives and binary variables. In addition, in cases where there is no PF solution, we are also interested in deriving a near-PF solution that can be quickly obtained by our algorithm rather than showing the absence of the PF solution.

References

1. Kelly, F.P., Maullo, A.K., and Tan, D.K.H.: Rate control for communication networks: shadow prices, proportional fairness, and stability. In: Journal of the Operational Research Society, 49(3), November 1997.
2. Kushner, H.J., and Whiting, P.A.: Convergence of proportional-fair sharing algorithms under general conditions. In: IEEE Transactions on Wireless Communications, 3(4):1250–1259, July 2004.
3. Nicosia, G., Pacifici, A., and Pferschy, U.: Price of Fairness for allocating a bounded resource. In: European Journal of Operational Research, 257(3), March 2017.
4. Bertsimas, D., Farias, V.F., and Trichakis, N.: The Price of Fairness. In: Operations Research, 59(1), pp 17-31, February 2011.
5. Nash, J.F.: The bargaining problem. In: Econometrica, Vol. 18, Issue 2, April 1950.
6. Ogryczak, W., Luss, H., Pioro, M., Nace, D., and Tomaszewski, A.: Fair Optimization and Networks: A survey. In: Journal of Applied Mathematics, Hindawi Publishing Corporation, pp 1-26, 2014.
7. Changho Sub, Seunghoon Park, and Youngkwon Cho.: Efficient Algorithm for Proportional Fairness Scheduling in Multicast OFDM Systems. In: IEEE 61st Vehicular Technology Conference, Stockholm, May-June 2005.
8. Johannes Brehmer, and Wolfgang Utschick.: On Proportional Fairness in Nonconvex Wireless Systems. In: International ITG Workshop on Smart Antennas - WSA, Berlin, February 2009.
9. Holger Boche, and Martin Schubert.: Nash Bargaining and Proportional Fairness for Wireless Systems. In: IEEE/ACM Transactions on Networking, Vol. 17, No. 5, October 2009.
10. Nguyen, M.H, Baiou, M., Nguyen, V.H., and T.Q.T. Vo.: Nash fairness solutions for balanced TSP. In: International Network Optimization Conference, March 2022. *DOI:10.48786/inoc.2022.17*
11. Nguyen, M.H, Baiou, M., and Nguyen, V.H.: Nash balanced assignment problem. International Symposium on Combinatorial Optimization, pp 172-186, May 2022. URL *https://doi.org/10.1007/978-3-031-18530-4_13*
12. Nguyen, M.H, Baiou, M., Nguyen, V.H., and T.Q.T. Vo.: Generalized Nash Fairness solutions for Bi-Objective Minimization Problems. In: Networks, pp 1-17, August 2023. URL *https://doi.org/10.1002/net.22182*
13. Nguyen, M.H., Baiou, M., and Nguyen, V.H.: Determining the generalized Nash Fairness solution set for Bi-Objective Discrete Optimization. In: Submitted to Discrete Applied Mathematics, March 2023. URL *https://hal.science/hal-04010827v1*
14. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. In: American Mathematical Society, 7(1), 1956, pp 48-50.

15. Camerini, P.M.: The Min-Max Spanning Tree Problem. In: Information Processing Letters, Vol. 7, Number 1, 1978.
16. Sayed, B.E., and Ehsan, B.: Optimizing profit and reliability using a bi-objective mathematical model for oil and gas supply chain under disruption risks. In: Computer & Industrial Engineering, Vol. 163, January 2022.
17. Erdos, P., and Renyi, A.: On random graphs. In: Publ. Math. 6, pp 290-297, 1959.

APPENDIX

Theorem 1. *$(P^{PF}, Q^{PF}) \in \mathcal{S}$ is the PF solution if and only if (P^{PF}, Q^{PF}) is a solution of $\mathcal{F}(\alpha^{PF})$ with $\alpha^{PF} = P^{PF}/Q^{PF}$.*

Proof. $\Longrightarrow$ Let (P^{PF}, Q^{PF}) be the PF solution. Let $\alpha^{PF} = P^{PF}/Q^{PF}$. We will show that (P^{PF}, Q^{PF}) is a solution of $\mathcal{F}(\alpha^{PF})$.

As (P^{PF}, Q^{PF}) is the PF solution, we have

$$\frac{P}{P^{PF}} + \frac{Q}{Q^{PF}} \leq 2,\ \forall (P,Q) \in \mathcal{S}, \tag{10}$$

Multiplying (10) by $P^{PF} > 0$ gives

$$P + \frac{P^{PF}}{Q^{PF}} Q \leq P^{PF} + \frac{P^{PF}}{Q^{PF}} Q^{PF},\ \forall (P,Q) \in \mathcal{S}, \tag{11}$$

Since $\alpha^{PF} = P^{PF}/Q^{PF}$, we deduce from (11)

$$P^{PF} + \alpha^{PF} Q^{PF} \geq P + \alpha^{PF} Q,\ \forall (P,Q) \in \mathcal{S},$$

Hence, (P^{PF}, Q^{PF}) is a solution of $\mathcal{F}(\alpha^{PF})$.

$\Longleftarrow$ Now suppose that (P^{PF}, Q^{PF}) is a solution of $\mathcal{F}(\alpha^{PF})$ with $\alpha^{PF} = P^{PF}/Q^{PF}$, we show that (P^{PF}, Q^{PF}) is the PF solution.

As (P^{PF}, Q^{PF}) is a solution of $\mathcal{F}(\alpha^{PF})$, we have

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

Replacing α^{PF} by P^{PF}/Q^{PF}, we obtain

$$\frac{P}{P^{PF}} + \frac{Q}{Q^{PF}} \leq 2,\ \forall (P,Q) \in \mathcal{S},$$

which implies (P^{PF}, Q^{PF}) is the PF solution. □

Lemma 4. *Let $[\alpha_i, \alpha_j]$ be an interval defined by $0 \leq \alpha_i < \alpha_j$, α_i, α_j are not PF coefficients. Let $(P_i, Q_i) \not\equiv (P_j, Q_j)$ be respectively the solutions of $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$ such that they are not PF solutions. Let $\alpha_k = \frac{P_i - P_j}{Q_j - Q_i}$ and (P_k, Q_k) be a solution of $\mathcal{F}(\alpha_k)$. If one of the following conditions is satisfied, then $\alpha^{PF} \notin [\alpha_i, \alpha_j]$.*

1. Either $\alpha_k = \alpha_i$ or $\alpha_k = \alpha_j$.
2. $\alpha_k \neq \alpha^{PF}$ and either $(P_k, Q_k) \equiv (P_i, Q_i)$ or $(P_k, Q_k) \equiv (P_j, Q_j)$.
3. $\alpha_k \neq \alpha^{PF}$, $(P_k, Q_k) \not\equiv (P_i, Q_i)$, $(P_k, Q_k) \not\equiv (P_j, Q_j)$ and $\alpha_j - \alpha_i < \epsilon$.

Proof. We first show that α_k is well defined. Since $\alpha_i < \alpha_j$, we have $P_i \geq P_j$, $Q_i \leq Q_j$ due to Lemma 1. Suppose that $Q_i = Q_j$. The optimality of (P_j, Q_j) gives

$$P_j + \alpha_j Q_j \geq P_i + \alpha_j Q_i, \tag{12}$$

Since $Q_i = Q_j$, we obtain $P_j \leq P_i$. Thus, $P_i = P_j$ and then $(P_i, Q_i) \equiv (P_j, Q_j)$ which leads to a contradiction.

Hence, $Q_j > Q_i$ and consequently, α_k is well defined.

We then show that $\alpha_k \in [\alpha_i, \alpha_j]$. The optimality of (P_i, Q_i) gives

$$P_i + \alpha_i Q_i \geq P_j + \alpha_i Q_j, \tag{13}$$

From (12) and (13), we obtain $\alpha_i \leq \frac{P_i - P_j}{Q_j - Q_i} \leq \alpha_j$ which leads to $\alpha_i \leq \alpha_k \leq \alpha_j$.

1. If $\alpha_k = \alpha_i$ then $P_i + \alpha_i Q_i = P_j + \alpha_i Q_j$. Thus, (P_i, Q_i) and (P_j, Q_j) are both solutions of $\mathcal{F}(\alpha_i)$. Hence, for all $\alpha \in (\alpha_i, \alpha_j)$, (P_j, Q_j) is the unique solution of $\mathcal{F}(\alpha)$ as a result of Lemma 1. Similarly, if $\alpha_k = \alpha_j$, (P_i, Q_i) is the unique solution of $\mathcal{F}(\alpha)$ for all $\alpha \in (\alpha_i, \alpha_j)$. Since (P_i, Q_i), (P_j, Q_j) are not PF solutions and α_i, α_j are not PF coefficients, we have $\alpha^{PF} \notin [\alpha_i, \alpha_j]$.

2. Let (P_k, Q_k) be a solution of $\mathcal{F}(\alpha_k)$. Without loss of generality, we suppose that $\alpha_k \neq \alpha^{PF}$ and $(P_k, Q_k) \equiv (P_i, Q_i)$. Consequently, (P_i, Q_i) is a solution of $\mathcal{F}(\alpha_k)$. Since $\alpha_k = \frac{P_i - P_j}{Q_j - Q_i}$, $P_i + \alpha_k Q_i = P_j + \alpha_k Q_j$. Thus, (P_j, Q_j) is also a solution of $\mathcal{F}(\alpha_k)$.

As a result of Lemma 1, when $\alpha \in (\alpha_i, \alpha_k)$ (resp. $\alpha \in (\alpha_k, \alpha_j)$), (P_i, Q_i) (resp. (P_j, Q_j)) is the unique solution of $\mathcal{F}(\alpha)$. Consequently, $\alpha^{PF} \notin [\alpha_i, \alpha_j]$.

3. Similar to the proof above, we also have $Q_i < Q_k < Q_j$ and

$$\alpha_i \leq \frac{P_i - P_k}{Q_k - Q_i} \leq \alpha_k \leq \frac{P_k - P_j}{Q_j - Q_k} \leq \alpha_j,$$

According to the definition of ϵ, if $\frac{P_i - P_k}{Q_k - Q_i} > \frac{P_k - P_j}{Q_j - Q_k}$ we obtain $\alpha_j - \alpha_i \geq \frac{P_i - P_k}{Q_k - Q_i} - \frac{P_k - P_j}{Q_j - Q_k} \geq \epsilon$ which leads to a contradiction.

Thus, we have $\frac{P_i - P_k}{Q_k - Q_i} = \frac{P_k - P_j}{Q_j - Q_k}$. Consequently, $\frac{P_i - P_k}{Q_k - Q_i} = \alpha_k = \frac{P_k - P_j}{Q_j - Q_k}$ which implies $P_k + \alpha_k Q_k = P_i + \alpha_k Q_i = P_j + \alpha_k Q_j$. In other words, (P_i, Q_i) and (P_j, Q_j) are also the solutions of $\mathcal{F}(\alpha_k)$. Similar to the case 2, we deduce $\alpha^{PF} \notin [\alpha_i, \alpha_j]$. □