

## THE MAGIC OF NASH SOCIAL WELFARE IN OPTIMIZATION: DO NOT SUM, JUST MULTIPLY!

HADI CHARKHGARD <sup>1</sup>, KIMIA KESHANIAN <sup>2</sup>, RASUL ESMAEILBEIGI <sup>3</sup> and  
PARISA CHARKHGARD <sup>4</sup>

(Received 29 April, 2021; accepted 2 May, 2022; first published online 7 July, 2022)

### Abstract

We explain some key challenges when dealing with a single- or multi-objective optimization problem in practice. To overcome these challenges, we present a mathematical program that optimizes the Nash social welfare function. We refer to this mathematical program as the Nash social welfare program (NSWP). An interesting property of the NSWP is that it can be constructed for any single- or multi-objective optimization problem. We show that solving the NSWP could result in more desirable solutions in practice than its single- or multi-objective counterpart. We also discuss several promising approaches that could be employed to solve the NSWP in practice.

2020 *Mathematics subject classification*: primary 90C29; secondary 90B50, 91A12.

*Keywords and phrases*: Nash social welfare, equity, single-objective optimization, multi-objective optimization, multiplicative program.

### 1. Introduction

Without loss of generality, a single-objective optimization problem can be stated as

$$\max \left\{ \sum_{i=1}^{p'} w_i f_i(\mathbf{x}) - \sum_{i=p'+1}^p w_i f_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X} \right\}, \quad (1.1)$$

where  $p \geq p' \geq 0$  and  $\mathcal{X} \subseteq \mathbb{R}^n$  represents the set of feasible solutions, and is assumed to be bounded. Moreover,  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_p(\mathbf{x}))$  is a vector of arbitrary functions

<sup>1</sup>Department of Industrial and Management Systems Engineering, University of South Florida, Tampa, FL 33620, USA; e-mail: [hcharkhgard@usf.edu](mailto:hcharkhgard@usf.edu)

<sup>2</sup>Information and Technology Management Department, University of Tampa, Tampa, FL 33606, USA; e-mail: [kkeshanian@ut.edu](mailto:kkeshanian@ut.edu)

<sup>3</sup>School of Information Technology, Deakin University, Geelong, Victoria 3220, Australia; e-mail: [r.esmaeilbeigi@deakin.edu.au](mailto:r.esmaeilbeigi@deakin.edu.au)

<sup>4</sup>School of Mathematical and Physical Sciences, University of Newcastle, New South Wales 2308, Australia; e-mail: [parisa.charkhgard@uon.edu.au](mailto:parisa.charkhgard@uon.edu.au)

© The Author(s), 2022. Published by Cambridge University Press on behalf of Australian Mathematical Publishing Association Inc.

and  $\mathbf{w} = (w_1, \dots, w_p)$  is a vector of weights with  $w_i > 0$  for all  $i \in \{1, \dots, p\} = [1, p]$ . In the remainder of this paper, we refer to the objective function of equation (1.1) as the (weighted) social welfare function and to equation (1.1) as the social welfare program (SWP).

Similarly, a multi-objective optimization problem can be stated as

$$\begin{aligned} & \max \quad \{f_i(\mathbf{x}) \mid i \in S^+\}, \\ & \min \quad \{f_i(\mathbf{x}) \mid i \in S^-\}, \\ & \text{subject to} \quad \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{1.2}$$

where  $S^+$  and  $S^-$  are the (index) sets of objective functions that need to be maximized and minimized, respectively. Let  $S^+ = [1, p']$  and  $S^- = [p' + 1, p]$  which implies that  $S^+ \cup S^- = [1, p]$ . In the remainder of this paper, we refer to equation (1.2) as the multi-objective optimization program (MOOP).

The main goal of this paper is to introduce a different problem, the Nash social welfare program (NSWP), that could result in solutions with desirable properties for both equations (1.1) and (1.2). In other words, instead of solving the SWP or MOOP, modellers and/or practitioners can solve the NSWP to obtain more practical solutions. We show that for any single- or multi-objective optimization problem, the NSWP can be easily constructed from its corresponding SWP or MOOP. We also show that the NSWP can be solved in polynomial time if its corresponding SWP or MOOP is a (continuous) convex optimization problem.

We would like to highlight that there are some studies in the literature addressing variations of the NSWP. To the best of our knowledge, the first study regarding the NSWP was conducted by Rao [15] in the context of multi-objective optimization. However, there is no article in the literature that provides some insights as to why the NSWP should be employed, how it can be generated for any single- or multi-objective optimization problem, and more importantly, how it can be solved. The present work is an attempt to fill this gap in the literature. We believe that the NSWP has not received the attention it deserves from the operations research community. We hope that this article will attract more researchers and practitioners to study and employ the NSWP as a promising approach to model many real-world problems.

The rest of this paper is organized as follows. In Section 2, we discuss the motivation for writing this paper by providing some reasons as to why solving the SWP and MOOP can be problematic in practice. In Section 3, the NSWP is explained in detail. In Section 4, our proposed solution approaches for solving the NSWP are introduced. Finally, in Section 5, some concluding remarks are given.

## 2. Motivation

In this section, we will explain the main weaknesses of the SWP and MOOP. The main motivation for developing the NSWP is to overcome these weaknesses.

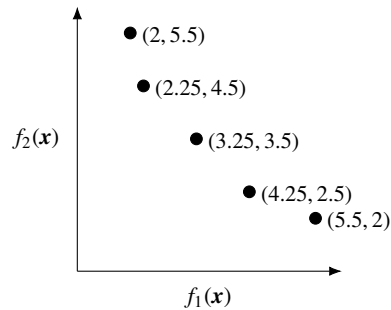


FIGURE 1. An illustration of a Pareto-optimal frontier with two (maximization) objective functions in the criterion space.

**2.1. The multi-objective optimization program** In general, when solving the MOOP, decision makers are only interested in the so-called *Pareto-optimal* solutions (see Definition 2.1).

**DEFINITION 2.1.** A feasible solution  $\mathbf{x} \in \mathcal{X}$  is called *Pareto-optimal* if there is no other  $\mathbf{x}' \in \mathcal{X}$  such that

$$\begin{aligned} f_i(\mathbf{x}) &\leq f_i(\mathbf{x}') && \text{for all } i \in S^+, \\ f_i(\mathbf{x}) &\geq f_i(\mathbf{x}') && \text{for all } i \in S^-, \\ f_i(\mathbf{x}) &\neq f_i(\mathbf{x}') && \text{there exists } i \in S^+ \cup S^-. \end{aligned}$$

The image of each Pareto-optimal solution in the criterion space is called a *Pareto-optimal point*. The set of all Pareto-optimal points, also known as the *Pareto-optimal frontier*, shows the trade-off between the objectives. An illustration of the trade-off of the MOOP involving only two objectives (in the form of maximization) can be found in Figure 1.

A typical challenge when dealing with the MOOP is to answer “how should a Pareto-optimal solution be selected for implementation in practice?” A natural answer to this question is that modellers have no responsibility in identifying a desirable solution and the decision makers (or managers) are expected to do that in practice. In other words, modellers need to compute the trade-off (between Pareto-optimal solutions) and ask the decision makers to select the one which is suitable. Such a process is typically implemented using either a two-phase approach or a step-by-step approach. In the two-phase approach, the complete trade-off will be computed in the first phase and then the entire information will be sent to the decision makers. However, the step-by-step approach is interactive in the sense that at each iteration, one (or some) Pareto-optimal solutions will be computed by the modellers and then the opinions of the decision makers will be obtained. If the decision makers are happy with the provided solution(s), then the search terminates, otherwise the search will be modified for finding some other Pareto-optimal solutions (if any). This process repeats until a desirable solution is found.

Although the approaches described above are useful in many practical situations for selecting a desirable Pareto-optimal solution, they do not work in many other situations, because they suffer from two main weaknesses described in the following.

**WEAKNESS 2.2.** *In some practical situations, computing even a single Pareto-optimal solution is computationally infeasible (or expensive). So, selecting a desirable Pareto-optimal solution using the existing approaches will be infeasible.*

**WEAKNESS 2.3.** *In some practical situations, either there is no decision maker or the decision maker(s) do not know how to select a desirable solution [8]. So, selecting a desirable Pareto-optimal solution using the existing approaches will be infeasible.*

For Weakness 2.2, a possible remedy is to employ heuristic or evolutionary solution methods. However, inexact methods do not guarantee optimality and may perform poorly. More importantly, they still do not address Weakness 2.3. Our proposed technique in this study, that is, the NSWP, addresses both weaknesses at the same time, because it is designed to directly generate a Pareto-optimal solution (see Proposition 3.3) that can create a desirable balance between the objectives.

**2.2. The social welfare program** To explain the main weaknesses of the SWP, we first show that the SWP itself is trying to solve the MOOP. Observe that the MOOP can be written in the following form:

$$\begin{aligned} & \max \quad \{f_i(\mathbf{x}) \mid i \in S^+\}, \\ & \max \quad \{-f_i(\mathbf{x}) \mid i \in S^-\}, \\ & \text{subject to} \quad \mathbf{x} \in \mathcal{X}. \end{aligned}$$

Now, a popular solution technique to deal with the MOOP is to aggregate its objective functions with some positive weights, and then solve the following single-objective optimization problem instead:

$$\max \left\{ \sum_{i \in S^+} w_i f_i(\mathbf{x}) - \sum_{i \in S^-} w_i f_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X} \right\}. \quad (2.1)$$

Equation (2.1) is sometimes called the weighted sum optimization in the literature of multi-objective optimization [7]. We can observe that the weighted sum optimization is precisely equation (1.1), because  $S^+ = [1, p']$  and  $S^- = [p' + 1, p]$ . Therefore, when modellers are trying to solve the SWP, they are basically trying to find a desirable Pareto-optimal solution for equation (1.2) by setting the weight of  $f_i(\mathbf{x})$  to  $w_i$  for all  $i \in [1, p]$ . In other words, they implicitly have  $p$  criteria in their mind, that is,  $f_1(\mathbf{x}), \dots, f_p(\mathbf{x})$  with  $w_1, \dots, w_p$  as their corresponding degrees of importance.

Given that the SWP can be viewed as an approach for computing a desirable Pareto-optimal solution for the MOOP, it is natural to ask whether it actually results in finding a desirable solution in practice. Note that one may argue that the SWP can remove Weaknesses 2.2 and 2.3 that the MOOP causes. While for many practical situations this can be indeed true, we will argue in the rest of this section that the SWP

should be generally avoided, because it may not generate a desirable Pareto-optimal solution, that is, it may not provide a good balance between the values of different objectives in practice.

**WEAKNESS 2.4.** *For nonconvex multi-objective optimization problems, there can exist many (and possibly infinite) Pareto-optimal points that cannot be obtained by optimizing a (positive) weighted summation of all objective functions over the feasible set (even by testing all possible positive weights). Such points are called unsupported Pareto-optimal points [7]. So, a weakness of the SWP is that it completely ignores the existence of unsupported Pareto-optimal points that can possibly better balance different objectives.*

To better understand Weakness 2.4, consider Figure 1 again. In the figure, we have  $p = 2$  and  $S^- = \emptyset$ , that is, there are two objectives in the form of maximization. It is not hard to see that if we use the SWP, then we will find either the point (2,5.5) or point (5.5,2) for any possible values of  $w_1$  and  $w_2$  (note that  $w_1, w_2 > 0$ ). So, none of the three middle points can be obtained using the SWP because they are unsupported Pareto-optimal points. Observe that this can be problematic (in some practical cases) because from the view point of multi-objective optimization, those middle points balance the conflicting objectives better than the endpoints.

**WEAKNESS 2.5.** *In many practical situations,  $f_1(\mathbf{x}), \dots, f_p(\mathbf{x})$  represent the objective functions of  $p$  independent players (or components) of an environment (or system). So, by developing the SWP for such cases, a modeller is basically trying to coordinate these players with respect to their powers, denoted by  $w_1, \dots, w_p$ , and make sure that the benefit of the system is distributed (almost) fairly among players with respect to their powers. However, a weakness of the SWP is that it does not necessarily ensure the fairness.*

By fairness, we mean less extreme solutions in which the total cost (or gain) of the system is distributed among the players according to their powers (or contributions). To explain Weakness 2.5 better, consider Figure 1 again where  $p = 2$  and  $S^- = \emptyset$ . We know that in this example, we will find either the point (2,5.5) or (5.5,2) for any possible values of  $w_1$  and  $w_2$ . So, suppose that  $w_1 = w_2 = 1$ . In this case, both (2,5.5) and (5.5,2) are optimal, and so the SWP returns one of them randomly. However, the powers of the players are exactly the same. So no matter which point is returned by the SWP, it will not be fair and make (one of) the players unhappy. For this particular example, we can easily see that the point (3.25,3) is indeed the only fair point that can make both players happy at the same time. Overall, our proposed technique in this study, that is, the NSWP, addresses both Weaknesses 2.4 and 2.5 at the same time, because it is designed to directly generate a Pareto-optimal solution that can create a desirable balance between the objectives.

**2.3. A practical example** To better illustrate the weaknesses mentioned earlier, we now provide a (simplified) real-world problem (or scenario). In the field

of conservation planning, one of the most important problems is the so-called reserve selection problem [20]. Suppose that there are  $n$  parcels of land, and parcel  $j \in [1, n]$  has an associated cost denoted by  $c_j$ . As a modeller, you have a budget, denoted by  $b$ , to spend on buying some of these parcels and you want to develop an optimization model that can help natural resource managers to identify which parcels should be selected to be reserved. Suppose that there are (in total)  $p$  species where  $S^+$  represents the index set of engendered (or important species) and  $S^-$  represents the index set of invasive species. Accruing parcel  $j \in [1, n]$  has an impact on species  $i \in [1, p]$ , for example, increases its population size, which is denoted by parameter  $u_{ij} \geq 0$ . Observe that for this problem, we have

$$\mathcal{X} = \left\{ \mathbf{x} \in \{0, 1\}^n \mid \sum_{j=1}^n c_j x_j \leq b \right\},$$

where  $x_j$  is a binary variable that shows whether parcel  $j$  should be bought or not. Also, for each species  $i \in [1, p]$ , we have

$$f_i(\mathbf{x}) = \sum_{j=1}^n u_{ij} x_j.$$

Modellers in the field of conservation planning address the problem mentioned above by either using the SWP or the MOOP [20]. However, it is easy to see that if they use the MOOP, Weaknesses 2.2 and 2.3 can arise when  $n$  and  $p$  are large, respectively. Also, if they use the SWP, Weakness 2.4 can arise because this problem is nonconvex. Moreover, Weakness 2.5 can arise because the modeller is indeed working on a system where each species is a player on its own.

### 3. The Nash social welfare program

The underlying idea of the NSWP comes from the field of cooperative game theory. A well-known problem in the field of cooperative game theory is the so-called bargaining problem. The bargaining problem is a game in which all (competing) players agree to create a grand coalition, instead of competing with each other to get a higher payoff [5, 19]. To be able to create a grand coalition, the agreement of all players is necessary. Therefore, the main concern when dealing with a bargaining problem is what the payoff for each player should be in a grand coalition (and how it should be computed). One of the well-known solution techniques to this problem, introduced by Nash [14] and extended further by Kalai [9], is what we use in this paper. Specifically, we assume that there are  $p$  number of (imaginary) players and each player  $i \in S^+ \cup S^-$  has an objective function  $f_i(\mathbf{x})$ . Note that  $|S^+ \cup S^-| = p$ . Hence, we try to create a coalition between these players by solving the following

optimization problem:

$$\begin{aligned} \max \quad & \prod_{i \in S^+} (f_i(\mathbf{x}) - d_i)^{w_i} \times \prod_{i \in S^-} (d_i - f_i(\mathbf{x}))^{w_i} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}, \\ & f_i(\mathbf{x}) \geq d_i \quad \text{for all } i \in S^+, \\ & f_i(\mathbf{x}) \leq d_i \quad \text{for all } i \in S^-, \end{aligned} \tag{3.1}$$

where  $\mathbf{d} \in \mathbb{R}^p$  is the *reference point*, an input parameter for the NSWP. The reference point in the literature of the bargaining problem is sometimes referred to as the disagreement point and it basically indicates the payoff of each player under no coalition. We will later in Section 3.1 explain some natural choices for the reference point (if it is unknown to modellers). The following remark is helpful.

**REMARK 3.1.** For changing the MOOP to the NSWP, one can assume  $w_i = 1$  for all  $i \in S^+ \cup S^-$  unless it is known that the objective functions do not have the same preference/power.

The objective function of equation (3.1) is basically the (weighted) Nash social welfare function which is why we refer to equation (3.1) as the NSWP. Note that since player  $i \in S^+$  is interested in maximizing its objective function,  $(f_i(\mathbf{x}) - d_i)$  captures the benefit that it will obtain as the result of creating a coalition (with respect to the reference point). Similarly, since player  $i \in S^-$  is interested in minimizing its objective function,  $(d_i - f_i(\mathbf{x}))$  captures the benefit that it obtains as the result of creating a coalition (with respect to the reference point). So, the (weighted) Nash social welfare function is the product of benefits of players (considering their negotiating powers/weights) and the goal of the NSWP is to maximize the value of this function.

**ASSUMPTION 3.2.** In the remainder of this paper, we assume that the optimal objective value of the NSWP is nonzero, that is, strictly positive. This implies that  $\mathcal{X} \neq \emptyset$  and the disagreement point is selected such that there exists a feasible solution  $\mathbf{x} \in \mathcal{X}$  with  $f_i(\mathbf{x}) > d_i$  for all  $i \in S^+$  and  $f_i(\mathbf{x}) < d_i$  for all  $i \in S^-$ .

**PROPOSITION 3.3.** *The NSWP returns a Pareto-optimal solution.*

**PROOF.** Suppose that  $\mathbf{x}^*$  is an optimal solution of the NSWP, but it is not a Pareto-optimal solution. By definition of Pareto-optimality (Definition 2.1), this implies that there must exist a feasible solution denoted by  $\mathbf{x} \in \mathcal{X}$  that *dominates*  $\mathbf{x}^*$ , that is,

$$\begin{aligned} f_i(\mathbf{x}^*) &\leq f_i(\mathbf{x}) \quad \text{for all } i \in S^+, \\ f_i(\mathbf{x}^*) &\geq f_i(\mathbf{x}) \quad \text{for all } i \in S^-, \\ f_i(\mathbf{x}^*) &\neq f_i(\mathbf{x}) \quad \text{there exists } i \in S^+ \cup S^-. \end{aligned}$$

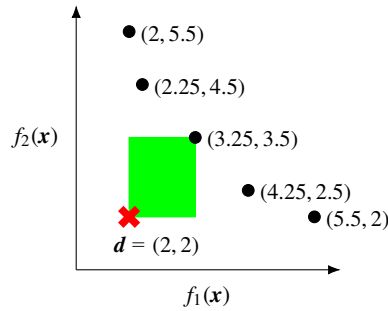


FIGURE 2. Selection of the Pareto-optimal point by the NSW. (Colour available online.)

In addition, by assumptions, we have  $w_i > 0$  for all  $i \in [1, p]$ . This, combined with Assumption 3.2, implies that

$$0 < \prod_{i \in S^+} (f_i(\mathbf{x}^*) - d_i)^{w_i} \times \prod_{i \in S^-} (d_i - f_i(\mathbf{x}^*))^{w_i} < \prod_{i \in S^+} (f_i(\mathbf{x}) - d_i)^{w_i} \times \prod_{i \in S^-} (d_i - f_i(\mathbf{x}))^{w_i}.$$

Consequently,  $\mathbf{x}^*$  cannot be an optimal solution, which is a contradiction. □

To illustrate what the NSW actually does, consider the same example described in Figure 1 where  $S_i^- = \emptyset$  and  $S_i^+ = \{1, 2\}$ . Suppose that  $w_1 = w_2 = 1$  and  $\mathbf{d} = (2, 2)$ . The Pareto-optimal point that NSW selects for this example is  $(3.25, 3.5)$ , as shown in Figure 2, which is not only an unsupported Pareto-optimal point, but also the only fair Pareto-optimal point from the view point of both players. Note that since  $w_1 = w_2 = 1$ , we observe that, geometrically, the NSW attempts to find a Pareto-optimal point such that the area of the box between this point and  $\mathbf{d}$  is maximized. In other words, the objective function of the NSW captures the area of the box. The green box (online) in Figure 2 shows the box with the maximum area. The following propositions and remark are also helpful.

**PROPOSITION 3.4.** *The NSW is global-power-scale-free, that is, the NSW with negotiation powers  $w_1, \dots, w_p$  is equivalent to the NSW with powers  $\alpha w_1, \dots, \alpha w_p$ , where  $\alpha$  is an arbitrary positive constant.*

**PROOF.** Using the notation presented in Section 4, we just need to show that

$$\max_{\mathbf{y} \in \mathcal{Y}} \prod_{i \in S^+ \cup S^-} y_i^{\alpha w_i}$$

is equivalent to equation (4.1). This immediately follows from two observations: first, equation (4.1) is equivalent to equation (4.2). Second, by multiplying the objective function of equation (4.2) by a positive constant, that is,  $\alpha$ , an equivalent problem can be constructed. □



Note that from Proposition 3.4, one can also infer that the NSWP with equal negotiating powers, that is,  $w_i = w_j$  for all  $i, j \in S^+ \cup S^-$ , is equivalent to the NSWP with unit negotiating powers, that is,  $w_i = 1$  for all  $i \in S^+ \cup S^-$ .

**PROPOSITION 3.5.** *The NSWP is local-benefit-scale-free, that is, by replacing the objective function of the NSWP with the following function:*

$$\max \prod_{i \in S^+} (\alpha_i f_i(\mathbf{x}) - \alpha_i d_i)^{w_i} \times \prod_{i \in S^-} (\alpha_i d_i - \alpha_i f_i(\mathbf{x}))^{w_i}, \quad (3.2)$$

an equivalent problem will be constructed where  $\alpha_1, \dots, \alpha_p$  are arbitrary positive constants.

**PROOF.** Observe that equation (3.2) is equivalent to

$$\max \prod_{i=1}^p \alpha_i^{w_i} \times \prod_{i \in S^+} (f_i(\mathbf{x}) - d_i)^{w_i} \times \prod_{i \in S^-} (d_i - f_i(\mathbf{x}))^{w_i}.$$

Since  $\prod_{i=1}^p \alpha_i^{w_i}$  is a positive constant, it has no impact on the optimization and can be dropped.  $\square$

Again note that in the NSWP, the term  $(f_i(\mathbf{x}) - d_i)$  captures the benefit of player  $i \in S^+$  and the term  $(d_i - f_i(\mathbf{x}))$  captures the benefit of player  $i \in S^-$  with respect to the reference point. In Proposition 3.5, if we set

$$\alpha_i = \frac{1}{\max_{\mathbf{x} \in \mathcal{X}'} f_i(\mathbf{x}) - d_i} \quad \text{for all } i \in S^+,$$

$$\alpha_i = \frac{1}{\max_{\mathbf{x} \in \mathcal{X}'} d_i - f_i(\mathbf{x})} \quad \text{for all } i \in S^-,$$

where

$$\mathcal{X}' = \{\mathbf{x} \in \mathcal{X} \mid f_i(\mathbf{x}) \geq d_i \text{ for all } i \in S^+, \text{ and } f_i(\mathbf{x}) \leq d_i \text{ for all } i \in S^-\},$$

then the maximum benefit that each player can obtain will be precisely one in the equivalent problem. This implies that from any NSWP, an equivalent NSWP with unit-maximum-benefit can be constructed, because for each  $i \in [1, p]$ ,  $\alpha_i f_i(\mathbf{x})$  and  $\alpha_i d_i$  can be viewed as the new  $f_i(\mathbf{x})$  and the new  $d_i$ , respectively.

**REMARK 3.6.** From Propositions 3.4 and 3.5, we know that the NSWP with equal negotiating powers/weights can be rewritten as the NSWP with unit-weights and unit-maximum-benefits. When solving the latter, the NSWP attempts to find a Pareto-optimal point where the area of the box between the reference point and the Pareto-optimal point is maximized. The area of the box cannot be more than one because the length of each side of the box indicates the benefit obtained by a player and the maximum benefit of each player is one. So, the NSWP with equal negotiating powers basically attempts to not only make the benefit of each player equal to one but

also make them relatively the same. So, the NSWP with equal negotiating powers is naturally designed to create equity in benefits or fairness [3, 6, 10, 13].

In summary, the NSWP is a single-objective optimization problem. It resolves Weakness 2.2, because it returns a single Pareto-optimal solution (just like the SWP) if being solved to optimality (see Proposition 3.3). Evidently, generating only a single Pareto-optimal solution can decrease the computational time significantly. Even for the cases where the NSWP is computationally expensive, one can develop a heuristic method for finding high-quality feasible solutions for it. The NSWP resolves Weakness 2.3, because it generally tries to avoid the endpoints of the Pareto-optimal frontier and generates a good balance between objectives (which is generally for what decision makers look). The NSWP resolves Weakness 2.4 because it does not ignore unsupported Pareto-optimal points and its output can be even an unsupported Pareto-optimal point. Finally, the NSWP resolves Weakness 2.5 because it attempts to ensure that the benefits are distributed fairly (based on Remark 3.6) with respect to the power of players. Specifically, if the negotiating powers are equal, then the NSWP attempts to create equality in benefits. Otherwise, it attempts to deviate from the equality based on the power of players.

**3.1. The reference point** As mentioned earlier, in the context of game theory, the reference point should denote the payoff of each player under no coalition, that is, the status quo of the game. While in many applications the reference point can be identified easily, in some cases, it is not clear how the reference point should be determined. For such cases, one natural choice for the reference point can be the so-called *nadir* point, that is, a point in the criterion space that denotes the worst value for each objective function in the Pareto-optimal frontier. Note that the nadir point is not necessarily a feasible point, that is, it may lie outside the feasible region in the criterion space. For example, in Figure 2, both objectives are in the form of maximization. So, the worst value for both  $f_1(x)$  and  $f_2(x)$  in the Pareto-optimal frontier is 2. Hence, the nadir point is (2,2).

However, computing the nadir point could be computationally expensive, especially for cases with  $p > 2$ . For cases with  $p = 2$ , the nadir point can be generated by computing the two endpoints of the Pareto-optimal frontier (see Figure 2). However, for cases with  $p > 2$ , this is not necessarily true in general [7]. Hence, alternatively, one can set the reference point to an approximation of the nadir point. For example, for each  $i \in S^+$ , one can set

$$d_i = \min \{f_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} \quad \text{or} \quad d_i = \min \{f_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}^R\},$$

where  $\mathcal{X}^R$  is a relaxation of  $\mathcal{X}$ , for example, linear programming relaxation. Similarly, for each  $i \in S^-$ , one can set

$$d_i = \max \{f_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} \quad \text{or} \quad d_i = \max \{f_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}^R\}.$$

We complete this section by emphasizing that the reference point is an input parameter for the NSWP. The choice of this point does highly impact the solution

obtained by solving the problem. In particular, a poor choice of the reference point could result in a solution which does not necessarily resolve all the weaknesses mentioned in this paper. Therefore, the reference point should be selected with care and by considering the particular problem to which it is applied.

#### 4. Solution methods for the Nash social welfare program

Observe that, in the NSWP, we have  $f_i(\mathbf{x}) - d_i \geq 0$  for all  $i \in S^+$ , because  $f_i(\mathbf{x}) \geq d_i$ . Similarly, we have  $d_i - f_i(\mathbf{x}) \geq 0$  for all  $i \in S^-$ , because  $f_i(\mathbf{x}) \leq d_i$ . Hence, for notational convenience, let

$$\begin{aligned} \mathcal{Y} = \{ \mathbf{y} \in \mathbb{R}^p \mid \mathbf{x} \in \mathcal{X}, \\ y_i = f_i(\mathbf{x}) - d_i \text{ and } y_i \geq 0 \text{ for all } i \in S^+, \\ y_i = d_i - f_i(\mathbf{x}) \text{ and } y_i \geq 0 \text{ for all } i \in S^- \}, \end{aligned}$$

be the feasible set of the NSWP in the criterion space. Using this notation, the NSWP can be rewritten as

$$\max_{\mathbf{y} \in \mathcal{Y}} \prod_{i \in S^+ \cup S^-} y_i^{w_i}. \quad (4.1)$$

**ASSUMPTION 4.1.** In the rest of this paper, we assume that  $\mathcal{Y}$  can be represented by only convex (and ideally linear) constraints involving continuous and/or integer decision variables.

We now describe a few solution approaches for solving the NSWP using Assumption 4.1. The first approach is straightforward. It applies the log-transformation to the objective function of the NSWP. So, the NSWP can be rewritten as follows:

$$\max_{\mathbf{y} \in \mathcal{Y}} \sum_{i \in S^+ \cup S^-} w_i \log(y_i). \quad (4.2)$$

Observe that equation (4.2) is basically maximizing a concave function over a mixed integer convex set (due to Assumption 4.1). So, the first approach calls a mixed integer convex programming solver to solve equation (4.2).

**REMARK 4.2.** Two advantages of the first approach are that (1) it can handle fractional weights directly and (2) it does not increase the size of the problem. However, a weakness of the first approach is that it involves log functions and (in practice) solvers generally perform better when dealing with problems involving linear/quadratic functions than log functions [21].

The second approach applies piecewise linearization techniques to first linearize the log functions in equation (4.2), and then call a (mixed) integer convex/linear programming solver to solve the linearized problem [3]. In general, the piecewise linear function is an approximation of  $\sum_{i \in S^+ \cup S^-} w_i \log(y_i)$ . However, if  $y_i$  can only take integer values in the interval  $[l_i, u_i]$  for all  $i \in S^+ \cup S^-$ , where  $l_i, u_i \in \mathbb{Z}_+$  and are nonnegative, then the piecewise linear function can be an exact representation of

$\sum_{i \in S^+ \cup S^-} w_i \log(y_i)$ . This is because one can set the breakpoints of the piecewise linear function for  $y_i$  where  $i \in S^+ \cup S^-$  is to be all integer values in the interval  $[l_i, u_i]$ . For example, if  $y_i$  can only take integer values for all  $i \in S^+ \cup S^-$ , then an exact piecewise linearized reformulation of equation (4.2) can be stated as follows:

$$\begin{aligned}
 & \max_{y \in \mathcal{Y}} \sum_{i \in S^+ \cup S^-} w_i \sum_{j=1}^{u_i - l_i + 1} b_{ij} \log(l_i + j - 1), \\
 & \text{subject to } y_i = \sum_{j=1}^{u_i - l_i + 1} b_{ij} (l_i + j - 1) \quad \text{for all } i \in S^+ \cup S^-, \\
 & \sum_{j=1}^{u_i - l_i + 1} b_{ij} = 1 \quad \text{for all } i \in S^+ \cup S^-, \\
 & b_{ij} \in \{0, 1\} \quad \text{for all } i \in S^+ \cup S^- \text{ and} \\
 & \quad \text{for all } j \in [1, u_i - l_i + 1].
 \end{aligned} \tag{4.3}$$

Observe that the interval  $[l_i, u_i]$  consists of  $u_i - l_i + 1$  integer values for any  $i \in S^+ \cup S^-$ . Hence, in equation (4.3),  $b_{ij}$  is a binary decision variable that takes the value of one if the value of  $y_i$  is equal to the  $j$ th smallest integer value in the interval  $[l_i, u_i]$ , that is,  $l_i + j - 1$ , and zero otherwise. The constraints of equation (4.3) ensure that  $b_{ij}$  follows its definition for all  $i \in S^+ \cup S^-$  and  $j \in [1, u_i - l_i + 1]$ . In addition, the terms

$$\sum_{j=1}^{u_i - l_i + 1} b_{ij} \log(l_i + j - 1)$$

in the objective function of equation (4.3) capture the value of  $\log(y_i)$  for all  $i \in S^+ \cup S^-$ . Note that the term  $\log(l_i + j - 1)$  is a parameter and can be computed in advance. The following remark is helpful.

**REMARK 4.3.** Two advantages of the second approach is that it can handle fractional weights and it can use the power of commercial integer linear programming solvers such as CPLEX or Gurobi, if  $\mathcal{Y}$  is represented by linear constraints involving continuous and/or integer decision variables. However, two weaknesses of the second approach include: (1) it introduces additional binary decision variables which can increase the size of the problem; (2) it can be an approximation approach if  $y_i$  is continuous for  $i \in S^+ \cup S^-$ .

The third approach assumes that  $w_i$  is a positive integer value for all  $i \in S^+ \cup S^-$ . Of course this is not an unreasonable assumption because, by Proposition 3.4, one can multiply the objective function of equation (4.2) by a sufficiently large (positive) number to change all fractional (rational) weights to integer values. So, we can assume that  $w_i \in \mathbb{Z}$  for all  $i \in S^+ \cup S^-$ . The third approach is motivated by this observation

that the NSWP can be reformulated as a geometric-mean optimization problem as follows:

$$\begin{aligned} & \max_{y \in \mathcal{Y}} \quad \gamma, \\ & \text{subject to} \quad 0 \leq \gamma \leq \sqrt[W]{\prod_{i \in S^+ \cup S^-} y_i^{w_i}}, \end{aligned}$$

where  $W = \sum_{i=1}^p w_i$  and  $\gamma$  is a nonnegative decision variable that captures the value of the geometric mean (for optimal solutions). Note that because the problem is in the form of maximization, we must have  $\gamma^W = \prod_{i \in S^+ \cup S^-} y_i^{w_i}$  for any optimal solution. However, the main question is how equation (4.3) should be solved. The underlying idea of the third approach is that equation (4.3) should be reformulated in a form that can be solved by commercial solvers. As stated by Chrahgard et al. [4], based on the work of Ben-Tal and Nemirovski [2], a geometric-mean constraint can be replaced by some linear constraints and second-order cone constraints (and introducing some additional continuous variables). To do so, in the geometric-mean constraint,  $y_i^{w_i}$  should be first replaced by the multiplication of  $w_i$  copies of  $y_i$  for each  $i \in S^+ \cup S^-$ . In other words, for each  $i \in S^+ \cup S^-$ , we can replace  $y_i^{w_i}$  by

$$\prod_{j=\mathcal{W}_{i-1}+1}^{\mathcal{W}_i} \tau_j^0,$$

$\mathcal{W}_i = \sum_{k=1}^i w_k$  and  $\tau_j^0 = y_i$  for all  $j \in [\mathcal{W}_{i-1} + 1, \mathcal{W}_i]$ . Note that the superscript of  $\tau$  is just an index and it does not indicate its power. With this in mind, let  $K$  be the smallest positive integer value such that  $2^K \geq W$ . By introducing some additional nonnegative variables and constraints, equation (4.3) can be reformulated as follows:

$$\begin{aligned} & \max \quad \gamma, \\ & \text{subject to} \quad 0 \leq \gamma \leq \sqrt{\tau_1^{K-1} \tau_2^{K-1}}, \\ & \quad 0 \leq \tau_j^l \leq \sqrt{\tau_{2j-1}^{l-1} \tau_{2j}^{l-1}} \quad \text{for all } j \in [1, 2^{K-l}] \text{ and} \\ & \quad \quad \quad \text{for all } l \in [1, K-1], \tag{4.4} \\ & \quad 0 \leq \tau_j^0 = y_i \quad \text{for all } i \in S^+ \cup S^- \text{ and} \\ & \quad \quad \quad \text{for all } j \in [\mathcal{W}_{i-1} + 1, \mathcal{W}_i], \\ & \quad 0 \leq \tau_j^0 = \gamma \quad \text{for all } j \in [W+1, 2^K], \\ & \quad y \in \mathcal{Y}. \end{aligned}$$

Interested readers may refer to the work of Ben-Tal and Nemirovski [2] for details regarding this transformation. Note that any constraint of the form  $\{u, v, z \geq 0 \mid u \leq \sqrt{vz}\}$

is equivalent to

$$\left\{ u, v, z \geq 0 \mid \sqrt{u^2 + \left(\frac{v-z}{2}\right)^2} \leq \frac{v+z}{2} \right\},$$

which is a *second-order cone* constraint.

**REMARK 4.4.** An advantage of the third approach is that it can use the power of commercial integer quadratic programming solvers such as CPLEX or Gurobi if  $\mathcal{Y}$  is represented by linear/quadratic constraints involving continuous and/or integer decision variables. However, a weakness of the third approach is that it requires all weights to be multiplied (at the same time) by a sufficiently large positive value to become integer and this can result in introducing many second-order cone constraints that can increase the size of the problem. For example, if  $p = 2$ ,  $w_1 = 0.01$  and  $w_2 = 1$ , then the weights should be multiplied by 100, and we will have  $w_1 = 1$  and  $w_2 = 100$ . So, in that case, we will have  $K = 7$  and, therefore, 127 second-order cone constraints need to be added.

**REMARK 4.5.** Based on Assumption 4.1, if  $\mathcal{Y}$  does not involve any integer decision variables, then equation (4.4) can be solved in polynomial time, because it is a convex program.

Finally, it is worth mentioning that some other effective (but sophisticated) approaches have been developed for solving the NSWP in recent years. Those approaches fall under the category of “solution methods for optimization over the Pareto-optimal solutions,” because they are based on this observation that the NSWP is trying to return a Pareto-optimal solution that maximizes the Nash social welfare function. In other words, those approaches try to find a Pareto-optimal point in each iteration and then check its optimality for the NSWP. If it is not optimal, then they modify the search and find a different Pareto-optimal point. To the best of our knowledge, those approaches are shown to be competitive with (and even better than) all the approaches mentioned above when  $\mathcal{Y}$  is represented by only linear constraints involving continuous and/or integer decision variables. Interested readers may refer to the papers by Charkhgard et al. [4], Saghand and Charkhgard [16, 17], Saghand et al. [18] and Vazirani [21] to learn more about those approaches. Specifically, a detailed computational study comparing different solution approaches was given by Saghand and Charkhgard [17].

## 5. Final remarks

In this paper, we introduced four main challenges when dealing with single- or multi-objective optimization problems including computational barrier, confusion (or nonexistence) of decision makers, ignoring unsupported Pareto-optimal solutions and fairness/equity. To overcome the challenges, we proposed to develop a game-theoretic mathematical program, named as the NSWP, for any single- or multi-objective

optimization problem. We showed that there are several effective techniques to solve the NSWP. To see the effectiveness of the NSWP in different fields, interested readers may refer to the works of Sierra-Altamiranda et al. [20] in conservation planning/ecology, Acuna et al. [1] and Mendoza-Alonzo et al. [12] in healthcare, and Melendez et al. [11] in energy. Overall, we hope that the simplicity and advantages of the proposed approaches encourage more practitioners as well as researchers to use the NSWP in practice.

### Acknowledgement

The first author of this paper has been working on the concept of the NSWP since 2013. He introduced this concept in his multi-objective optimization course in Spring 2017 and the technique became quite popular among many engineering PhD students/researchers at the University of South Florida from different fields in a short period of time. This motivated the authors to write this short article for disseminating the idea of the NSWP in the optimization community. Therefore, the authors would like to thank all the colleagues that employed the NSWP in their research, because they were indeed the true motivation for writing this short article.

### References

- [1] J. A. Acuna, J. L. Zayas-Castro and H. Charkhgard, “Ambulance allocation optimization model for the overcrowding problem in US emergency departments: a case study in Florida”, *Socio-Econ. Plann. Sci.* **71** (2019) 100747; doi:[10.1016/j.seps.2019.100747](https://doi.org/10.1016/j.seps.2019.100747).
- [2] A. Ben-Tal and A. Nemirovski, “On polyhedral approximations of the second-order cone”, *Math. Oper. Res.* **26** (2001) 193–205; doi:[10.1287/moor.26.2.193.10561](https://doi.org/10.1287/moor.26.2.193.10561).
- [3] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah and J. Wang, “The unreasonable fairness of maximum Nash welfare”, *Proc. 2016 ACM Conf. Economics and Computation, EC’16* (ACM, New York, 2016) 305–322; <https://doi.org/10.1145/2940716.2940726>.
- [4] H. Charkhgard, M. Savelsbergh and M. Talebian, “A linear programming based algorithm to solve a class of optimization problems with a multi-linear objective function and affine constraints”, *Comput. Oper. Res.* **89** (2018) 17–30; doi:[10.1016/j.cor.2017.07.015](https://doi.org/10.1016/j.cor.2017.07.015).
- [5] H. Charkhgard, M. Savelsbergh and M. Talebian, “Nondominated Nash points: application of biobjective mixed integer programming”, *4OR* **16** (2018) 151–171; doi:[10.1007/s10288-017-0354-2](https://doi.org/10.1007/s10288-017-0354-2).
- [6] R. Cole and V. Gkatzelis, “Approximating the Nash social welfare with indivisible items”, *SIAM J. Comput.* **47** (2018) 1211–1236; doi:[10.1137/15M1053682](https://doi.org/10.1137/15M1053682).
- [7] M. Ehrgott, *Multicriteria optimization*, 2nd edn (Springer, New York, 2005) doi:[10.1007/3-540-27659-9](https://doi.org/10.1007/3-540-27659-9).
- [8] J. M. Jorge, “An algorithm for optimizing a linear function over an integer efficient set”, *European J. Oper. Res.* **195** (2009) 98–103; doi:[10.1016/j.ejor.2008.02.005](https://doi.org/10.1016/j.ejor.2008.02.005).
- [9] E. Kalai, “Nonsymmetric Nash solutions and replications of 2-person bargaining”, *Internat. J. Game Theory* **6** (1977) 129–133; doi:[10.1007/BF01774658](https://doi.org/10.1007/BF01774658).
- [10] M. Mariotti, “Fair bargains: distributive justice and Nash bargaining theory”, *Rev. Econ. Stud.* **66** (1999) 733–741; doi:[10.1111/1467-937X.00106](https://doi.org/10.1111/1467-937X.00106).
- [11] K. A. Melendez, V. Subramanian, T. K. Das and C. Kwon, “Empowering end-use consumers of electricity to aggregate for demand-side participation”, *Appl. Energy* **248** (2019) 372–382; doi:[10.1016/j.apenergy.2019.04.092](https://doi.org/10.1016/j.apenergy.2019.04.092).

- [12] J. Mendoza-Alonzo, J. L. Zayas-Castro and H. Charkhgard, “Office-based and home-care for older adults in primary care: a comparative analysis using the Nash bargaining solution”, *Socio-Econ. Plann. Sci.* **69** (2019) 100710; doi:[10.1016/j.seps.2019.05.001](https://doi.org/10.1016/j.seps.2019.05.001).
- [13] H. Moulin, *Fair division and collective welfare* (The MIT Press, Cambridge, MA, 2003); doi:[10.7551/mitpress/2954.001.0001](https://doi.org/10.7551/mitpress/2954.001.0001).
- [14] J. F. Nash, “The bargaining problem”, *Econometrica* **18** (1950) 155–162; doi:[10.2307/1907266](https://doi.org/10.2307/1907266).
- [15] S. Rao, “Game theory approach for multiobjective structural optimization”, *Comput. & Struct.* **25** (1987) 119–127; doi:[10.1016/0045-7949\(87\)90223-9](https://doi.org/10.1016/0045-7949(87)90223-9).
- [16] P. G. Saghand and H. Charkhgard, “A criterion space search algorithm for mixed integer linear maximum multiplicative programs: a multiobjective optimization approach”, *Int. Trans. Oper. Res.* **29** (2022) 1659–1687; doi:[10.1111/itor.12964](https://doi.org/10.1111/itor.12964).
- [17] P. G. Saghand and H. Charkhgard, “Exact solution approaches for integer linear generalized maximum multiplicative programs through the lens of multi-objective optimization”, *Comput. Oper. Res.* **137** (2022) 105549; doi:[10.1016/j.cor.2021.105549](https://doi.org/10.1016/j.cor.2021.105549).
- [18] P. G. Saghand, H. Charkhgard and C. Kwon, “A branch-and-bound algorithm for a class of mixed integer linear maximum multiplicative programs”, *Comput. Oper. Res.* **101** (2019) 263–274; doi:[10.1016/j.cor.2018.08.004](https://doi.org/10.1016/j.cor.2018.08.004).
- [19] R. Serrano, “Fifty years of the Nash program 1953–2003”, *Investigaciones Econ.* **29** (2005) 219–258; doi:[10.2139/ssrn.724233](https://doi.org/10.2139/ssrn.724233).
- [20] A. Sierra-Altamiranda, H. Charkhgard, M. Eaton, J. Martin, S. Yurek and B. J. Udell, “Spatial conservation planning under uncertainty using modern portfolio theory and Nash bargaining solution”, *Ecol. Model.* **423** (2020) 109016; doi:[10.1016/j.ecolmodel.2020.109016](https://doi.org/10.1016/j.ecolmodel.2020.109016).
- [21] V. V. Vazirani, “The notion of a rational convex program, and an algorithm for the Arrow–Debreu Nash bargaining game”, *J. ACM* **59** (2012) 1–36; doi:[10.1145/2160158.2160160](https://doi.org/10.1145/2160158.2160160).