



**HAL**  
open science

# Anytime Algorithms for Adaptive Robust Optimization with OWA and WOWA

Nadjet Bourdache, Patrice Perny

► **To cite this version:**

Nadjet Bourdache, Patrice Perny. Anytime Algorithms for Adaptive Robust Optimization with OWA and WOWA. 5th International Conference on Algorithmic Decision Theory (ADT 2017), Oct 2017, Luxembourg, Luxembourg. pp.93-107, 10.1007/978-3-319-67504-6\_7. hal-01663776

**HAL Id: hal-01663776**

**<https://hal.science/hal-01663776>**

Submitted on 14 Dec 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Anytime Algorithms for Adaptive Robust Optimization with OWA and WOWA

Nadjet Bourdache and Patrice Perny

Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6  
CNRS, UMR 7606, LIP6, F-75005, Paris, France  
4 Place Jussieu, 75005 Paris, France  
nadjet.bourdache@gmail.com, patrice.perny@lip6.fr

**Abstract.** We consider optimization problems in graphs where the utilities of solutions depend on different scenarios. In this context, we study incremental approaches for the determination of robust solutions, i.e. solutions yielding good outcomes in all scenarios. Our approach consists in interleaving adaptive preference elicitation methods aiming to assess the attitude of the Decision Maker towards robustness or risk with combinatorial optimization algorithms aiming to determine a robust solution. Our work focuses on the use of ordered weighted average (OWA) and weighted ordered weighted average (WOWA) to respectively model preferences under uncertainty and risk while accounting for the idea of robustness. These models are parameterized by weighting coefficients or weighting functions that must be fitted to the value system of the Decision Maker. We introduce and justify anytime algorithms for the adaptive elicitation of these parameters until a robust solution can be determined. We also test these algorithms on the robust assignment problem.

**Keywords:** Robust optimization, Preference Elicitation, OWA, WOWA, Ranking Algorithms, Assignment Problem.

## 1 Introduction

The practice of decision support in complex environments has shown the importance of developing new models and algorithms for optimization under partial information. Uncertainty is pervasive in discrete optimization and various contributions concern robust optimization problems [11] in which multiple instances of the same problem must be considered simultaneously. These instances correspond to possible scenarios and the goal is to determine a feasible solution that remains as good as possible in all scenarios.

In graph optimization, several problems have been revisited under this perspective. For example, assuming that uncertainty only impacts the valuation of the graph (and not on its structure), several contributions address the discrete scenario case, and propose various reformulations of shortest-path problems, assignment problems, minimum spanning tree problems, as a min-max or min-max regret optimization problems, see [11, 15, 29, 28, 1, 5]. For the same problems, a

similar work has been carried out in the case of graphs valued with interval data, which corresponds to an infinity of possible scenarios [11, 27, 2, 14, 1, 24]. In this paper we consider the discrete scenario case and propose new (interactive) approaches for robust optimization, with an implementation on the assignment problem. Let us introduce an instance of the robust assignment problem, for illustrative purpose:

**Example 1** *We consider a problem with 4 items that must be assigned to 4 agents (one item per agent and one agent per item). The utility of every item for every agent depends on the context which remains unknown. Three scenarios  $\{s_1, s_2, s_3\}$  are considered leading to the following three utility matrices:*

$$U_1 = \begin{pmatrix} 10 & 3 & 6 & 4 \\ 10 & 1 & 0 & 1 \\ 9 & 9 & 4 & 0 \\ 4 & 2 & 3 & 4 \end{pmatrix} \quad U_2 = \begin{pmatrix} 10 & 5 & 0 & 2 \\ 6 & 7 & 1 & 4 \\ 7 & 0 & 7 & 3 \\ 1 & 9 & 10 & 2 \end{pmatrix} \quad U_3 = \begin{pmatrix} 0 & 2 & 8 & 0 \\ 2 & 0 & 6 & 5 \\ 4 & 9 & 4 & 2 \\ 7 & 8 & 3 & 9 \end{pmatrix}$$

where  $U_s$  is the utility matrix giving in row  $i$  and column  $j$  the utility  $u_{ij}^s$  of item  $j$  for agent  $i$  in scenario  $s$ , for  $i = 1, \dots, 4$ ,  $j = 1, \dots, 4$ ,  $s = 1, \dots, 3$ . We want to find an assignment that remains as good as possible in all possible scenarios.

Any solution of an  $n \times n$  assignment problem is characterized by a one-to-one mapping  $\alpha$  defined from the set of agents to the set of items and associating item  $\alpha(i)$  to agent  $i$  for  $i = 1, \dots, n$ . Equivalently assignment  $\alpha$  will be represented by the set of arcs  $\{(i, \alpha(i)), i = 1, \dots, n\}$  in the bi-partite assignment graph. Given an assignment  $\alpha$ , its total utility in scenario  $s$  is denoted as  $u_s(\alpha)$  and defined by  $u_s(\alpha) = \sum_{i=1}^n u_{i\alpha(i)}^s$ ,  $s = 1, \dots, n$ . If  $q$  distinct scenarios are considered, then any assignment  $\alpha$  is characterized by the utility vector  $u(\alpha) = (u_1(\alpha), \dots, u_q(\alpha))$  representing the possible utilities in the  $q$  scenarios. For example, in the problem described in Example 1, the assignment  $\alpha = \{(1, 3), (2, 1), (3, 2), (4, 4)\}$  leads to the utility vector  $u(\alpha) = (29, 8, 28)$  whereas  $\alpha' = \{(1, 3), (2, 4), (3, 1), (4, 2)\}$  leads to the utility vector  $u(\alpha') = (18, 20, 25)$ . Hence the comparison of  $\alpha$  and  $\alpha'$  amounts to comparing utility vectors  $u(\alpha)$  and  $u(\alpha')$ .

In order to be able to compare the utility vectors attached to feasible solutions, we need to assess the attitude of the Decision Maker (DM) toward uncertainty or risk (the latter situation occurring when the probabilities of the scenarios are known), and to decide how the outcomes attached to the  $q$  scenarios must be aggregated to define the overall value of a solution. One standard criterion for decision making under uncertainty is the *Laplace criterion* that consists in maximizing the average utility taken over all scenarios. In Example 1, the optimal assignment w.r.t. the Laplace criterion is  $\alpha$  with an average of  $65/3$ .

A more cautious approach consists in choosing a solution maximizing the utility measured in its worst scenario. This corresponds to max-min optimization, a standard approach in robust optimization (equivalent to min-max optimization when costs are considered). In Example 1, the max-min optimal assignment is  $\alpha'$  introduced above. It can be seen as more robust than  $\alpha$  since it guarantees a utility greater or equal to 18 whereas we could obtain 8 with  $\alpha$  in scenario  $s_2$ .

However the max-min criterion is often considered as overpessimistic and poorly discriminating. In particular, we may obtain solutions having a much lower average utility than with the Laplace criterion. On the other hand, by compensating high and low utilities, the Laplace criterion does not provide any guarantee on the robustness of solutions. In order to have a better flexibility in modelling the DM's attitude toward uncertainty, and to find compromise attitudes between overpessimism and full compensation, we will use an *ordered weighted average* (OWA) to aggregate the utilities obtained in the different scenarios. OWA operators have been introduced by Yager [26] and axiomatically justified in the context of robust discrete optimization in [18]. They are also widely used in fair multiagent optimization for their ability to generate Pareto-optimal solutions with well-balanced profiles [17, 12, 8].

As we shall see later more formally, the OWA of a given vector is a kind of weighted sum where the weights are not attached to positions of components in the vector but to their ranks. This allows the importance attached to good or bad outcomes to be controlled and provides a continuum of attitudes in the aggregation, ranging from the Laplace criterion (modelled by the average) to pure pessimism (modelled by the min). These various attitudes are defined by the weighting vector parameterizing the OWA model.

When the probabilities of the scenarios are known, the scenarios do not play symmetric roles. In this case we will use the WOWA model [22] also known as Yaari's model [25] which is a non-symmetric extension of OWA allowing weights to be attached to components (here scenarios). In the latter model, the DM's attitude toward risk is controlled by a probability weighting function.

Whether OWA or WOWA is used, assessing the weighting parameter of the model is a critical issue. The aim of this paper is to propose an incremental elicitation method for facilitating the parametrization of these models in order to determine solutions that are well fitted to the value system of the DM. Our aim is not to determine precise values of these parameters for the DM, prior to the optimization stage. Instead, we propose to interleave preference queries with the exploration of solutions in order to progressively reduce the uncertainty attached to these weighting parameters until a robust solution can be found. These preference queries consist in asking the DM to compare between pairs of solutions, his preferences can then be easily translated into linear constraints that we can add to our model. By integrating the elicitation to the exploration, we aim to save a large part of the elicitation burden.

The paper is organized as follows: we introduce in Section 2 some background on OWA and WOWA. Then we present in Section 3 a ranking algorithm for the determination of possibly optimal solutions when the weighting parameters of OWA or WOWA are imprecise. In Section 4 we present an incremental elicitation algorithm to determine or approximate an OWA-optimal or WOWA-optimal solution. In Section 5 we implement the proposed approach on the robust assignment problem and present numerical tests, as well as some preliminary results on the robust shortest path problem.

## 2 Models for Robust Optimization

In this section, we consider a general robust optimization problem involving  $q$  distinct scenarios and we discuss the evaluation of solutions represented by vectors of type  $(x_1, \dots, x_q)$  where  $x_i$  represents the utility of solution  $x$  in scenario  $i$ . In particular we recall some background on OWA and WOWA operators.

**OWA Optimization.** In robust optimization problems with discrete scenarios, one basically looks for Pareto-optimal solutions having a well-balanced utility profile. This vision of robustness in the context of uncertainty can be related to the notion of fairness in social choice (the scenarios acting as different agents providing different views on solutions). In particular, the preference  $\succ$  of the DM is expected to satisfy *monotonicity w.r.t.  $\epsilon$ -transfers*, a standard axiom used in inequality measurement that reads as follows: for all  $j, k, j \neq k$ , for all  $\epsilon$  such that  $0 < x_k - x_j < \epsilon$ ,  $(x_1, \dots, x_j + \epsilon, \dots, x_k - \epsilon, \dots, x_q) \succ (x_1, \dots, x_j, \dots, x_k, \dots, x_q)$ .

Moving from a solution to another one using such  $\epsilon$ -transfers contributes to reducing the utility gap between pairs of scenarios and thus makes the DM better off. It is well known that the minimal preference relation satisfying this condition is the Lorenz dominance relation (L-dominance) [13] defined by  $x \succ_L y$  if and only if  $L(x) \succ_P L(y)$  where  $\succ_P$  is the Pareto dominance and  $L(x)$  is the Lorenz vector the  $i^{\text{th}}$  component of which is defined by  $L_i(x) = \sum_{k=1}^i x_{\sigma(k)}$ , where  $\sigma$  is the permutation of  $(1, \dots, q)$  that reorders the components of  $x$  by increasing order ( $x_{\sigma(i)} \leq x_{\sigma(i+1)}, i = 1, \dots, q-1$ ). However, L-dominance is a partial order and many solutions remain incomparable. A natural way to extend this partial order, axiomatically justified in the context of robust optimization [18], is to resort to Ordered Weighted Averages (OWA for short).

OWA is an aggregation function that weights the components of a vector in function of their rank. Let  $w \in \mathbb{R}_+^q$  be a weighting vector. The OWA defined by  $w$  reads:  $f(x, w) = \sum_{i=1}^q w_i x_{\sigma(i)}$ . This function is symmetric because the weights are not attached to the components of  $x$  but to the components of the reordered vector  $(x_{\sigma(1)}, \dots, x_{\sigma(q)})$ . The OWA family of aggregation functions  $f(x, w), w \in \mathbb{R}_+^q$  includes the minimum, the maximum, the median and all order statistics as particular cases.

OWA is widely used in fair optimization because it enables a linear extension of the Lorenz Dominance order. Remark indeed that  $x_{\sigma(i)} = L_i(x) - L_{i-1}(x)$  for all  $i > 1$ . Hence we have:  $f(x, w) = \sum_{i=1}^{q-1} (w_i - w_{i+1}) L_i(x) + w_q L_q(x)$ . Thus, function  $f$  is nothing else but a linear combination of the components of the Lorenz vector. Then, if the weights are decreasing (i.e.  $w_i > w_{i+1}$  for all  $i = 1, \dots, q-1$ ), then the coefficients  $w_i - w_{i+1}$  are strictly positive. Therefore, an  $f$ -optimal solution is necessary  $L$ -non-dominated. Thus, OWA used with strictly decreasing weights  $w_i$  leads to Pareto-optimal solutions that cannot be improved in terms of  $\epsilon$ -transfers reducing inequalities. This favours solutions having a well-balanced utility vector. Considering the properties of OWA recalled above, robust optimization under uncertainty can soundly be reformulated as the problem of maximizing  $f(x, w)$  for some  $w$  with decreasing components.

**Example 1 (continued)** Let us compare solutions  $\alpha$  and  $\alpha'$  such that  $u(\alpha) = (29, 8, 28)$  and  $u(\alpha') = (18, 20, 25)$  using  $f(\cdot, w)$  with  $w = (1/2, 1/3, 1/6)$  we obtain  $f(u(\alpha), w) = 8/2 + 28/3 + 29/6 = 18.17$  and similarly  $f(u(\alpha'), w) = 119/6 = 19.83$ . Here, we observe that  $\alpha'$  is preferred to  $\alpha$  because the average loss in utility incurred when passing from  $u(\alpha')$  to  $u(\alpha)$  is compensated by the improvement of the minimum (18 against 8). Remark that, although  $\alpha'$  is the max-min optimal solution in Example 1, it is not OWA-optimal. The OWA-optimal assignment for the chosen weights is  $\alpha'' = \{(1, 1), (2, 4), (3, 2), (4, 3)\}$  with  $u(\alpha'') = (23, 24, 17)$  and  $f(u(\alpha''), w) = 121/6 = 20.17$ . This is a compromise between  $\alpha$  and  $\alpha'$  that improves the average utility of  $\alpha'$  but slightly downgrades the worst case utility.

**WOWA Optimization.** As explained before, one typical property of OWA is to be a symmetric aggregator. This property seems natural when the same attention or importance is attached to every scenario. This is no longer the case when the probabilities  $(p_1, \dots, p_q)$  of the scenarios are known. In such cases we may naturally consider a weighted extension of OWA defined as follows:

$$g(x, \varphi) = \sum_{i=1}^q [x_{\sigma(i)} - x_{\sigma(i-1)}] \varphi \left( \sum_{k=i}^q p_{\sigma(k)} \right) \quad (1)$$

$$= \sum_{i=1}^q \left[ \varphi \left( \sum_{k=i}^q p_{\sigma(k)} \right) - \varphi \left( \sum_{k=i+1}^q p_{\sigma(k)} \right) \right] x_{\sigma(i)} \quad (2)$$

where  $x_{\sigma(0)} = 0$ ; function  $\varphi$  is strictly increasing on the unit interval. A solution  $x$  is as least as good as a solution  $y$  when  $g(x, \varphi) \geq g(y, \varphi)$ . This formulation is known as the *Yaari's model* in the literature on decision under risk because it has been introduced and axiomatically justified by Yaari [25] in this context. Since  $\varphi$  is increasing on the unit interval, the preferences induced by the Yaari's model are monotonic with respect to first-order stochastic dominance (FSD). This means that if two solutions  $x$  and  $y$  are such that the probability  $G_x(t) = P[u(x) > t]$  is greater or equal than the probability  $G_y(t) = P[u(y) \geq t]$  for all  $t$  then  $g(x, \varphi) \geq g(y, \varphi)$ . Moreover, a necessary and sufficient condition for these preferences to be monotonic with respect to second order stochastic dominance (SSD) is the convexity of  $\varphi$ . This means that  $x SSD y$  implies  $g(x, \varphi) \geq g(y, \varphi)$  whenever  $\varphi$  is convex, where  $x SSD y$  means that  $\int_{-\infty}^u G_x(t) dt \geq \int_{-\infty}^u G_y(t) dt$  for all  $u$ . This property established in [9] has a major importance in the context of robust optimization since monotonicity with respect to SSD is the standard model of strong risk aversion. This is due to the fact that SSD is strongly related to the existence of mean-preserving spread of the utility distribution increasing the risk attached to a solution (just as Lorenz dominance relates to the existence of  $\epsilon$ -transfers reducing inequalities); for more details see [21]. For this reason, we shall use convex functions  $\varphi$  to account for risk aversion in robust optimization problem with probabilities over scenarios.

**Example 2** We come back to Example 1 and assume now that the probabilities of the 3 scenarios are given by  $p = (1/2, 1/6, 1/3)$ . Let us compare solutions  $\alpha$  and  $\alpha'$  such that  $u(\alpha) = (29, 8, 28)$  and  $u(\alpha') = (18, 20, 25)$  using Yaari's model

with  $\varphi(x) = x^2$ . We have:  $g(\alpha, \varphi) = 8 + (28 - 8)\varphi(5/6) + (29 - 28)\varphi(1/2) = 22.1$  and  $g(\alpha', \varphi) = 18 + (20 - 18)\varphi(1/2) + (25 - 20)\varphi(1/3) = 19.1$ . Therefore  $\alpha$  is preferred to  $\alpha'$ . The fact that scenario  $s_1$  has a high probability compared to the others gives the advantage to  $\alpha$  even if it is more risky than  $\alpha'$ . Note that the conclusion would be different if the requirement of risk-aversion were strengthened, using a function  $\varphi$  of higher convexity. For example, if  $\varphi(x) = x^4$  we have  $g(\alpha, \varphi) = 17.7$  whereas  $g(\alpha', \varphi) = 18.2$ . In this case  $\alpha'$  is preferred to  $\alpha$ .

Note that when scenarios are equiprobable then  $p_i = 1/q$  and the coefficient of  $x_{\sigma(i)}$  in Equation(2) becomes:  $w_i = [\varphi(\frac{q-i+1}{q}) - \varphi(\frac{q-i}{q})]$  which is now constant (independent of  $\sigma$ ). In this case the Yaari's model reduces to a standard OWA. Hence Yaari's model can be seen as a weighted generalization of OWA which explains the name WOWA due to Torra [22] and used in this paper.

Considering the properties of WOWA recalled above, robust optimization under multiple scenarios of known probabilities can soundly be reformulated as the problem of maximizing  $g(x, \varphi)$  for some proper convex weighting function  $\varphi$ . The determination of an  $f$ -optimal or  $g$ -optimal solution in graph optimization is generally a challenging problem. Standard constructive algorithms based on dynamic programming or greedy search do not apply directly because neither  $f$ -optimality nor  $g$ -optimality satisfy the Bellman principle. An  $f$ -optimal solution for a given vector  $w$  can include sub-optimal subsolutions due to the non-linearity of  $f$  with respect to outcomes. For example, when  $w = \{(1, 0, \dots, 0)\}$ ,  $f$ -optimization is nothing else but  $\sum$ -min maximization ( $\sum$ -max minimization for costs) which is known to be NP-hard for assignment problems but also for shortest-path problems [11, 1]. The same remarks holds for  $g$ -optimality. Yet some solution methods are available both for OWA and WOWA optimization, e.g., [17, 12].

### 3 Optimization with Imprecise Parameters

**Possibly OWA-optimal Solutions.** It is often difficult to precisely determine the weighting vector  $w$  to be used in the OWA model. Indeed, the only prior information we have is that the weighting vector is positive and strictly decreasing, providing an exponential number of possible weights, and then an exponential number of preference queries to be assessed. Yet, in most cases, some preference information is available, putting some constraints on the set of admissible weighting vectors. Note that, any judgment of type "i prefer  $x$  to  $y$ " (where  $x$  and  $y$  are two feasible utility vectors) translates into the following inequality:  $f(x, w) \geq f(y, w)$ , which is a linear constraint in  $w$  bounding the set of admissible weighting vectors  $W$ . Thus, we define the *uncertainty set*  $W$  as a convex polyhedron including all weighting vectors compatible with the preference information collected so far. Given the uncertainty set  $W$  and a set  $X$  of utility vectors attached to feasible solutions, we define  $PO_W(X)$  as the set of possibly  $f$ -optimal solutions in  $X$ , i.e., the elements of  $X$  which are  $f$ -optimal for some weighting vector  $w$  in  $W$ . More formally:  $\forall X \subseteq \mathbb{R}^n, PO_W(X) = \bigcup_{w \in W} \arg \max_{x \in X} f(x, w)$ .

The uncertainty set  $W$  allows a  $W$ -dominance relation over utility vectors to be defined as follows:  $x \succ_W y \iff [\forall w \in W, f(x, w) > f(y, w)]$ . This relation can be extended to set-wise dominance: a solution  $x \in X$  is said to be dominated by a set  $Y \subseteq X$  (denoted  $Y \succ_W x$ ) if there exists  $y \in Y$  such that  $y \succ_W x$ . If  $Y$  is explicitly defined, we can decide whether a solution  $x$  is dominated by  $Y$  in polynomial time by testing whether  $\min_{w \in W} \max_{y \in Y} f(y, w) - f(x, w) > 0$  which is done by solving a linear program. Hence, when  $X$  is given explicitly,  $PO_W(X)$  can be computed in polynomial time. It is sufficient to iteratively remove from  $X$  any solution  $x_i$  dominated by  $X$ , for  $i = 1, \dots, n$ . This was shown in [3] for decision models based on the minimization of a weighted sum. We kept the same idea while adapting the algorithm for the OWA maximization problem.

In robust optimization problems, the set  $X$  is only implicitly defined and computing  $PO_W(X)$  is more difficult. To overcome the problem we now introduce a ranking approach to compute  $PO_W(X)$ . This approach consists of three steps: 1) *linear scalarization*: a scalar valued instance of the problem is constructed by replacing utility vectors attached to the edges of the graph by their average over all scenarios. 2) *ranking*: we perform an enumeration of solutions by decreasing order of utilities; several algorithms are available in the literature to rank the solutions of an optimization problem by decreasing order of preferences, e.g., Murty algorithm [16] for assignment problems or Eppstein algorithm [6] for shortest path problems. 3) *stopping condition*: we stop the enumeration when we can prove that all possibly optimal solutions have been enumerated. The stopping condition used in step 3 is justified by the following propositions:

**Proposition 1** *For any weighting vector  $w \in \mathbb{R}_+^q$  with decreasing weights such that  $\sum_{i=1}^q w_i = 1$ , we have, for all  $x \in \mathbb{R}^q$ ,  $f(x, w) \leq \frac{1}{q} \sum_{i=1}^q x_i$ .*

Proposition 1, that directly derives from the result presented in proposition 3 of [7], allows us to establish the following result:

**Proposition 2** *Let  $X$  be the set of all feasible vectors, and let  $X^k = \{x^1, \dots, x^k\}$  be the list of the  $k$  best elements of  $X$  ordered by decreasing average. We have:  $\max_{x \in PO_W(X^k)} \min_{w \in W} f(x, w) > \frac{1}{q} \sum_{i=1}^q x_i^k \Rightarrow PO_W(X) \subseteq PO_W(X^k)$ .*

*Proof.* We show that when the “if” condition holds at step  $k$ , then any element that does not belong to  $X^k$  cannot be optimal. Let us consider  $x \in X \setminus X^k$ . Then if  $\max_{x \in PO_W(X^k)} \min_{w \in W} f(x, w) > \frac{1}{q} \sum_{i=1}^q x_i^k$  then there exists  $y \in X^k$  such that  $\min_{w \in W} f(y, w) > \frac{1}{q} \sum_{i=1}^q x_i^k$ . Moreover since  $x$  comes after  $x^k$  in the ordered enumeration, we have  $\frac{1}{q} \sum_{i=1}^q x_i^k \geq \frac{1}{q} \sum_{i=1}^q x_i \geq f(x, w)$  for any  $w \in W$ . Hence, for any  $w \in W$  we have  $f(y, w) > f(x, w)$  and therefore  $x \notin PO_W(X)$ . This shows that  $PO_W(X) \subseteq X^k$ . Moreover an element in  $PO_W(X)$  cannot be  $\succ_W$ -dominated in  $X^k$  since  $X^k \subseteq X$ . Hence  $PO_W(X) \subseteq PO_W(X^k)$ .  $\square$

Proposition 2 provides a stopping condition for a ranking algorithm based on the mean of utilities to determine the set  $PO_W(X)$ . If at step  $k$  of the ranking algorithm the condition is fulfilled, then all solutions that would be enumerated



after step  $k$  are  $W$ -dominated by a solution in  $PO_W(X^k)$ . The enumeration can be stopped since the elements we are looking for are all included in  $PO_W(X^k)$  which can easily be computed (as previously explained) since  $X^k$  is explicitly known at step  $k$ . To summarize, when  $X$  is implicitly defined, the set  $PO_W(X)$  can still be determined by ranking its elements by increasing average utilities. This ranking is stopped as soon as the condition described above is activated. Remark that, for the robust version of assignment problem [11, 1], we cannot expect the stopping condition to be activated within a polynomial number of steps since the problem of determining  $PO_W(X)$  is already NP-hard when  $W$  is reduced to the single vector  $\{(1, \dots, 0)\}$ . Nevertheless, we shall see in the section dedicated to numerical tests that the stopping condition is activated after a reasonable number of iterations on average. Moreover, the stopping condition can be relaxed into  $\frac{1}{q} \sum_{i=1}^q x_i^k - \max_{x \in PO_W(X^k)} \min_{w \in W} f(z, w) \leq \delta$  where  $\delta$  is a positive threshold representing the maximum admissible error. This possibly saves multiple iterations while providing good approximations of  $PO_W(X)$ .

**Possibly WOWA-optimal Solutions.** The ranking approach described above can also be adapted to compute  $PO_\Phi(X)$  for  $g$ -optimization under strong risk aversion. In this case  $\varphi$  is assumed to be convex. Consistently with the approach proposed for OWA, we consider now an uncertainty set  $\Phi$  of admissible convex functions (functions compatible with the preferences observed so far). We will show later that, under mild hypothesis,  $\Phi$  can also be represented by a convex polyhedron. Given the uncertainty set  $\Phi$ , we define  $PO_\Phi(X)$  as the set of possibly  $g$ -optimal solutions in  $X$ , i.e., the elements of  $X$  which are  $g$ -optimal for at least one function  $\varphi$  in  $\Phi$ . More formally:  $\forall X \subseteq \mathbb{R}^n$ ,  $PO_\Phi(X) = \bigcup_{\varphi \in \Phi} \arg \max_{x \in X} g(x, \varphi)$ . The associated dominance relation is defined by:  $x \succ_\Phi y \Leftrightarrow [\forall \varphi \in \Phi, g(x, \varphi) > g(y, \varphi)]$ . This relation obviously extends to set-wise dominance. Deciding whether a solution  $y$  is dominated by a set  $X$  amounts to testing whether  $\min_{\varphi \in \Phi} \max_{x \in X} g(x, \varphi) - g(y, \varphi) > 0$  which may be done by linear programming provided that  $\Phi$  is represented by a convex polyhedron. Moreover, when  $X$  is explicitly defined, the set  $PO_\Phi(X)$  can easily be computed by iteratively eliminating dominated elements, as done for  $PO_W(X)$ . Now, in order to install a ranking procedure to determine  $PO_\Phi(X)$  when  $X$  is implicitly defined, we need to establish a counterpart of Propositions 1 and 2 for  $g$ -optimization. This is exactly the role of the two following propositions:

**Proposition 3** *For any convex function  $\varphi : [0, 1] \rightarrow [0, 1]$  such that  $\varphi(0) = 0$  and  $\varphi(1) = 1$  we have, for all  $x \in \mathbb{R}^q$ ,  $g(x, \varphi) \leq \sum_{i=1}^q p_i x_i$ .*

*Proof.* Since  $\varphi$  is convex we have  $\varphi(ta + (1-t)b) \leq t\varphi(a) + (1-t)\varphi(b)$  for all  $a, b, t \in [0, 1]$ . Setting  $a = 1$  and  $b = 0$  we obtain:  $\varphi(t) \leq t$  for all  $t \in [0, 1]$ . Hence  $\varphi(\sum_{k=i}^n p_{\sigma(k)}) \leq \sum_{k=i}^n p_{\sigma(k)}$  for  $i = 1, \dots, q$ . Therefore we have:  $g(x, \varphi) \leq \sum_{i=1}^q [x_{\sigma(i)} - x_{\sigma(i-1)}] \sum_{k=i}^q p_{\sigma(k)}$  by Equation (1) since  $x_{\sigma(i)} - x_{\sigma(i-1)} \geq 0$  for  $i = 1, \dots, q$ . Hence we obtain:  $g(x, \varphi) \leq \sum_{i=1}^q [\sum_{k=i}^q p_{\sigma(k)} - \sum_{k=i+1}^q p_{\sigma(k)}] x_{\sigma(i)} = \sum_{i=1}^q p_{\sigma(i)} x_{\sigma(i)} = \sum_{i=1}^q p_i x_i$ .  $\square$

**Proposition 4** *Let  $X$  be the set of all feasible vectors and  $X^k = \{x^1, \dots, x^k\}$  be the list of the  $k$  best elements of  $X$  ordered by decreasing expected utility. Let  $p_1, \dots, p_q$  be the probabilities of the  $q$  scenarios. The following property holds:  $\max_{x \in PO_W(X^k)} \min_{\varphi \in \Phi} g(z, \varphi) > \sum_{i=1}^q p_i x_i^k \Rightarrow PO_\Phi(X) \subseteq PO_\Phi(X^k)$ .*

The proof is very similar to the one of Proposition 2 and is deliberately omitted. To complete the parallel with the approach proposed for computing  $PO_W(X)$ , when  $X$  implicitly defined, the set  $PO_\Phi(X)$  can be determined by ranking the solutions of the instance obtained by replacing the utility vectors attached to the edges of the graph by their expected utility. This approach applies to problems for which a ranking algorithm is available, in particular, assignment problems, shortest path and minimum spanning tree problems.

## 4 Interleaving Elicitation and Ranking

**Incremental Elicitation.** As shown in Example 1, preferences induced by  $f(\cdot, w)$  or  $g(\cdot, \varphi)$  models may be sensitive to variations of their respective parameters,  $w$  and  $\varphi$ . It is therefore necessary to design elicitation procedures aiming to reduce the uncertainty set  $W$  (resp.  $\Phi$ ) introduced in the previous section. The elicitation of these parameters may require numerous preference queries if it is performed independently on the problem instance to be solved. For this reason, it is preferable to interleave elicitation and search. We suggest inserting preference queries in the ranking algorithm presented above in order to progressively enrich the set of preference statements and the list of constraints defining  $W$  (resp.  $\Phi$ ). This will iteratively reduce the  $W$  (resp.  $\Phi$ ), and therefore the set  $PO_W(X)$  (resp.  $PO_\Phi(X)$ ) until the obtention of a necessarily optimal solution, i.e., a solution that is  $f$ -optimal (resp.  $g$ -optimal) for all remaining parameter values in the uncertainty set. This incremental elicitation process should save a large part of the elicitation burden since an optimal solution can be identified although the parameters of the models remain largely imprecise.

**Regret Minimization.** We want to design an *anytime* algorithm that can return a valid solution to the problem even if it is interrupted before it ends. To make such a recommendation upon request at any step of the algorithm we use a standard regret based elicitation approach [23] based on the following definitions:

$$\text{PMR}(x, y, W) = \max_{w \in W} \{f(y, w) - f(x, w)\} \quad (3)$$

$$\text{MR}(x, X, W) = \max_{y \in X} \text{PMR}(x, y) \quad (4)$$

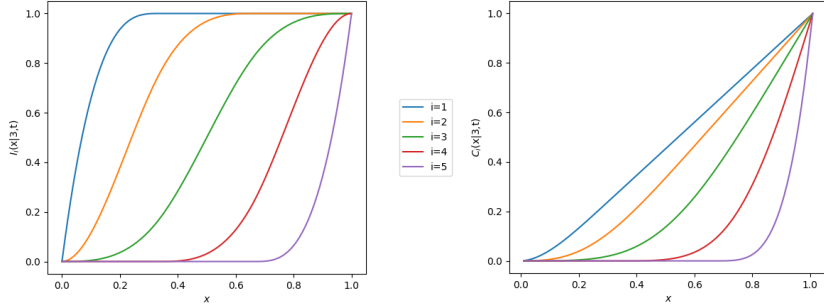
$$\text{MMR}(X, W) = \min_{x \in X} \text{MR}(x, X, W) \quad (5)$$

The pairwise regret  $\text{PMR}(x, y, W)$  is the maximum regret of choosing  $x$  instead of  $y$ , defined as the maximum gap of OWA values. When  $X$  is explicitly known, this regret can easily be computed by linear programming since  $f$  is linear in  $w$ . The maximum regret (MR) attached to a solution  $x$  is the maximum regret of choosing  $x$  instead of any other solution. Finally, the minimax regret (MMR)

is the minimal MR regret over  $X$ . If the algorithm is stopped at a given step with a set  $X$  of possibly optimal solutions, then we shall recommend an element  $x$  in  $X$  that achieves the MMR. This solution will be named the MMR solution hereafter. As suggested in [23], these regrets can also be used to select informative preference queries during an incremental elicitation process to iteratively reduce the MMR to zero. An efficient strategy introduced in [4] under the name the *Current Solution Strategy* (CSS for short) consists in asking the DM to compare the current MMR solution  $x^*$  with its strongest challenger defined by  $y^* = \arg \max_{y \in X} \text{PMR}(x^*, y, W)$ . Whatever the answer to this query, a new constraint will be derived, further restricting the set  $W$ .

A similar approach could be implemented for WOWA optimization, using regrets  $\text{PMR}(x, y, \Phi)$ ,  $\text{MR}(x, X, \Phi)$  and  $\text{MMR}(X, \Phi)$  that simply derive from (3-5) by substituting  $f(\cdot, w)$  by  $g(\cdot, \varphi)$ . However the optimization of such regrets might be challenging because we have to optimize over a continuous set of weighting functions. To overcome this problem, we use a spline representation of function  $\varphi$ . Spline functions are piecewise polynomials whose elements connect with a high degree of smoothness. They are widely used in data interpolation due to their ability to approximate complex shapes [20]. Interestingly enough, spline functions can be generated by linear combinations of basis spline functions. This allows to reduce the elicitation of a spline function to the determination of its weights in the spline basis. The use of spline representations for function  $\varphi$  in WOWA model has been recently introduced in [19]. It enables an efficient incremental elicitation of the model to describe DM's preferences over probabilistic distributions. The proposed construction relies on the definition of  $\varphi$  as a convex combination of  $m$  basis spline functions of degree 3, increasing from 0 to 1 on the unit interval, and known as I-spline functions [20]. More precisely we have:  $\varphi(x) = \sum_{j=1}^m b_j I_j(x)$  where  $I_j(x), j = 1, \dots, m$  are the basic spline functions (see [19, 20] for a formal definition of  $I_j$ ).

Note however that this construction does not completely fit to our context because we have the additional constraint that  $\varphi$  must be convex, in order to enforce strong risk aversion, as explained in the previous section. To overcome this problem, we use another spline basis to generate spline functions that are both increasing and convex on the unit interval. To this end,  $\varphi(x)$  is defined by  $\varphi(x) = \sum_{j=1}^m \phi_j C_j(x), j = 1, \dots, m$  where  $C_j$  are C-spline functions defined as the normalized integrals of the I-spline functions. More precisely:  $C_j(x) = \int_0^x I_j(t) dt / \int_0^1 I_j(t) dt$ . As the integrals of positive and increasing functions,  $C_j$  functions are increasing and convex. Moreover we have  $C_j(0) = 0$  and  $C_j(1) = 1$  for all  $j$ . Therefore,  $\varphi(x)$  will also be increasing and convex since coefficients  $\phi_j$  will be constrained to be non-negative. This is the model we use hereafter because  $g$  defined in this way is a linear function of coefficients  $\phi_j, j = 1, \dots, m$ , a key property for regret optimization. Hence, any preference constraint of type  $g(x, \varphi) \geq g(y, \varphi)$  translates into a linear equation in coefficients  $\phi_j$ ; thus  $\Phi$  can be soundly defined as the convex polyhedron of vectors  $(\phi_1, \dots, \phi_m)$  compatible with the preference statements collected so far. We give below on Figure 1 the I-spline basis and the associated C-spline basis used to generate  $\varphi$ .



**Fig. 1.** I-spline and C-spline cubic functions  $I_j(x)$  and  $C_j(x)$  for  $m = 5$

**Adaptative Elicitation in a Ranking Algorithm.** Our adaptive elicitation procedure of OWA's weights uses progressive reductions of MMR values to discriminate between the solutions generated by the ranking algorithm. The baseline of our algorithm is a ranking procedure enumerating the solutions by decreasing average utilities (or increasing average costs). During the enumeration, if the current solution is not dominated by the previous ones, it is inserted in a bag of solution named  $X$ . When  $|X| = sb$ , preferences queries are generated to discriminate between the element of  $X$  and  $W$  is reduced accordingly until  $\text{MMR}(X, W) = 0$ . Then all solutions with a strictly positive MR value are removed from  $X$ . The process is iterated until the stopping condition introduced in Proposition 2 holds. Due to this proposition, we know that, at this point,  $X$  includes all possibly optimal solutions. Moreover, since there exists  $x^* \in X$  such that  $\text{MR}(x^*, X, W) = 0$  (since  $\text{MMR}(X, W) = 0$ ),  $x^*$  dominates all other solutions in  $X$  for all  $w \in W$  and is therefore necessarily optimal. This is the solution returned by the algorithm. The pseudo-code is given in Algorithm 1.

Function  $\text{next}(G, k)$  generates the  $k^{\text{th}}$  best solution  $x^k$  in the ranking process and returns its outcome vector (one component per scenario) or the empty vector if there are no solutions left. Function  $\text{MMR}(X, W)$  returns the best MR value in  $X$  for the uncertainty set  $W$ . We keep asking queries, applying CSS, until the MMR becomes zero. The stopping condition of the ranking process holds when  $\text{current-distance} = 0$  where  $\text{current-distance}$  is defined in line 20. So  $\delta$  should be set to 0. However, in practice, the stopping condition can be relaxed by using a positive tolerance threshold  $\delta$  in line 4 in order to save many iterations. More generally the algorithm can be stopped at any step  $k$  of the ranking (anytime property). The current MMR solution  $x^*$  will be returned as the current best solution. It is necessarily optimal within the set  $\{x^1, \dots, x^k\}$  of solutions enumerated so far. Moreover, all solutions coming after  $x^k$  in the ranking have an  $f$ -value lower than the average utility of  $x^k$ . Hence  $\text{current-distance}$  provides an upper bound on the gap to optimality in the case of an early interruption of the ranking process. Consequently, when  $\text{current-distance}$  is less than  $\delta$ , the gap to optimality for  $x^*$  is at most  $\delta$ .

**Algorithm 1:** MMR based elicitation in ranking for OWA optimization

---

**Input:**  $G = (V, E)$  undirected graph;  $u_i(e) \geq 0$  edge utility in scenario  $i$ ;  
 $q$ : number of scenarios,  $sb$ : size of bags;  $\delta$ : tolerance threshold  
**Output:**  $x^*$ : a near-optimal solution

- 1  $X \leftarrow \{\}, W \leftarrow \{w \in \mathbb{R}_+^q : \sum_{i=1}^q w_i = 1\}, x^1 \leftarrow \text{next}(G, 0), k \leftarrow 2,$
- 2  $\text{current-distance} \leftarrow \frac{1}{q} \sum_{i=1}^q x_i^k - \min_{w \in W} f(x^1, w)$
- 3  $\text{enumeration-completed} \leftarrow \text{false}$
- 4 **while** ( $\text{current-distance} > \delta$  **and**  $\text{enumeration-completed} = \text{false}$ ) **do**
- 5      $x^k \leftarrow \text{next}(G, k - 1)$
- 6     **if**  $x^k = \text{emptyVector}$  **then**  $\text{enumeration-completed} \leftarrow \text{true}$
- 7     **else**
- 8         **if**  $\text{not}(X \succ_W \{x^k\})$  **then**
- 9              $X \leftarrow X \cup \{x^k\}$
- 10            **if**  $|X| = sb$  **and**  $MMR(X, W) > 0$  **then**
- 11                **repeat**
- 12                    ask a preference query to the DM (selected by CSS)
- 13                    restrict  $W$  according to the answer
- 14                    update regrets PMR, MR and MMR
- 15                    **until**  $MMR(X, W) = 0$
- 16                     $X \leftarrow \{x \in X : MR(x, W) = 0\}$
- 17                **end**
- 18            **end**
- 19             $\text{current-distance} \leftarrow \frac{1}{q} \sum_{i=1}^q x_i^k - \max_{x \in X} \min_{w \in W} f(x, w)$
- 20             $k \leftarrow k + 1$
- 21     **end**
- 22 **end**
- 23 **return**  $x^* = \arg \min_{x \in X} MR(x, X, W), \text{current-distance}$

---

**Example 1 (continued)** *Let us briefly illustrate the behavior of Algorithm 1 on Example 1. We use bags of size 5 ( $sb = 5$ ) and simulate the answers of the DM using an OWA with  $w = (1/2, 1/3, 1/6)$ . The best solution according to the mean value is  $x^1 = \{(1, 3), (2, 1), (3, 2), (4, 4)\}$  with utility vector  $u(x^1) = (29, 8, 28)$  and a mean at 21.67. Then, a lower bound of the OWA value of  $x^1$  is obtained by minimizing  $f(u(x^1), w) = 8w_1 + 28w_2 + 29w_3$  over all possible weighting vectors. We obtain 8, hence  $\text{current-distance} = 21.67 - 8 > 0$  and another iteration is necessary. The ordered enumeration continues until step 5 where the algorithm computes the MMR value which is strictly positive. So a preference query is asked to the DM: the MMR solution  $x^2 = \{(1, 1), (2, 4), (3, 2), (4, 3)\}$  such that  $u(x^2) = (23, 24, 17)$  must be compared to its best challenger  $x^4$  chosen as explained before and such that  $u(x^4) = (18, 20, 25)$ . The DM prefers  $x^2$  therefore  $x^4$  is removed from  $X$  and the constraint  $f(x^2, w) \geq f(x^4, w)$  is added to the definition of polyhedron  $W$  i.e.  $17w_1 + 23w_2 + 24w_3 \geq 18w_1 + 20w_2 + 25w_3$  or equivalently  $-w_1 + 3w_2 - w_3 \geq 0$ . At this time, the MMR is still positive so the algorithm asks another query, this time between  $x^2$  and  $x^1$  (chosen in the same*

way as  $x^4$ ), the DM still prefers  $x^2$ , the algorithm proceeds similarly, removing  $x^1$  and adding a constraint according to this preference. After this second question, the MMR is equal to 0, therefore all solutions but  $x^2$  are removed from  $X$ . Then the ranking algorithm continues until step 15 without inserting new solutions in  $X$  because all of them are dominated by  $x^2$ . At step 15, the stopping condition is activated and the algorithm returns the optimal solution  $x^2$ .

A variant of Algorithm 1 can be used for WOWA optimization. It is sufficient to replace  $f$  by  $g$  and to modify the definition of regrets accordingly. The definition of the current distance must also be adapted not only by substituting  $f$  by  $g$  but also by replacing the weighted average by  $\sum_{i=1}^q p_i x_i^k$ . The correctness of this variant derives from Proposition 4 instead of Proposition 2.

## 5 Numerical Tests

We have implemented Algorithm 1 on the robust assignment problem using both OWA and WOWA models. For these tests, function  $next(G, k)$  (line 5) was implemented with Murty’s algorithm [16]. The complexity of Murty’s algorithm to rank assignments by increasing utility is  $\mathcal{O}(Kn^4)$  for  $n$  agents and  $K$  enumerations. We used the Gurobi library of Python to solve the linear programs required for dominance tests and regret minimization. During the elicitation steps, the DM’s answers to preference queries are simulated using a hidden OWA or WOWA model. We evaluate the performance of the algorithm in terms of computation time, number of preference queries and number of ranking steps, we performed tests on multiple instances of different size and number of scenarios. For every case, performances are averaged over 20 runs. The tests are performed on a Intel Core i7-4770 CPU with 11GB of RAM. Table 1 shows the results obtained for OWA and WOWA elicitation and optimization. The performance are obtained for an error threshold  $\delta$  set to 10% of the initial error (obtained for solution  $x^1$ ), with a bag size of 10 and a timeout of 20 minutes (1200s). Time is given in seconds in the tables and the gap is the maximal error attached to the returned solution, expressed as a percentage of the range of the valuation scale.

**Table 1.** Tests for OWA and WOWA optimization on the robust assignment problem

Number of agents		q = 3				q = 5				q = 10			
		time	steps	queries	gap	time	steps	queries	gap	time	steps	queries	gap
OWA	20	274	13607	1.3	0	719	34009	2.3	0.5	940	37739	2	1.4
	60	1200	7236	4	1.1	1200	6945	6.7	5	1200	6382	11.6	4.7
	100	1200	2075	5.1	2	1200	2550	11.3	6	1200	2216	17.5	7.5
WOWA	20	270	5338	10.3	1	475	4657	9.45	1.1	865	6743	9.2	1.7
	60	1200	5446	8.95	8.1	1200	3192	5.3	7.6	1200	4381	5.4	6.5
	100	1200	2978	6.15	10.3	1200	2302	5	15.9	1200	1901	4.9	11.3

To test the generality of the approach, we also made some preliminary tests on the robust shortest path problem. In this problem, the arcs of the graph are valued by cost vectors with  $q$  components corresponding to the cost of the arc in the different scenarios. The definition of OWA and WOWA aggregators as well as Algorithm 1 have been modified to fit to minimization problems. For ranking the paths by increasing average costs, we used a lazy version of Eppstein’s algorithm introduced in [10] with a complexity of  $\mathcal{O}(m + n \log n)$  in the worst case for a graph with  $n$  nodes and  $m$  arcs. To give an idea of the performance of Algorithm 1 on robust shortest path problems, we solve instances including 500 nodes and 5 scenarios in 548 seconds and 5 preference queries with a gap of 0.1 on average (after 28460 steps on average). In the case of 10 scenarios, the algorithm needs 681 seconds on average on similar graphs, 6.3 queries with a gap of 0.1 on average (after 41796 steps on average).

## 6 Conclusion

We have introduced a new adaptive elicitation approach for OWA and WOWA optimization and tested its practical efficiency on robust assignment problems and on robust shortest path problems in the discrete scenarios case. Our approach is quite general and applies to any other optimization problem for which an efficient ranking algorithm is known. An interesting extension of this work would be to design a similar approach for the incremental elicitation of Choquet integrals under the constraint of convex capacity (this is a more general model to account for robustness in optimization under uncertainty, including OWA and WOWA as special cases). The implementation of the ranking approach for Choquet integrals and the definition of a valid stopping condition are challenging questions because the capacity is imprecisely known.

## References

1. Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Complexity of the minmax and minmax regret assignment problems. *OR letters*, 33(6):634–640, 2005.
2. Ionuț D Aron and Pascal Van Hentenryck. On the complexity of the robust spanning tree problem with interval data. *OR Letters*, 32(1):36–40, 2004.
3. Nawal Benabbou and Patrice Perny. Incremental weight elicitation for multiobjective state space search. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1093–1099, 2015.
4. Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713, 2006.
5. Vladimir G Dei and Gerhard J Woeginger. On the robust assignment problem under a fixed number of cost scenarios. *OR Letters*, 34(2):175–179, 2006.
6. David Eppstein. Finding the  $k$  shortest paths. *SIAM Journal on computing*, 28(2):652–673, 1998.
7. Lucie Galand and Olivier Spanjaard. Exact algorithms for owa-optimization in multiobjective spanning tree problems. *Computers & Operations Research*, 39(7):1540–1554, 2012.

8. Tobias Heinen, Nhan-Tam Nguyen, and Jörg Rothe. Fairness and rank-weighted utilitarianism in resource allocation. In *Proceeding of ADT*, pages 521–536, 2015.
9. Chew Soo Hong, Edi Karni, and Zvi Safra. Risk aversion in the theory of expected utility with rank dependent probabilities. *J. of Econ. Theory*, 42(2):370–381, 1987.
10. Víctor M Jiménez and Andrés Marzal. A lazy version of Eppstein’s k shortest paths algorithm. In *International Workshop on Experimental and Efficient Algorithms*, pages 179–191. Springer, 2003.
11. Panos Kouvelis and Gang Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.
12. Julien Lesca and Patrice Perny. Lp solvable models for multiagent fair allocation problems. In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 393–398. IOS Press, 2010.
13. Albert W Marshall, Ingram Olkin, and Barry C Arnold. *Inequalities: theory of majorization and its applications*, volume 143. Springer, 1979.
14. Roberto Montemanni and Luca Maria Gambardella. An exact algorithm for the robust shortest path problem with interval data. *Computers & Operations Research*, 31(10):1667–1680, 2004.
15. Ishwar Murthy and Shenq-Shyong Her. Solving min-max shortest-path problems on a network. *Naval Research Logistics (NRL)*, 39(5):669–683, 1992.
16. Katta G Murthy. An algorithm for ranking all the assignments in order of increasing costs. *Operations Research*, 16(3):682–687, 1968.
17. Włodzimierz Ogryczak. Multicriteria models for fair resource allocation. *Control and Cybernetics*, 36(2), 2007.
18. Patrice Perny, Olivier Spanjaard, and Louis-Xavier Storme. A decision-theoretic approach to robust optimization in multivalued graphs. *Annals of Operations Research*, 147(1):317–341, 2006.
19. Patrice Perny, Paolo Viappiani, and Abdellah Boukhatem. Incremental preference elicitation for decision making under risk with the rank-dependent utility model. In *Proceedings of UAI 2016*, 2016.
20. J. O. Ramsay. Monotone regression spline in action. *Statistical Science*, pages 425–441, 1988.
21. Joseph E Stiglitz and M Rothschild. Increasing risk. i. a definition. *Journal of Economic Theory*, 2(3):225–243, 1970.
22. Vincent Torra. The weighted owa operator. *International Journal of Intelligent Systems*, 12:153–166, 1997.
23. Tianhan Wang and Craig Boutilier. Incremental utility elicitation with the mini-max regret decision criterion. In *proceedings of IJCAI*, pages 309–318, 2003.
24. Wei Wu, Manuel Iori, Silvano Martello, and Mutsunori Yagiura. Algorithms for the min-max regret generalized assignment problem with interval data. In *proc of IEEM*, pages 734–738. IEEE, 2014.
25. Menahem E. Yaari. The dual theory of choice under risk. *Econometrica*, 55:95–115, 1987.
26. Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. In *IEEE Trans. Systems, Man and Cybern.*, volume 18, pages 183–190, 1998.
27. Hande Yaman, Oya Ekin Kardeş, and Mustafa Ç Pınar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29(1):31–40, 2001.
28. Gang Yu. Min-max optimization of several classical discrete optimization problems. *Journal of Optimization Theory and Applications*, 98(1):221–242, 1998.
29. Gang Yu and Jian Yang. On the robust shortest path problem. *Computers & Operations Research*, 25(6):457–468, 1998.