Multiple Objectives and Multiple Actors Load/Resource Dispatching or Priority Setting: Satisficing Game Approach

A. Ph. Tchangani[1,2]
[1]Dept. GEII, IUT de Tarbes, Université Toulouse III - Paul Sabatier
1, rue Lautréamont, 65016 Tarbes Cedex, France.
E-Mail: ayeley.tchangani@iut-tarbes.fr

[2]Laboratoire Génie de Production, Ecole Nationale d'Ingénieurs de Tarbes
47, Avenue d'Azereix - BP 1629 - 65016 Tarbes Cedex, France.
E-Mail: Ayeley.Tchangani@enit.fr

August 26, 2006

Abstract

In this paper, we consider the problem of dispatching a production load of a certain good or a resource (money, energy, water, ...) among a group of production units or alternatives; or simply establishing priority for that group of alternatives when optimizing a number of (possible) antagonist objectives (cost, benefit, environnmental impact, reliability, safety, ...) in pressence of different actors or stakeholders that do not have the same opinion regarding the importance of each objective. The first part of this problem (single stakeholder or a group of stakeholders acting as one person) is a part of more general problems known in the literature as multiple objetives/criteria optimization/decision aid problems that are widely recognized as a framework for realistic and practical decision making by individuals or corporates. There are few works in the literature that combine multiple objectives and multiple stakeholders for modeling decision making problems whereas such problems are common in practice. In this paper we will adopt a hierarchical analysis approach that will go from general purpose objectives to more precise objectives about which stakeholders are able to do one-on-one comparisons in the framework of analytic hierarchy process (AHP) or its network extension (ANP) to determine some relative weights for objectives. These weights are then used in the process of computing the "satisfiability (selectability and rejectability) functions" using satisficing game theory and finally a set of "satisficing" or "good enough" allocations, assignments, dispatchings or settings is defined to be those for which the selectability exceeds the rejectability with regard to a boldeness index. The boldeness index is a parameter that permits decision maker(s) to adjust the size of satisficing set leading to some flexibility. Two real world problems (porfolio management and thermal power dispatch) are solved by this approach to show its applicability.
Keywords: Multiple Objectives Decision Making, AHP, Satisficing Game Theory, Multiple Actors.

1 Introduction

We consider in this paper the problem of assigning a fraction x_i of a resource, a task load or simply a priority to an alternative i of a group of n alternatives. Without loss of generality, the admissible assignments set $\mathcal{X}$ is given by

$$\mathcal{X} = \left\{ x \in \mathbb{R}_+^n : \mathbf{1}^T x = 1,\ x_{\min} \leq x \leq x_{\max} \right\}. \tag{1}$$

where $x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$; $\mathbb{R}_+^n$ is a n dimensional real vector space with non negative components, $\mathbf{1}$ is a column vector with all entries equal to 1; y^T is the transpose of y and the inequalities $x_{\min} \leq x \leq x_{\max}$ are considered componentwise. Each assignment, dispatching or setting $x \in \mathbb{R}_+^n$ is with associated m objectives $f_j(x)$, $j = 1, 2, .., m$ and we suppose that d actors that we designate generically as stakeholders must express their opinion about the relative importance of objectives with regard to the overall assignment goal. The version of this problem without stakeholders intervention is what is typically known in the literature (see for instance [4, 8, 10, 16, 17, 25, 28, 30]) as multicriteria, multiattributes or mutiltiobjectives decision making or decision analysis. In our opinion three issues characterize any practical decision making (choosing, sorting or ranking alternatives or options) problem:

- multiple objectives: an admissible solution of a decision making problem is always chosen to optimize multiple objectives; the classical constrained optimization problems (see [6, 11]) can be seen as multiple objectives optimization problems where some objectives are transformed to constraints;
- uncertainty: it is rarely possible to define precisely objectives and to assign to them infinitely divisible grade; in many cases only the sense of their goodness (larger the better or smaller the better) is known;
- multiple actors (stakeholders): for a number of practical decision making problems, the (antagonist) opinions of many actors have to be taken into account.

Classical (treating stakeholders as a single decision maker) multiple objectives decision making has been used in economics and management science for years and has gradually crept in engineering. Many real-world problems are often formulated in terms of multiple objectives optimization problems, see for instance [4, 8, 10, 16, 17, 21, 25, 28, 30] and references therein. For instance in a production planning problem one wants to maximize the output and minimize the resources utilized. The problem of production planning in a company that produces two types of manufacturing tools where the objectives were to maximize simultaneously the profit and the quantity of one type of tool that has less profit margin under constraints of resource in terms of available time has been considered in [14]. In the domain of mechanical engineering, civil engineering, and material engineering, the design of a structure is a multiple objectives optimization problem in the sense that, it is required in many case to minimize the mass or the volume of the material used and to maximize some index of safety. In [4] the problem of choosing the dimensions of a beam that minimize the mass and minimize the deflection has been considered; this problem is clearly the optimization of two conflicting objectives. Software design and implementation require considerations of many conflicting objectives as minimization of the cost of development, maximization of the speed of the system, minimization of power consumption and the weight of the system mainly in what concern embedded systems design. For embedded systems there is more and more need of co-synthesis and optimization of hardware/software implementation. In Electronics design mainly when one is dealing with VLSI design many conflicting criteria will be being considered simultaneously such as minimization of area occupied by components, minimization of power consumption, maximization of heat evacuation. Other objectives related to environment for instance can be considered [21]. In the following paragraphs, we will review the main approaches that are used to solve classical multiple objectives decision making problems. Most of the time the analyst (the person or expert in charge of establishing the decision model) face intervention of different actors with opposite opinions that must be integrated in the model, this paper addresses these issues.

The remainder of this paper is organized as follows: in the second section some classical approaches used to solve multiple objectives decision making problems are reviewed; the third section (the main contribution of this paper) establishes the satisficing game model for solving the assignment problems presented in the introduction section including the presentation of the basic materials of satisficing decision theory that are relevant to our modeling problem and a brief recall of analytic hierarchy process approach; the section four is devoted to the application of the approach established in the paper to two real world practical problems and concluding remarks are given in the fifth section.

2 Classical approaches for solving multiple objectives optimization problems

Classical approaches for solving multiple objectives decision problems rely on the notion of the so-called Pareto dominance [30] and Pareto-optimal set and the resolution is organized around two processes: search and decision making. Depending on how search (finding a sample of Pareto-optimal set) and decision process are combined, multiple objectives optimization methods can be classified in three categories [30].

- Decision making before search: The objective functions are aggregated into a single objective by using some preference of the decision maker.
- Search before decision making: Here a sample (or totality) of Pareto-optimal set is obtained first and then a choice is made by a decision maker.
- Decision making during search: Here an interactive sequential optimization is performed where after each search step, the decision maker is presented with a number of alternatives.

The first approach to deal with multiple objectives decision making problems has been the aggregation of objectives into a single objective in different ways leading to weighting methods, constraint methods and goal programming methods. The advantage of these methods is that efficient and broad algorithms developed to single objective optimization problems (see [6, 11, 13] and references therein) can be used to solve the resulting problems. The drawback of these techniques is that the subjective intervention of the user is needed to fix weighting factors and it is known [30] that these methods are most of the time not able to finding Pareto-optimal solutions in the case of non convex feasible space. To overcome these drawbacks, new methods have been designed based on evolutionary algorithms, mainly genetic algorithms that are able to generating efficiently Pareto-optimal solutions. In the following paragraphs we will review some basic methods that are used to solve multiple objectives decision making problems.

2.1 Weighting method ([25])

Here the original multiple objectives optimization problem is converted to a single objective optimization problem as (2)

$$\max_x / \min \left(\sum_{j=1}^{m} \omega_j f_j(x) \right) \text{ s.t. } x \in \mathcal{X}. \tag{2}$$

where s.t. stands for "subjected to". The parameters ω_j are called the weights and are most of the time normalized as $\sum_{j=1}^{k} \omega_j = 1$, $\omega_j \geq 0$. By varying the weights ω_j, different Pareto-optimal solutions can be generated.

2.2 Constraint method ([25])

Here $m-1$ objective functions are transformed into constraints and the remaining objective is optimized under these constraints. The resulting single objective optimization problem is (3)

$$\max_x / \min \{f(x) = f_h(x)\} \text{ s.t. } f_i(x) - \delta_i \leq 0,\ 1 \leq i \leq m,\ i \neq h,\ x \in \mathcal{X}. \tag{3}$$

By varying parameters δ_i, different Pareto-optimal solutions are obtained.

Another approach that fall in the framework of transforming a multiple objectives optimization problem into a single objective optimization problem is the so-called goal programming. This technique has had many applications and variant. It is a particular case of method of inequalities where all objectives are transformed into constraints.

2.3 Goal Programming

Goal programming was first introduced by [3] and gained its popularity after the work by [8, 10]. A number of engineering applications where goal programming has been used can be found in [16]. The main idea in the goal programming approach is to find solutions which attain a pre-specified target for one or more objective functions; if there is no solution which achieves targets in all objective functions, the task is then to find solutions which minimize deviations from targets. The specification of the goal can take 4 possibilities. For each objective function $f_j(x)$ the goal may be:

1) less-than-equal-to, $f_j(x) \leq t_j$,

2) greater-than-equal-to, $f_j(x) \geq t_j$,

3) equal-to, $f_j(x) = t_j$,

4) within a range, $f_j(x) \in [t_j^l,\ t_j^u]$.

To tackle this problem, two deviation variables n and p are introduced. For the less-than-equal-to type goal, a positive value p_j is substracted from $f_j(x)$ so that $f_j(x) - p_j \leq t_j$ (if $f_j(x) > t_j$ then p_j must take a positive value and zero if not); for the greater-than-equal-to type goal, a positive value n_j is added to $f_j(x)$ so that $f_j(x) + n_j \geq t_j$ (if $f_j(x) < t_j$ then n_j must take a positive value and zero if not); for the equal-to type goal, we have $f_j(x) - p_j + n_j = t_j$. For the within type goal, two constraints are introduced for each target t_j^l and t_j^u. To solve the resulting problem different techniques are used.

2.3.1 Weighted Goal Programming

Here the problem is reduced to classical optimization problem (4)

$$\min_x \left(\sum_{j=1}^{m} \left(\alpha_j p_j + \beta_j n_j \right) \right) \text{ s.t. } f_j(x) - p_j + n_j = t_j,\ n_j,\ p_j \geq 0\ \forall\ j,\ x \in \mathcal{X}. \tag{4}$$

where α_j and β_j are weighting factors fixed by the user. This is a drawback for this method, as weighting factors may be not easy to chose and this make the method subjective.

2.3.2 Lexicographic Goal Programming

Here goals are categorized into several levels of preemptive priorities; goals of lower-level are infinitely more important than goals of higher level and so are considered first in the solving process. The solving process is then sequential, first goals and corresponding constraints of first level priority are considered in the formulation of goal programming and solved. If there is only one solution then the rest of goals are ignored. If there are many solutions then the goals and corresponding constraints of second level priority are considered with solution of first level as a hard constraint. This process is repeated until only one solution is found.

2.3.3 Minimax Goal Programming

This method is similar to weighted goal programming. The objective to minimize is the maximum deviation in any goal from its target. The optimization problem is (5)

$$\min_x (d) \text{ s.t. } \alpha_j p_j + \beta_j n_j \leq d,\ f_j(x) - p_j + n_j = t_j,\ n_j,\ p_j \geq 0\ \forall\ j,\ x \in \mathcal{X}. \tag{5}$$

The main drawback of previous techniques to deal with multiple objectives optimization is the necessity for the user to intervene by specifying weights and the fact that these techniques in general don't find Pareto solutions for a non-convex decision space.

2.4 Other approaches: outranking and evolutionary algorithms

Other approaches that are considered in the multiple objectives decision aid community are dominated by outranking approaches where a partial order of alternatives is derived by an interactive procedure between the analyst and the decision maker (see [1, 2, 17, 28]) and the evolutionary algorithms that are a class of stochastic optimization methods that attempt to simulate the process of natural evolution. Evolutionary algorithms have been proved useful in optimizing difficult functions that might mean: non-differentiable objective functions, many local optima, a large number of parameters, or a large number of configurations of parameters [30].

In this paper we consider a modeling approach, known as satisficing dispatching and setting, that differs from classical ones presented previously in two ways: first many stakeholders will be considered and second the stakeholders preferences aggregation is based on the idea that there are two categories of objectives with regard to the setting or dispatching goal, those that behave as "larger is better" and those which are such that "smaller is better"; the aggregation procedure will be carried separately on each group of objectives to obtain the selectability measure and the rejectability measure respectively in the framework of satisficing game theory.

3 Satisficing dispatching and setting

The approach considered in this paper is based on the idea that given setting or dispatching objectives as defined in the introduction section, there are (almost always) those which variation is positively correlated to the overall goal (larger is better) and those which variation is negatively correlated to the goal (smaller is better). The former can be interpreted as delivery and the later as the effort to be furnished to obtain deliveries. By so doing one can establish a setting or dispatching model based on two measures: satisfiability measure $\mu_S(x)$ that aggregate deliveries contributions and the rejectability measure $\mu_S(x)$ that aggregate efforts contributions in the framework of satisficing game theory [26]; this interpretation has been successfully used for production units evaluation by the author in [27]. In the following paragraph we will recall the materials of satisficing game theory that are relevant to our problem. For more details to this theory, see [26].

3.1 Satisficing game theory

The underlying philosophy of most of the techniques used in the literature to construct the evaluation model is the superlative rationality, looking for the best, all the alternatives must be compared against each other. But the superlative rationality paradigm is not necessarily the way humans evaluate alternatives (and maybe not the best one). Most of the time humans content themselves with alternatives that are just "*good enough*" because their cognitive capacities are limited and information in their possession is almost always imperfect that is the fundamental idea behind the theory of bounded rationality that has its roots in the work by H. Simon [23]; the concept of being good enough allows a certain flexibility because one can always adjust its aspiration level. On the other hand, decision makers more probably tend to classify units as good enough or not good enough in terms of their positive attributes (benefit) and their negative attributes (cost) with regard to the decision goal instead of ranking units with regard to each other. For instance, to evaluate cars, we often make a list of positive attributes (driving comfort, speed, robustness, etc.) and a list of negative attributes (price, consumption per kilometer, maintainability, etc.) of each car and then make a list of cars for which positive attributes "exceed" negative attributes in some sense. This way of evaluation falls into the framework of praxeology or the study of theory of practical activity (the science of efficient action). Here decision maker(s), instead of looking for the best options, look for satisficing alternatives. Satisficing is a term that refers to a decision making strategy where options, units or alternatives are selected which are "good enough" instead of being the best [26]. Let us consider a universe U of alternatives; then for each alternative $u \in U$, a *selectability function* $\mu_S(u)$ and a *rejectability function* $\mu_R(u)$ are defined to measure the degree to which u works towards success in achieving the decision maker's goal and costs associated with this alternative respectively. This pair of measures called *satisfiability functions or measures* are *mass functions* (they have the mathematical structure of the probabilities [26]): they are non negative and sum to one on U. The following definition then gives the set of options arguable to be "good enough" because for these options, the "benefit" expressed by the function μ_S exceeds the cost expressed by the function μ_R with regard to the index of boldness q.

Definition 1 *The satisficing set $\Sigma_q \subseteq U$ with the index of boldness q is the set of alternatives defined by equation (6)*

$$\Sigma_q = \{u \in U : \mu_S(u)(u) \geq q\mu_R(u)\}\,. \tag{6}$$

The boldness index q can be used to adjust the aspiration level: increase q if Σ_q is too large or on the contrary decrease q if Σ_q is empty for instance.

Applying the satisficing game theory to the setting and dispatching problem return then to determining satisfiability measures $\mu_S(u)$ and $\mu_R(u)$; the process of determining these measures will be considered in the following paragraphs.

3.2 Defining satisfiability measures

As stated previously, we argue that for a multiple objectives optimization problem as that of setting and/or dispatching considered in this paper, it is possible to divide the set of objectives $\mathcal{O}$ into two groups:

- objectives that acts in the sense of optimization goal (larger is better); we denote their set by $\mathcal{O}_S$ and the equation (7) gives the vector representation of these objective functions

$$F_S(x) = \begin{bmatrix} f_1(x) & f_2(x) & . & . & f_{|\mathcal{O}_S|}(x) \end{bmatrix}^T, \ f_i \in \mathcal{O}_S \tag{7}$$

where $|\mathcal{O}_S|$ stands for the cardinality of $\mathcal{O}_S$;

- objectives that acts in the opposite sense of optimization goal which set is denoted by $\mathcal{O}_R$ and their vector representation given by equation (8)

$$F_R(x) = \begin{bmatrix} f_1(x) & f_2(x) & . & . & f_{|\mathcal{O}_R|}(x) \end{bmatrix}^T, \ f_i \in \mathcal{O}_R. \tag{8}$$

We consider this partition to be beyond the scope of this paper and is accepted by the stakeholders. To definitely define the satisfiability measures, one needs to integrate stakeholders opinion about the importance of objectives; this is done using the analytic hierarchy process approach [19] that is briefly recalled in the following paragraph.

3.2.1 Brief recall of AHP

The analytic hierarchy process is a comprehensive, powerful and flexible decision making process to help people set priorities and make the best decision when both qualitative and quantitative aspects are used to evaluate alternatives, see [19, 20]. By reducing complex decisions to a series of one-on-one comparisons, then synthesizing the results, AHP not only helps decision makers arrive at the best decision, but also provides a clear rationale that it is the best. It is designed to reflect the way people actually think and is a widely used decision-making theory. The basic AHP decomposes a decision problems in different elements, grouped in clusters, that it arranges in a linear hierarchy form where the top element of the hierarchy is the overall goal of the decision making and is based on the following axioms (see [18]).

- Axiom 1 (reciprocity): if element A is x times as important than element B, then element B is $\frac{1}{x}$ times as important as element A.
- Axiom 2 (homogeneity): only comparable elements are compared. Homogeneity is essential for comparing similar things, as errors in judgement become larger when comparing widely disparate elements.
- Axiom 3 (independence): the relative importance of elements at any level does not depend on what elements are included at a lower level.
- Axiom 4 (expectation): the hierarchy must be complete and include all the criteria and alternatives in the subject being studied. No criteria and alternatives left out and no criteria and alternatives are included.

The hierarchy goes from the general to more particular until a level of operational criteria against which the decision alternatives can be evaluated is reached. The elements of cluster C_c in a top down hierarchy are pairwise compared with regard to each element of the cluster C_{c-1} to obtain a $n_c \times n_{c-1}$ weighting matrix $\mathbf{W}_c$ where n_i is the number of elements in the cluster C_i. This matrix is given by equation (9)

$$\mathbf{W}_c = \begin{bmatrix} w_c^1 & w_c^2 & ... & w_c^{n_{c-1}} \end{bmatrix} \tag{9}$$

where w_c^i are n_c column vectors obtained as follows: for each element i of the cluster C_{c-1}, a pairwise comparison matrix $\mathbf{W}_c^i$ of elements of cluster C_c is constructed by answering questions of the form "how important is element X compared to the element Y of the cluster C_c with regard to upper level element Z of the cluster C_{c-1} ?" using the scales given by the following Table I (see [19], [20])

Verbal scale	Numerical values
Equally important	1
Moderately more important	3
Strongly more important	5
Very strongly more important	7
Extremely more important	9
Intermediate scales (compromise)	2, 4, 6, 8

Table I: scales for AHP comparison procedure.

Once this matrix is constructed, the vector w_c^i is computed as the unique eigenvector of this matrix associated with eigenvalue n_c, that is the solution of the equation (10)

$$\mathbf{W}_c^i w_c^i = n_c w_c^i \tag{10}$$

and a consistency[1] index is computed for possible modification of comparison weights (see [20]). The overall weights vector $\boldsymbol{\omega}$ of bottom cluster, that is the alternatives cluster, with regard to the decision goal is then given by the following equation (11)

$$\boldsymbol{\omega} = \mathbf{W}_N \mathbf{W}_{N-1} ... \mathbf{W}_1 \mathbf{W}_0 \tag{11}$$

where $\mathbf{W}_0$ is a column vector representing the comparison weights of the first cluster with regard to the overall decision goal and $\mathbf{W}_N$ (N is the number of clusters) is the comparison matrix of alternatives with regard to the direct upper level cluster elements (measurable criteria). A typical AHP structure of the setting and dispatching problem we are considering here is given by the directed graph of Figure 1 where the arrows mean dependency; elements that are pointed by an arrow are evaluated against the elements from which the arrow emanate. Let

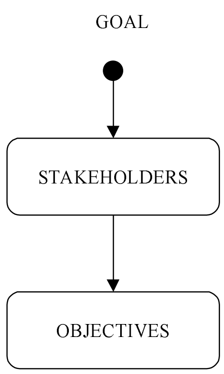

Figure 1: The AHP architecture of the problem under consideration

us denote by $\mathbf{W}_S^k$ (respect. $\mathbf{W}_R^k$), for $k = 1, 2, .., d$, the pairwise comparison matrix of the objectives in $\mathcal{O}_S$ (respect. of the objectives in $\mathcal{O}_R$) according to the stakeholder k that supplies weights ω_{ij}^k (how important is objective i compared to the objective j according to the stakeholder k ?). One rapid way to obtain this matrix is to ask the stakeholder to choose a pivot objective p and compare other objectives to it using the standard AHP scales by supplying weights ω_{ip}^k (how important is objective i compared to the pivot objective p according to stakeholder k ?). Then one constructs a consistent comparison matrix $\mathbf{W}_\times^k$ (where $\times$ stands for S or R) using the relations defined by equation (12)

$$\mathbf{W}_\times^k(i,i) = 1,\ \mathbf{W}_\times^k(i,p) = \omega_{ip}^k,\ \mathbf{W}_\times^k(p,i) = \frac{1}{\omega_{ip}^k},\ \mathbf{W}_\times^k(i,j) = \mathbf{W}_\times^k(i,l) \cdot \mathbf{W}_\times^k(l,j). \tag{12}$$

[1] A comparison matrix M is said to be consistent if it verifies: $M_{ii} = 1$, $M_{ji} = \frac{1}{M_{ij}}$ and $M_{ik} = M_{ij} M_{jk}$.

Once these matrices are obtained, the column weight vectors $\boldsymbol{\omega}_S^k$ and $\boldsymbol{\omega}_R^k$ for objectives according to each stakeholder k are computed as shown by equation (13)

$$\boldsymbol{\omega}_S^k(i) = \frac{1}{|\mathcal{O}_S|}\sum_j \left(\frac{\mathbf{W}_S^k(i,j)}{\sum_l \mathbf{W}_S^k(l,j)}\right) \text{ and } \boldsymbol{\omega}_R^k(i) = \frac{1}{|\mathcal{O}_R|}\sum_j \left(\frac{\mathbf{W}_R^k(i,j)}{\sum_l \mathbf{W}_R^k(l,j)}\right), \tag{13}$$

and the $|\mathcal{O}_S| \times d$ (respect. $|\mathcal{O}_R| \times d$) weighting matrix $\mathbf{W}_S$ (respect. $\mathbf{W}_R$) is given by equation (14)

$$\mathbf{W}_S = \begin{bmatrix} \boldsymbol{\omega}_S^1 & \boldsymbol{\omega}_S^2 & \ldots & \boldsymbol{\omega}_S^d \end{bmatrix} \text{ and } \mathbf{W}_R = \begin{bmatrix} \boldsymbol{\omega}_R^1 & \boldsymbol{\omega}_R^2 & \ldots & \boldsymbol{\omega}_R^d \end{bmatrix} \tag{14}$$

where the weight $\boldsymbol{\omega}_S(i)$ for the objective i that will be used for computing the selectability/rejectability measures is determined by (15)

$$\boldsymbol{\omega}_S(i) = \frac{1}{d}\sum_{k=1}^{d} \mathbf{W}_S(i,k) \ \&\ \boldsymbol{\omega}_R(i) = \frac{1}{d}\sum_{k=1}^{d} \mathbf{W}_R(i,k). \tag{15}$$

As we dispose with the stakeholders opinions in terms of weighting vectors $\boldsymbol{\omega}_S$ and $\boldsymbol{\omega}_R$, we are ready to define the satisfiability measures $\mu_S(x)$ and $\mu_R(x)$ for each setting or dispatching x; that will be done in the following paragraph.

3.2.2 Satisfiability measures

Now we are ready to define these measures; first of all we have to normalize the objectives functions. The normalization of the original objective functions is necessary before weighting because objectives are not, in general, expressed in the same units (money, memory capacity, human resources, surface, machines, qualitative, etc.). Let us then define the normalized column vectors (utilities) $F_S^n(x)$ of the objectives corresponding to the selectability and $F_R^n(x)$ corresponding to the rejectability objectives by

$$F_S^n(x) = \begin{bmatrix} f_1^n(x), & f_2^n(x), & \ldots, & f_{|\mathcal{G}_S|}^n(x) \end{bmatrix}^T, \ f_i \in \mathcal{G}_S \tag{16}$$

$$F_R^n(x) = \begin{bmatrix} f_1^n(x), & f_2^n(x), & \ldots, & f_{|\mathcal{G}_R|}^n(x) \end{bmatrix}^T, \ f_i \in \mathcal{G}_R. \tag{17}$$

There is not a unique way to define normalized function $f_i^n(x)$ that can be interpreted as utility associated with objective function f_i; but as the utilities are unique only up to a positive affine transformation (see for instance [22]), to ensure comparability of utilities, both the scale and the zero point need to be chosen, so we consider the following normalization scheme (18)

$$f_i^n(x) = \frac{f_i(x) - f_{i,\min}}{f_{i,\max} - f_{i,\min}}. \tag{18}$$

where

$$f_{i,\max} = \max_{x\in\mathcal{X}} f_i(x) \text{ and } f_{i,\min} = \min_{x\in\mathcal{X}} f_i(x). \tag{19}$$

Notice that the values $f_{i,\max}$ and $f_{i,\min}$ will exist as the set $\mathcal{X}$ is a compact closed set and the objective functions f_i will be considered to be continuous functions. The following definition then gives the way to obtain the satisfiability functions or measures μ_S and μ_R on $\mathcal{X}$.

Definition 2 *The selectability measure μ_S and the rejectability measure μ_R of the setting and dispatching problem are given by (20)*

$$\mu_S(x) = \frac{\boldsymbol{\omega}_S^T F_S^n(x)}{\int_{\mathcal{X}} \boldsymbol{\omega}_S^T F_S^n(x)dx} \text{ and } \mu_R(x) = \frac{\boldsymbol{\omega}_R^T F_R^n(x)}{\int_{\mathcal{X}} \boldsymbol{\omega}_R^T F_R^n(x)dx}. \tag{20}$$

Notice that these measures define probability density functions over the compact closed set $\mathcal{X}$ and so fulfill the requirements of satisficing game theory. The following paragraph presents different approaches to select a setting or dispatching vector x arguable to be satisficing or good enough.

3.3 Satisficing or good enough setting/dispatching

The satisficing or good enough setting/dispatching vectors x at the boldness index q_1 are given by equation (21)

$$x \in \mathcal{X} : \mu_S(x) \geq q_1 \mu_R(x) \Leftrightarrow \boldsymbol{\omega}_S^T F_S^n(x) \geq q_1 \frac{\int_{\mathcal{X}} \boldsymbol{\omega}_S^T F_S^n(x)dx}{\int_{\mathcal{X}} \boldsymbol{\omega}_R^T F_R^n(x)dx} \boldsymbol{\omega}_R^T F_R^n(x) = q \boldsymbol{\omega}_R^T F_R^n(x) \tag{21}$$

so that the satisficing set Σ_q at the boldness index $q = q_1 \frac{\int_{\mathcal{X}} \boldsymbol{\omega}_S^T F_S^n(x)dx}{\int_{\mathcal{X}} \boldsymbol{\omega}_R^T F_R^n(x)dx}$ is defined by equation (22)

$$\Sigma_q = \left\{ x \in \mathcal{X} : \boldsymbol{\omega}_S^T F_S^n(x) - q \boldsymbol{\omega}_R^T F_R^n(x) \geq 0 \right\} \tag{22}$$

and a particular satisficing setting or dispatching $x \in \Sigma_q$ can be calculated by solving the following optimization problem (23)

$$\min_x (0) \text{ s.t. } -\boldsymbol{\omega}_S^T F_S^n(x) + q \boldsymbol{\omega}_R^T F_R^n(x) \leq 0, \; x \in \mathcal{X} \tag{23}$$

that can be solved using a general purpose software such as Matlab^{TM} with Optimization Toolbox or writing one's own code. The final setting/dispatching can then be selected using different criteria such as the following.
.

- Most selectable setting/dispatching x^* defined by (24)

$$x^* = \arg\max_x \left(\boldsymbol{\omega}_S^T F_S^n(x) \right) \text{ s.t. } -\boldsymbol{\omega}_S^T F_S^n(x) + q \boldsymbol{\omega}_R^T F_R^n(x) \leq 0, \; x \in \mathcal{X}. \tag{24}$$

- Least rejectable setting/dispatching x^* defined by (25)

$$x^* = \arg\min_x \left(\boldsymbol{\omega}_R^T F_R^n(x) \right) \text{ s.t. } -\boldsymbol{\omega}_S^T F_S^n(x) + q \boldsymbol{\omega}_R^T F_R^n(x) \leq 0, \; x \in \mathcal{X}. \tag{25}$$

- Maximal discriminant setting/dispatching x^* defined by (26)

$$x^* = \arg\max_x \left(\boldsymbol{\omega}_S^T F_S^n(x) - q \boldsymbol{\omega}_R^T F_R^n(x) \right) \text{ s.t. } -\boldsymbol{\omega}_S^T F_S^n(x) + q \boldsymbol{\omega}_R^T F_R^n(x) \leq 0, \; x \in \mathcal{X}. \tag{26}$$

- Maximum boldness setting/dispatching x^* defined by (27)

$$x^* = \arg\max_x \left(\frac{\boldsymbol{\omega}_S^T F_S^n(x)}{\boldsymbol{\omega}_R^T F_R^n(x)} \right) \text{ s.t. } x \in \mathcal{X}. \tag{27}$$

- Other criteria: for instance one or a combination of original objectives $o(x)$ may be optimized (maximized or minimized) subjected to satisficing condition, that is

$$x^* = \arg\max_x / \min \left(o(x) \right) \text{ s.t. } -\boldsymbol{\omega}_S^T F_S^n(x) + q \boldsymbol{\omega}_R^T F_R^n(x) \leq 0, \; x \in \mathcal{X}. \tag{28}$$

A summary of the important steps to apply the approach established in this paper is presented in the following paragraph.

3.4 Summary

In summary, the approach presented so far to solve a setting or dispatching problem is organized around 5 steps as presented in the following.

1. Considering the partition of objectives into selectable objectives and rejectable objectives as done, do an AHP analysis with stakeholders to obtain weight vectors $\boldsymbol{\omega}_S$ and $\boldsymbol{\omega}_R$.

2. Compute the maximum and the minimum values $f_{j,\max}$ and $f_{j,\min}$ on $\mathcal{X}$ of each objective function f_j by solving mathematical programming problems (29).

$$\max_{x\in\mathcal{X}} / \min f(x). \tag{29}$$

3. Define normalized objectives or utilities $f_i^n(x) = \frac{f_i(x)-f_{i,\min}}{f_{i,\max}-f_{i,\min}}$.
4. Find the maximum boldness index $q_{\max}$ by solving the problem (27).
5. Choose a boldness index $q \in \left[\ 0,\ \ q_{\max}\ \right]$ and solve one of the problems (23) - (28) and implement the solution or possibly reconsider the boldness index or the weighting vectors until a good solution is obtained.

In the following section, two real world applications will be considered to show the potential applicability of the approach presented in this paper.

4 Applications

Two applications will be considered in this section: the first application is related to portfolio management and the second one is concerned by power load dispatching among thermal power plants.

4.1 Portfolio management

Modern theory of portfolio management has been initiated in 1952 by Markowitz [12] when he proposed his mean-variance model for selection purpose. According to this theory, any portfolio investor should seek the optimization of two conflicting criteria: maximize the mean return and minimize the risk measured by the variance of this return. Thus, managing a portfolio is a multiple objectives or attributes decision problem [7] and actually an efficient management of portfolio must consider more than two conflicting criteria because each firm is determined by different performance indices that must be optimized by portfolio manager when selecting firms for investment. Investment decision is made in two stages: at the first stages the portfolio manger selects a set of firms from a stock exchange database for instance and in second stage solves an affectation problem that is which proportion of his fund will be invested in each selected firm. Two main performance indices are used in practice: financial performance indices, namely:

- current ratio (**CR**), an index related to cash that is a "larger is better" objective;
- return on equity (**ROE**), a "larger is better" objective measuring capital profitability;
- cash flow over liability ratio (**CFLR**), a creditworthiness behaving in the sense of "larger is better";

and stock exchange performance indices given by:

- earnings per share (**EPS**), a "larger is better" index;
- monthly mean return (**MMR**), a "larger is better" objective;
- $\beta - 1$ (**Beta-1**), a technical coefficient which absolute value is a "smaller is better" objective;
- price earning ratio (**PER**), a "smaller is better" objective.

In summary, we have 5 objectives working toward selectability, that is

$$\mathcal{O}_S = \{\mathbf{CR,\ ROE,\ CFLR,\ EPS,\ MMR}\} \tag{30}$$

and 2 objectives that contribute to the rejectability

$$\mathcal{O}_R = \{\boldsymbol{\beta} - \mathbf{1},\ \mathbf{PER}\}\,. \tag{31}$$

For simulation, we consider data of 31 firms of a certain stock exchange (extracted from [7]) given on the following Table II . Notice that the intention here is not a comparative study but rather a proof of a possible use of our approach for the prioritizing and allocating fund for firms in a portfolio.

	Selectability objectives					Rejectability objectives	
Firms	**CR**	**ROE**	**CFLR**	**EPS**	**MMR**	$\boldsymbol{\beta - 1}$	**PER**
X_{01}	1.23	0.147	4.91	9538	0.095	0.300	0.8097
X_{02}	1.36	0.098	0.74	518	0.011	0.106	−8.0645
X_{03}	0.85	0.141	0.19	600	0.019	0.013	32.2581
X_{04}	0.97	0.118	0.69	328	0.009	0.158	22.7273
X_{05}	1.63	0.230	0.26	10762	0.024	0.066	11.6279
X_{06}	1.72	0.241	0.64	105	0.010	0.101	52.6316
X_{07}	0.89	0.163	0.52	68	−0.002	0.070	20.8333
X_{08}	1.10	0.212	0.88	1312	0.013	0.031	7.4627
X_{09}	1.31	0.202	1.72	2335	0.012	0.097	9.4340
X_{10}	1.57	0.137	0.58	1018	0.018	0.079	13.5135
X_{11}	0.82	0.171	0.88	639	0.003	0.156	19.6078
X_{12}	1.28	0.177	0.31	86	0.110	5.740	2.2883
X_{13}	1.58	0.216	0.32	217	−0.001	0.131	30.3030
X_{14}	1.41	0.186	0.24	168	−0.001	0.205	−22.2222
X_{15}	1.07	0.181	0.19	2651	0.007	0.017	58.8235
X_{16}	1.10	0.177	1.01	859	0.017	0.140	11.1111
X_{17}	2.60	0.164	0.51	25	0.005	0.002	13.1579
X_{18}	1.06	0.114	0.34	212	−0.001	0.176	9.2308
X_{19}	1.43	0.299	1.66	294	0.010	0.090	19.2308
X_{20}	1.04	0.064	0.71	168	0.001	0.084	18.1818
X_{21}	1.87	0.104	0.31	235	0.002	0.059	16.3934
X_{22}	0.68	−0.57	0.96	−88	−0.015	0.288	2.3866
X_{23}	0.64	0.150	0.23	316	0.011	0.064	−0.5851
X_{24}	2.48	0.150	9.41	371	0.005	0.350	13.6986
X_{25}	1.91	0.066	4.87	127	−0.006	0.417	55.5556
X_{26}	0.43	0.112	0.82	176	0.005	0.656	333.3333
X_{27}	0.44	0.075	1.36	139	0.008	0.808	25.0000
X_{28}	0.74	0.025	2.99	125	0.002	0.192	13.3333
X_{29}	2.88	0.172	3.67	1485	0.004	0.016	10.1010
X_{30}	2.31	0.163	0.62	3155	0.042	0.207	20.8333
X_{31}	0.85	0.152	1.31	687	0.010	0.172	12.3457

Table II: Data for portfolio management application

In [7] two methods were used for the first stage (selection of firms where to invest) purpose: MINORA that uses interactive UTA algorithms (see [9]) for preference breakup and an outranking method ELECTRE TRI (see [29]) and finally ADELAIS (see [24]), an interactive method for multiple objectives linear programming is used for second stage process.

By solving problem (27) with these data and the equal importance assumption for objectives, we find that the maximum boldness index is $q_{\max} = 15.9531$. By solving problem (23) we find that for a small index of boldness q (less than 3), all the firms have the same priority of 1/31 and the following Table III presents the priority index and the rank for each firm for $q = 5$, 10 & 15; the numbers in brackets in the rank column correspond to the rank obtained in [7] where the authors had an initial preferences of a portfolio manager. We can see that as the index of boldness increase less and less firms have a priority greater than 0 (what is conform to theory because as the index of boldness increase, the satisficing set tend to empty set) and without any qualitative knowledge, our approach performs well. By asking an expert portfolio manager to express its opinion about objectives by doing a pairwise comparison, the results obtained here will be surely improved. The portfolio manager may use the approach presented here to select a group of firms (for instance the top X firms) and then allocate fund proportionally to the priority index within the group.

q	5		10		15	
	priority	**rank**	**priority**	**rank**	**priority**	**rank**
X_{01}	0.0408	01 (01)	0.1624	02 (01)	0.0000	16 (01)
X_{02}	0.0370	06 (14)	0.1206	05 (14)	0.1073	02 (14)
X_{03}	0.0328	22 (20)	0.0000	21 (20)	0.0000	14 (20)
X_{04}	0.0328	23 (17)	0.0000	25 (17)	0.0000	08 (17)
X_{05}	0.0390	02 (03)	0.1247	04 (03)	0.0000	27 (03)
X_{06}	0.0300	27 (27)	0.0000	13 (27)	0.0000	31 (27)
X_{07}	0.0332	19 (23)	0.0000	27 (23)	0.0000	12 (23)
X_{08}	0.0362	09 (10)	0.0664	07 (10)	0.0000	11 (10)
X_{09}	0.0364	08 (06)	0.0515	09 (06)	0.0000	17 (06)
X_{10}	0.0355	12 (09)	0.0196	10 (09)	0.0000	18 (09)
X_{11}	0.0332	20 (24)	0.0000	31 (24)	0.0000	09 (24)
X_{12}	0.0004	30 (25)	0.0000	27 (25)	0.0000	05 (25)
X_{13}	0.0329	21 (16)	0.0000	26 (16)	0.0000	10 (16)
X_{14}	0.0377	04 (18)	0.1635	01 (18)	0.8927	01 (18)
X_{15}	0.0306	26 (07)	0.0000	15 (07)	0.0000	23 (07)
X_{16}	0.0350	14 (12)	0.0010	12 (12)	0.0000	30 (12)
X_{17}	0.0369	07 (08)	0.0715	06 (08)	0.0000	19 (08)
X_{18}	0.0328	24 (21)	0.0000	22 (21)	0.0000	04 (21)
X_{19}	0.0352	13 (13)	0.0000	18 (13)	0.0000	28 (13)
X_{20}	0.0334	18 (19)	0.0000	26 (19)	0.0000	15 (19)
X_{21}	0.0349	15 (15)	0.0000	20 (15)	0.0000	07 (15)
X_{22}	0.0307	25 (31)	0.0000	23 (31)	0.0000	21 (31)
X_{23}	0.0355	11 (30)	0.0651	08 (30)	0.0000	25 (30)
X_{24}	0.0373	05 (02)	0.0185	11 (02)	0.0000	29 (02)
X_{25}	0.0295	28 (11)	0.0000	30 (11)	0.0000	03 (11)
X_{26}	0.0000	31 (28)	0.0000	28 (28)	0.0000	22 (28)
X_{27}	0.0275	29 (29)	0.0000	14 (29)	0.0000	06 (29)
X_{28}	0.0335	17 (26)	0.0000	29 (26)	0.0000	20 (26)
X_{29}	0.0389	03 (04)	0.1352	03 (04)	0.0000	26 (04)
X_{30}	0.0361	10 (04)	0.0000	19 (04)	0.0000	13 (04)
X_{31}	0.0341	16 (22)	0.0000	24 (22)	0.0000	24 (22)

Table III: Simulation results for portfolio management application

4.2 Power dispatch

Dispatching a demand power load among a number of thermal power generators must be done when paying attention to some objectives such as fuel cost, emission cost , transmission losses cost as well as risk measured by the deviation of the generated power around the demand load. Other objectives such as the reliability and/or the safety of operating as well as economical benefit such as employment must be taken into account. The problem is specified as follows: a random power demand P_D must be satisfy by n thermal power generation plants when ensuring the following objectives.

- Minimization of the expected fuel cost, objective F; in general the fuel cost curve F_i of a thermal generator i is approximated by a quadratic function (see [5]) of generator output power P_i by (32)

$$F_i = a_i P_i^2 + b_i P_i + c_i \qquad [\$/\mathrm{H}] \tag{32}$$

where a_i, b_i and c_i are random parameters independent of P_i. The expected value $\overline{F}_i$ of the fuel cost for the plant i is then given by

$$\overline{F}_i = \overline{a}_i \overline{P}_i^2 + \overline{b}_i \overline{P}_i + \overline{c}_i + \overline{a}_i var(P_i) = \overline{a}_i \left(1 + C_{VP_i}^2\right) \overline{P}_i^2 + \overline{b}_i \overline{P}_i + \overline{c}_i \tag{33}$$

where C_{VP_i} is the coefficient of variation of random variable P_i; the expected total fuel cost $\overline{F}$ is (34)

$$\overline{F} = \sum_{i=1}^{n} \overline{F}_i = \overline{\mathbf{P}}^T \overline{\mathbf{A}} \overline{\mathbf{P}} + \overline{\mathbf{b}}^T \overline{\mathbf{P}} + \overline{\mathbf{c}}^T \mathbf{1} \tag{34}$$

with

$$\begin{aligned}
\overline{\mathbf{A}} &= diag(\overline{a}_i \left(1 + C_{VP_i}^2\right)),\ i = 1, 2, ..., n & (35)\\
\overline{\mathbf{b}} &= \left[\begin{array}{ccccc} \overline{b}_1 & \overline{b}_2 & . & . & \overline{b}_n \end{array}\right]^T, & (36)\\
\overline{\mathbf{c}} &= \left[\begin{array}{ccccc} \overline{c}_1 & \overline{c}_2 & . & . & \overline{c}_n \end{array}\right]^T, & (37)\\
\overline{\mathbf{P}} &= \left[\begin{array}{ccccc} \overline{P}_1 & \overline{P}_2 & . & . & \overline{P}_n \end{array}\right]^T & (38)
\end{aligned}$$

where $diag(\alpha_i)$, $i = 1, 2, ..., n$ is a n dimensional square diagonal matrix with the coefficients of the principal diagonal equal to α_i.

- Minimization of the expected emission cost (emission of CO2, NOx and SO2), objective E; this cost is also considered to be a quadratic function of the form (39), see [5]

$$E_i = d_i P_i^2 + e_i P_i + f_i \qquad [\text{Kg/H}] \tag{39}$$

so that the total expected emission cost is given by (40)

$$\overline{E} = \sum_{i=1}^{n} \overline{E}_i = \overline{\mathbf{P}}^T \overline{\mathbf{D}} \overline{\mathbf{P}} + \overline{\mathbf{e}}^T \overline{\mathbf{P}} + \overline{\mathbf{f}}^T \mathbf{1} \tag{40}$$

where $\overline{\mathbf{D}}$, $\overline{\mathbf{e}}$ and $\overline{\mathbf{f}}$ are defined similar to $\overline{\mathbf{A}}$, $\overline{\mathbf{b}}$ and $\overline{\mathbf{c}}$ respectively.

- Minimization of the expected transmission loss $\overline{L}$; the transmission lost L is in general expressed by a simple approximate expression using B-coefficient (see [5]) as (41)

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j \qquad [\text{MW}] \tag{41}$$

where B_{ij} and P_i are independent random variables; the expected transmission loss is given by (42)

$$\overline{L} = \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{P}_i \overline{B}_{ij} \overline{P}_j + \sum_{i=1}^{n} \overline{B}_{ii} var(P_i) = \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{P}_i \overline{B}_{ij} \overline{P}_j + \sum_{i=1}^{n} \overline{B}_{ii} C_{VP_i}^2 \overline{P}_i^2 = \overline{\mathbf{P}}^T \overline{\mathbf{B}} \overline{\mathbf{P}} \tag{42}$$

with

$$\overline{\mathbf{B}}(i,j) = \overline{B}_{ij} \text{ for } i \neq j \text{ and } \overline{\mathbf{B}}(i,i) = \overline{B}_{ii} \left(1 + C_{VP_i}^2\right). \tag{43}$$

- Minimization of the expected risk, $\overline{R}$ is defined as the expected value of the deviation (44)

$$\overline{R} = Expected\left\{\left(\overline{P}_D + \overline{L} - \sum_{i=1}^{n} P_i\right)^2\right\} \qquad [\text{MW}^2] \tag{44}$$

which is reducible (with the balance constraint $\sum_{i=1}^{n} \overline{P}_i = \overline{P}_D + \overline{L}$) to (45)

$$\overline{R} = \sum_{i=1}^{n} var(P_i) = \overline{\mathbf{P}}^T \overline{\mathbf{R}} \overline{\mathbf{P}} \text{ with } \overline{\mathbf{R}} = daig(C_{VP_i}^2). \tag{45}$$

- The balance constraint $\sum_{i=1}^{n} \overline{P}_i = \overline{P}_D + \overline{L}$ and mini-max constraints $\overline{P}_{i,\min} \le \overline{P}_i \le \overline{P}_{i,\max}$ define the admissible set (46)

$$\mathcal{X} = \left\{ \overline{\mathbf{P}} \in \mathbb{R}_+^n : \sum_{i=1}^{n} \overline{P}_i = \overline{P}_D + \overline{L},\ \overline{P}_{i,\min} \le \overline{P}_i \le \overline{P}_{i,\max} \right\}. \tag{46}$$

All the objectives, $\overline{F}$, $\overline{E}$, $\overline{L}$, and $\overline{R}$ behave in the sense of "smaller is better" so they work in the sense of rejectability, that is (47)

$$\mathcal{G}_R = \left\{ \overline{F},\ \overline{E},\ \overline{L},\ \overline{R} \right\}. \tag{47}$$

Most of the published works concerning power dispatch problems consider only these objectives whereas in practical situation one has to consider also other objectives such as reliability, safety, economical benefit of operating a plant, ... Let us suppose that one determine a reliability function for each plant i that decrease exponential with the mean output power $\overline{P}_i$ as (48)

$$\mathrm{Re}_i = \exp\left(-\lambda_i \overline{P}_i\right) \tag{48}$$

where λ_i is a positive constant parameter. The overall reliability Re of operating the n plants, is then given by (49)

$$\mathrm{Re} = \prod_{i=1}^{n} \mathrm{Re}_i = \exp\left(-\sum_{i=1}^{n} \lambda_i \overline{P}_i\right), \tag{49}$$

an objective that works in the sense of selectability, that is (50)

$$\mathcal{G}_S = \{\mathrm{Re}\}. \tag{50}$$

In the following paragraph, we consider a practical case of 6 interconnected power generators for simulations purpose.

4.2.1 Simulation results

Let us consider the case of 6 generators network with the following parameters extracted from [5].

- Fuel cost parameters

$$\overline{\mathbf{A}} = \begin{bmatrix} 0.0050 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.010 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.020 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.003 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.015 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.010 \end{bmatrix},\ \overline{\mathbf{b}} = \begin{bmatrix} 2.0 \\ 2.0 \\ 2.0 \\ 1.95 \\ 1.45 \\ 0.95 \end{bmatrix},\ \overline{\mathbf{c}} = \begin{bmatrix} 100 \\ 200 \\ 300 \\ 80 \\ 100 \\ 120 \end{bmatrix}. \tag{51}$$

- Parameters related to emission (of NOx) cost

$$\overline{\mathbf{D}} = 10^{-3} \begin{bmatrix} 0.6573 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5917 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4906 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.378 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4906 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5174 \end{bmatrix},\ \overline{\mathbf{e}} = \begin{bmatrix} -0.05497 \\ -0.05880 \\ -0.05014 \\ -0.03150 \\ -0.05014 \\ -0.05548 \end{bmatrix},\ \overline{\mathbf{f}} = \begin{bmatrix} 4.111 \\ 2.593 \\ 4.268 \\ 5.526 \\ 4.268 \\ 6.132 \end{bmatrix}. \tag{52}$$

- Transmission loss parameters

$$\overline{\mathbf{B}} = 10^{-2}\begin{bmatrix} 0.0200 & 0.0010 & 0.0015 & 0.0005 & 0.0000 & 0.0030 \\ 0.0010 & 0.0300 & -0.0020 & 0.0001 & 0.0012 & 0.0010 \\ 0.0015 & -0.0020 & 0.0100 & 0.0010 & 0.0010 & 0.0008 \\ 0.0005 & 0.0001 & 0.0010 & 0.0150 & 0.0006 & 0.0050 \\ 0.0000 & 0.0012 & 0.0010 & 0.0006 & 0.0250 & 0.0020 \\ 0.0030 & 0.0010 & 0.0008 & 0.0050 & 0.0020 & 0.0210 \end{bmatrix}. \tag{53}$$

Furthermore we consider that $\lambda_1 = \lambda_2 = .. = \lambda_6 = \lambda = 3.72 \times 10^{-4}$ so that Re is constant and equal to (54)

$$\mathrm{Re} = \exp\left(-\sum_{i=1}^{n}\lambda_i\overline{P}_i\right) = \exp\left(-\lambda\sum_{i=1}^{n}\overline{P}_i\right) = \exp\left(-\lambda\left(\overline{P}_D + \overline{L}\right)\right). \tag{54}$$

The maximum boldness index $q_{\max}$ obtained by solving the optimization problem (27) when considering all objectives of each category to be with equal importance is given by the following Table IV.

$\overline{P}_D + \overline{P}_L$	$q_{\max}$
200 [MW]	6.2079
400 [MW]	6.5120
600 [MW]	7.6928

Table IV: Maximum boldness of power dispatch problem
with equal importance objectives

The Table V shows the dispatching results obtained by optimizing different criteria with a boldness index of 1 ($q = 1$); notice that as the selectability measure is constant, the maximally discriminant dispatching and the least rejectable (min μ_R) dispatching are equivalent.

$\overline{P}_D + \overline{P}_L$	**200** [MW]			**400** [MW]			**600** [MW]		
	min $\overline{F}$	min $\overline{E}$	min μ_R	min $\overline{F}$	min $\overline{E}$	min μ_R	min $\overline{F}$	min $\overline{E}$	min μ_R
$\overline{P}_1$	36.44	33.33	33.33	66.67	66.67	66.67	100	100	100
$\overline{P}_2$	25.14	33.33	33.33	66.67	66.67	66.67	100	100	100
$\overline{P}_3$	2.52	33.33	33.33	66.67	66.67	66.67	100	100	100
$\overline{P}_4$	42.66	33.33	33.33	66.67	66.67	66.67	100	100	100
$\overline{P}_5$	32.48	33.33	33.33	66.67	66.67	66.67	100	100	100
$\overline{P}_6$	60.76	33.33	33.34	66.67	66.67	66.67	100	100	100
$\overline{F}$	1287.5	1315	1315	1870	1870	1870	2565.1	2565.1	2565.1
$\overline{E}$	21.31	20.34	20.34	20.72	20.72	20.72	28.05	28.05	28.05
$\overline{L}$	2.35	1.72	1.72	6.86	6.86	6.86	15.44	15.44	15.44
$\overline{R}$	0.85	0.67	0.67	2.67	2.67	2.67	6.00	6.00	6.00

Table V: Results of dispatching for different criteria and $q = 1$

In the paper [5], the surrogate worth trade-off algorithm was used and the results obtained in terms of principal objectives (fuel cost, emission cost, transmission loss and risk) are globally less than those obtained here.

Let us suppose now that the fuel cost is considered to be Extremely more important than transmission loss and risk (deviation) costs and moderately more important than emission cost. In terms of AHP analysis we obtain the following pairwise comparison matrix $\mathbf{W}_R$ and the corresponding weights vector ω_R (55)

$$\mathbf{W}_R = \begin{bmatrix} & \overline{F} & \overline{E} & \overline{L} & \overline{R} \\ \overline{F} & 1 & 3 & 9 & 9 \\ \overline{E} & 1/3 & 1 & 3 & 3 \\ \overline{L} & 1/9 & 1/3 & 1 & 1 \\ \overline{R} & 1/9 & 1/3 & 1 & 1 \end{bmatrix} \text{ and } \omega_R = \begin{bmatrix} 0.6429 \\ 0.2143 \\ 0.0714 \\ 0.0714 \end{bmatrix}. \tag{55}$$

In this case the maximum boldness index $q_{\max}$ is given by the following Table VI.

$\overline{P}_D + \overline{P}_L$	$q_{\max}$
200 [MW]	14.4774
400 [MW]	16.0433
600 [MW]	17.0373

Table VI: Maximum boldness of power dispatch problem with weight vector ω_R

and the corresponding dispatching results with $q = 1$ are given on the Table VII.

$\overline{P}_D + \overline{P}_L$	**200** [MW]			**400** [MW]			**600** [MW]		
	min $\overline{F}$	min $\overline{E}$	min μ_R	min $\overline{F}$	min $\overline{E}$	min μ_R	min $\overline{F}$	min $\overline{E}$	min μ_R
$\overline{P}_1$	36.44	33.33	35.41	82.25	66.67	79.21	100	100	100
$\overline{P}_2$	25.13	33.33	25.47	59.58	66.67	57.66	100	100	100
$\overline{P}_3$	2.52	33.33	11.96	14.26	66.67	27.29	100	100	100
$\overline{P}_4$	42.66	33.33	41.10	93.02	66.67	91.64	100	100	100
$\overline{P}_5$	32.48	33.33	31.71	55.62	66.67	55.27	100	100	100
$\overline{P}_6$	60.76	33.33	54.35	95.28	66.67	88.93	100	100	100
$\overline{F}$	1287.5	1315	1288.8	1801.30	1870	1801.3	2565.1	2565.1	2565.1
$\overline{E}$	21.31	20.34	20.89	23.29	20.72	22.41	28.05	28.05	28.05
$\overline{L}$	2.35	1.72	2.12	8.49	6.86	7.98	15.44	15.44	15.44
$\overline{R}$	0.85	0.67	0.77	3.13	2.67	2.97	6.00	6.00	6.00

Table V: Results of dispatching for different criteria with weight vector ω_R and $q = 1$

Once again, we obtain globally better results in terms of principal objectives than those of [5].

5 Conclusion

The problem of setting priority, dispatching load or resource for a group of alternatives that are characterized by multiple objectives or attributes in presence of different actors that do not have the same opinion regarding the importance of objectives has been considered in this paper. The main modeling idea of this problem rely on two procedures. First, a distinction is made among objectives in terms of objectives that behave as "larger is better" and those behaving in the sense of "smaller is better" with regard to the overall goal and actors or stakeholders are asked to do a pairwise comparison of each group of objectives, using the analytic hierarchy process approach, to obtain some relative weights. In a second stage, the obtained weights and the objectives performance interpreted as utilities are combined to define two measures: the selectability measure related to "larger is better" objectives and the rejectability measure with regard to "smaller is better" objectives in the framework of satisficing game theory. The settings, assignments or dispatching arguable to be satisficing or "good enough" are those for which the selectability measure exceeds the rejectability measure times a boldness index that is used to adjust the size of satisficing set. The final assignment or dispatching can be selected by optimizing an extra criteria subjected to satisfiability. The application of this approach to two real world problems show its potentiality ; the Matlab files used for this purpose are given at the end of the paper as annex.

References

[1] Brans J.P., Mareschal B. and Vincke Ph. (1984). PROMOTHE: A new family of outranking methods in multicriteria analysis, in Brans J.P. (Ed), Operational Research 84, pp. 477-490, North Holland.

[2] Brans J.P., Mareschal B. and Vincke Ph. (1986). How to select and how to rank projects: the PROMETHE method, European Journal of Operational Research, Vol. 24, pp. 228-238.

[3] Charnes A., Cooper W. and Ferguson R. (1955). Optimal estimation of executive compensation by linear programming, Management Science, **1**, pp. 138-151.

[4] Coello Coello C. A. (1998). Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design, PhD Thesis, Tulane University.

[5] Dhillon J.S., Parti S.C. and Kothari D. P. (1990). Multi-Objective Stochastic Optimal Thermal Power Dispatch, Proceedings of XVI Annual Convention and Exhibition of the IEEE In India, ACE'90, pp. 136 - 139.

[6] Dantzig G.B.(1963). Linear programming and extensions, Princeton University Press, Princeton, NJ.

[7] Hurson C. and Zopounidis C. (1997). Gestion de Portefeuilles et Analyse Multicritère, Economica.

[8] Ignizio J.P. (1976). Goal programming and extensions, Lexington Books, Lexington, MA.

[9] Jacquet-Lagrèze E. and Siskos J. (1982). Assesing a set of additive utility functions for multicriteria decision making: The UTA method, European Journal of Operational Research, **Vol. 10**, pp. 151-164.

[10] Lee S.M. (1972). Goal programming for decision analysis, Auerbach Publishers, Philadelphia.

[11] Luenberger D.G. (1984). Linear and Nonlinear Programming, 2nd edition, Addison Wesley.

[12] Markowitz H. M. (1952). Portfolio selection, The Journal of finance, Vol. VII, No. 1, pp. 77 - 91.

[13] Moré J.J. and Wright S.J. (1993). Optmization Software Guid, Frontiers Appl. Math. 14, SIAM, Philadelphia.

[14] Osyczka A. (1984). Muticriterion Optimization in Engineering with Fortran Programs, John Wiley & Sons, New York.

[15] Pareto V. (1896). Cours d'Economie Politique, Volume 1, Lausane: F. Rouge.

[16] Romeo C. (1991). Handbook of critical issues in goal programming, Pergamon Press, Oxford.

[17] Roy B. and Bouyssou D. (1993). Aide Multicritere a la Decision: Methodes et Cas, Edition Economica

[18] Saaty T. (1986). Axiomatic foundation of analytic hierarcy process, Management Science, Vol. 32, No. 7, pp. 841-855.

[19] Saaty T. (1980). The Analytic Hierarchical Process: Planning, Priority, Resource Allocation, McGraw Hill, New York.

[20] Saaty T. (2005). The Analytic Network Process: Decision Making with Dependence and Feedback, RWS Publications, Pittsburgh, 2005.

[21] Salminen P., Hokkanen J. and Lahdelma R. (1996). Muticriteria Decision Analysis Project on Environmental Problems, Report 5/1996, Department of Mathematics, Laboratory of Scientific Computing, University of Jyväskylä.

[22] Salo A. A. and Hämäläinen R. P. (1997). On the Measurement of Preferences in the Analytic Hierarchy Process, Journal of Multicriteria Decision Analysis, Vol. 6, pp. 309-319.

[23] Simon H. A. (1997). Administrative Behavior. A study of decision-making processes in administrative organizations, Fourth Edition, The Free Press.

[24] Siskos J. and Despotis (1989). A DSS oriented method for multi objectives linear programming problems, Decision Support Systems, Vol. 5, pp. 47-55.

[25] Steuer R. E. (1986). Muticriteria Optimization: Theory, Computation, and Application, Wiley, New York.

[26] Stirling W. C.(2003). Satisficing Games and Decision Making: With Applictions to Engineering and Computer Science, Cambridge University.

[27] Tchangani A. P. (2005). A Satisficing Game Approach for Group Evaluation of Production Units, Decision Support Systems (In Press). Published online, *DOI: 10.1016/j.dss.2005.05.010*, Elsevier.

[28] Vincke Ph. (1989). L'aide multicritere a la decision, Editions de l'Universite Libre de Bruxelles.

[29] Yu W. (1992). ELECTRE TRI: Aspects méthodologiques et manuel d'utilisation, Document LAMSADE, No. 74, Université Paris Dauphine.

[30] Zitzler E. (1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, PhD Thesis, Swiss Federal Institute of Technology Zurich.

6 Annex: Matlab files for the applications considered in the paper

The following paragraphs contain the Matlab files of the two applications problems considered in this paper that can be modified for a personal usage. To modify these files one needs to be familiar with Matlab software and its Optimization Toolbox.

6.1 Matlab Files for Portfolio Management Example

This paragraph contains the material related to the portfolio priority setting problem that can be executed using Matlab with Optimization Toolbox. There are 3 M_files and 3 Functions.

6.1.1 M_files

- **data**: this file contains all data related to portfolio problem. Any interested user can modify parameters of this file for custom use.

  CurrentRatio = [1.23 1.36 0.85 0.97 1.63 0.72 0.89 1.1 1.31 1.57 0.82 1.28 1.58 1.41 1.07 1.1 2.6 1.06 1.43 1.04 1.87 0.68 0.64 2.48 1.91 0.43 0.44 0.74 2.88 2.31 0.85] ;

  ROE = [0.147 0.098 0.141 0.118 0.23 0.241 0.163 0.212 0.202 0.137 0.171 0.177 0.216 0.186 0.181 0.177 0.164 0.114 0.299 0.064 0.104 -0.57 0.15 0.15 0.066 0.112 0.075 0.025 0.172 0.163 0.152] ;

  CFLOW = [4.91 0.74 0.19 0.69 0.26 0.64 0.52 0.88 1.72 0.58 0.88 0.31 0.32 0.24 0.19 1.01 0.51 0.34 1.66 0.71 0.31 0.96 0.23 9.41 4.87 0.82 1.36 2.99 3.67 0.62 1.31] ;

  EPS = [9538 518 600 328 10762 105 68 1312 2335 1018 639 86 217 168 2651 859 25 212 294 168 235 -88 316 371 127 176 139 125 1485 3155 687] ;

  INVPER = [1.235 -0.124 0.031 0.044 0.086 0.019 0.048 0.134 0.106 0.074 0.051 0.437 0.033 -0.045 0.017 0.09 0.076 0.052 0.052 0.055 0.061 0.419 -1.709 0.073 0.018 0.003 0.04 0.075 0.099 0.048 0.081] ;
  PER = 1./INVPER ;

  BetaMoinsUn = [0.3 0.106 0.013 0.158 0.066 0.101 0.07 0.031 0.097 0.079 0.156 5.74 0.131 0.205 0.017 0.14 0.002 0.176 0.09 0.084 0.059 0.288 0.064 0.35 0.417 0.656 0.808 0.192 0.016 0.207 0.172] ;

  Rend = [0.095 0.011 0.019 0.009 0.024 0.01 -0.002 0.013 0.012 0.018 0.003 0.11 -0.001 -0.001 0.007 0.017 0.005 -0.001 0.01 0.001 0.002 -0.015 0.011 0.005 -0.006 0.005 0.008 0.002 0.004 0.042 0.01];

  Gs = [CurrentRatio' ROE' CFLOW' EPS' Rend'] ; [ns,ms] = size(Gs) ; [n, m] =size(Gs) ;
  Gr = [BetaMoinsUn' PER'] ; [nr,mr] = size(Gr) ;
  ws = ones(1,ms)/sum(ones(1,ms)) ; wr = ones(1,mr)/sum(ones(1,mr)) ; % weights that must be defined by the user.
  q = index of boldness used ;

- **Qmax**: this file use the Matlab function *fmincon* and the user defined function *QmaxFun* to compute the maximum boldness index.

  data ;
  A = [] ; b = []; Aeq = ones(1,n) ; beq = 1; Xmin = zeros(n,1) ; Xmax = ones(n,1) ; X0 = zeros(n,1) ;
  [prio,qmin] = fmincon('QmaxFun', X0,A,b,Aeq,beq,Xmin,Xmax,[]) ; qmax = 1/qmin ;

- **dispatch**: this file computes the priority for a desired boldness index given in the data file.

  data ;
  A = [] ; b = [] ; Aeq = ones(1,n) ; beq = 1 ; Xmin = zeros(n,1) ; Xmax = ones(n,1) ; X0 = zeros(n,1);
  prio = fmincon('DispatchFun', X0,A,b,Aeq,beq,Xmin,Xmax,'SatConstr') ; [X, I] = sort(-prio) ;

6.1.2 Functions

- **QmaxFun**: this function defines the objective to minimize to obtain the maximum boldness index.

```
function [q] = QmaxFun(x)
data ; A = [] ; b = [] ; Aeq = ones(1,n) ; beq = 1 ; Xmin = zeros(n,1) ; Xmax = ones(n,1) ;
% Determination of maximum and minimum value of each objective function for normalization purpose
for j=1:ms
    x1 = linprog(-Gs(:,j), A,b, Aeq, beq, Xmin, Xmax) ; Gsmax(1,j) = Gs(:,j)'*x1 ;
    x2 = linprog(Gs(:,j), A,b, Aeq, beq, Xmin, Xmax) ;
    Gsmin(1,j) = Gs(:,j)'*x2 ;
end
for j=1:mr
    x1 = linprog(-Gr(:,j), A,b, Aeq, beq, Xmin, Xmax) ; Grmax(1,j) = Gr(:,j)'*x1 ;
    x2 = linprog(Gr(:,j), A,b, Aeq, beq, Xmin, Xmax) ;
    Grmin(1,j) = Gr(:,j)'*x2 ;
end
% Normalized objective functions
for j=1:ms
    fs(j,1) = (Gs(:,j)'*x - Gsmin(1,j))/(Gsmax(1,j)-Gsmin(1,j)) ;
end
for j=1:mr
    fr(j,1) = (Gr(:,j)'*x - Grmin(1,j))/(Grmax(1,j)-Grmin(1,j)) ;
end
% Definition of the function to be minimized
q = (wr*fr)/(ws*fs) ;
```

- **SatConstr**: this function defines the constraint of equation (23). This function is similar to **QmaxFun;** only the ouputs change.

```
function [C,Ceq] = satconstr (x)
data ; A = [] ; b = [] ; Aeq = ones(1,n) ; beq = 1 ; Xmin = zeros(n,1) ; Xmax = ones(n,1) ;
for j=1:ms
    x1 = linprog(-Gs(:,j), A,b, Aeq, beq, Xmin, Xmax) ; Gsmax(1,j) = Gs(:,j)'*x1 ;
    x2 = linprog(Gs(:,j), A,b, Aeq, beq, Xmin, Xmax) ;
    Gsmin(1,j) = Gs(:,j)'*x2 ;
end
for j=1:mr
    x1 = linprog(-Gr(:,j), A,b, Aeq, beq, Xmin, Xmax) ; Grmax(1,j) = Gr(:,j)'*x1 ;
    x2 = linprog(Gr(:,j), A,b, Aeq, beq, Xmin, Xmax) ;
    Grmin(1,j) = Gr(:,j)'*x2 ;
end
for j=1:ms
    fs(j,1) = (Gs(:,j)'*x - Gsmin(1,j))/(Gsmax(1,j)-Gsmin(1,j)) ;
```

```
end
for j=1:mr
        fr(j,1) = (Gr(:,j)'*x - Grmin(1,j))/(Grmax(1,j)-Grmin(1,j)) ;
end
C = q*wr*fr - ws*fs ; Ceq = [] ;
```

- **DispatchFun**: this function defines the objective the user wants to optimize, the user can modify this function (mainly the last line) to set up its own objective. By default this function does not optimize a criterion (F = 0). Remove comments % and change the last line to define a particular criterion function.

```
function [F] =DispatchFun(x)
% data ; A = [] ; b = [] ; Aeq = ones(1,n) ; beq = 1 ; Xmin = zeros(n,1) ; Xmax = ones(n,1) ;
%for j=1:ms
        % x1 = linprog(-Gs(:,j), A,b, Aeq, beq, Xmin, Xmax) ; Gsmax(1,j) = Gs(:,j)'*x1 ;
        x2 = linprog(Gs(:,j), A,b, Aeq, beq, Xmin, Xmax) ;
        %Gsmin(1,j) = Gs(:,j)'*x2 ;
%end
% for j=1:mr
        % x1 = linprog(-Gr(:,j), A,b, Aeq, beq, Xmin, Xmax) ; Grmax(1,j) = Gr(:,j)'*x1 ;
        x2 = linprog(Gr(:,j), A,b, Aeq, beq, Xmin, Xmax) ;
        %Grmin(1,j) = Gr(:,j)'*x2 ;
%end
%for j=1:ms
        % fs(j,1) = (Gs(:,j)'*x - Gsmin(1,j))/(Gsmax(1,j)-Gsmin(1,j)) ;
%end
%for j=1:mr
        % fr(j,1) = (Gr(:,j)'*x - Grmin(1,j))/(Grmax(1,j)-Grmin(1,j)) ;
%end
F = 0 ; % q*wr*fr - ws*fs ; wr*fr ; -ws*fs ; .....
```

6.2 Matlab Files for Power Dispatching Example

This paragraph contains the material related to the power dispatching problem that can be executed using Matlab with Optimization Toolbox. There are 3 M_files and 3 Functions.

6.2.1 M_files

- **data**: this file contains all data related to power dispatch problem. Any interested user can modify parameters of this file for custom use.

```
CV = 0.01 ; % Coefficient of variation
A1 = [0.005 0 0 0 0 0 ; 0 0.010 0 0 0 0 ; 0 0 0.020 0 0 0 ; 0 0 0 0.003 0 0 ; 0 0 0 0 0.015 0 ; 0 0 0 0 0 0.010] ;
n = length(A1) ; A = A1 + CV^2*A1 ;
b = [2.0 2.0 2.0 1.95 1.45 0.95]' ; c = [100 200 300 80 100 120]' ;
D1 = 10^(-3)*[0.6572 0 0 0 0 0 ; 0 0.5916 0 0 0 0 ; 0 0 0.4906 0 0 0 ; 0 0 0 0.378 0 0 ; 0 0 0 0 0.4906 0 ; 0 0 0 0 0 0.5173] ;
```

*D = D1 + CV^2*D1 ;*

e = [-0.05497 -0.05880 -0.05014 -0.03150 -0.05014 -0.05548]' ;

f = [4.111 2.593 4.268 5.526 4.268 6.132]' ;

B1=10^(-2)[0.0200 0.0010 0.0015 0.0005 0.0000 0.0030 ; 0.0010 0.0300 -0.0020 0.0001 0.0012 0.0010 ; 0.0015 -0.0020 0.0100 0.0010 0.0010 0.0008 ; 0.0005 0.0001 0.0010 0.0150 0.0006 0.0050 ; 0.0000 0.0012 0.0010 0.0006 0.0250 0.0020 ; 0.0030 0.0010 0.0008 0.0050 0.0020 0.0210] ;*

```
for i = 1:length(B1)
    for j = 1:length(B1)
        if i==j
            B(i,j) = (1+CV^2)*B1(i,i) ;
        else
            B(i,j) = B1(i,j) ;
        end
    end
end
```

*R = diag(ones(1,n))*CV^2 ; PD = 200 ; % power demand to be supplied by the user*

PL = 0 ; % average loss to be supplied by the user

lambda = -log(0.8)/600 ; % parameter defined in equation (48) of the corresponding paper to be supplied by the user

Relmax = exp(-lambda(PD+PL)) ; Relmin = 0 ; q = 1 ; % boldness index to be supplied by the user*

OmegaS = 1 ; % weights to be supplied by the user

OmegaR = ones(4,1)/sum(ones(4,1)) ; % [0.6429 0.2143 0.0714 0.0714]', weights to be supplied by the user

- **Qmax**: this file use the Matlab function *fmincon* and the user defined function *QmaxFun* to compute the maximum boldness index.

  data ;

  *[P, qmin] = fmincon('QmaxFun', zeros(n, 1), [], [], ones(1,n), PD+PL, zeros(n,1), Inf*ones(n,1), []) ;*
  qmax = 1/qmin ;

- **dispatch**: this file computes the dispatching and its related cost when one supply the desired boldness index in the data file.

  data ;

  *P = fmincon('dispatchfun', zeros(n, 1), [], [], ones(1,n), PD+PL, zeros(n,1), Inf*ones(n,1), 'dispatchconstr') ;*

  *F = P'*A*P + b'*P + ones(1,n)*c ; E = P'*D*P + e'*P + ones(1,n)*f ; L = P'*B*P ; R = P'*R*P ;*

  Constr = dispatchconstr(P) ;

6.2.2 Functions

- **QmaxFun**: this function defines the objective to minimize to obtain the maximum boldness index.

  function [q] = QmaxFun(x)

  data ;

  % Determination of maximum and minimum value of each objective function for normalization purpose

  *[PFmax,Fmax] = quadprog(-2*A, -b, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

Fmax = -Fmax ;

*[PFmin,Fmin] = quadprog(2*A, b, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

*[PEmax,Emax] = quadprog(-2*D, -e, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

Emax = -Emax ;

*[PEmin,Emin] = quadprog(2*D, e, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

*[Plmax,Lmax] = quadprog(-B, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

Lmax = -Lmax ;

*[Plmin,Lmin] = quadprog(B, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

*[PRmax,Rmax] = quadprog(-R, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

Rmax = -Rmax ;

*[PRmin,Rmin] = quadprog(R, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

Relmax = exp(-lambda(PD+PL)) ; Relmin = 0 ;*

% Normalized objective functions

*gS = (exp(-lambda*ones(1,n)*x)-Relmin)/(Relmax-Relmin) ; % related to selectability measure*

f = OmegaR(1,1)((x'*A*x+b'*x - Fmin)/(Fmax - Fmin)) ;*

e = OmegaR(2,1)((x'*D*x+e'*x - Emin)/(Emax - Emin)) ;*

l = OmegaR(3,1)((x'*B*x - Lmin)/(Lmax - Lmin)) ;*

r = OmegaR(4,1)((x'*R*x - Rmin)/(Rmax - Rmin)) ;*

gR = f + e + l + r ; % related to selectability measure

% Definition of the function to be minimized

q = gR/gS ; % minimizing gR/gS is equivalent to maximizing gS/gR

- **dispatchconstr**: this function defines the nonlinear constraint of equation (23) of the above paper. This function is similar to **QmaxFun ;** only the ouputs change.

  function [C,Ceq] = dispatchconstr(x)

  data ;

  *[PFmax,Fmax] = quadprog(-2*A, -b, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  Fmax = -Fmax ;

  *[PFmin,Fmin] = quadprog(2*A, b, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  *[PEmax,Emax] = quadprog(-2*D, -e, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  Emax = -Emax ;

  *[PEmin,Emin] = quadprog(2*D, e, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  *[Plmax,Lmax] = quadprog(-B, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  Lmax = -Lmax ;

  *[Plmin,Lmin] = quadprog(B, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  *[PRmax,Rmax] = quadprog(-R, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  Rmax = -Rmax ;

  *[PRmin,Rmin] = quadprog(R, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;*

  Relmax = exp(-lambda(PD+PL)) ; Relmin = 0 ;*

  *gS = (exp(-lambda*ones(1,n)*x)-Relmin)/(Relmax-Relmin) ;*

  f = OmegaR(1,1)((x'*A*x+b'*x - Fmin)/(Fmax - Fmin)) ;*

```
e = OmegaR(2,1)*((x'*D*x+e'*x - Emin)/(Emax - Emin)) ;
l = OmegaR(3,1)*((x'*B*x - Lmin)/(Lmax - Lmin)) ;
r = OmegaR(4,1)*((x'*R*x - Rmin)/(Rmax - Rmin)) ;
gR = f + e + l + r ;
C = q*gR-gS ; Ceq = [] ;
```

- **dispatchfun**: this function defines the objective the user wants to optimize, the user can modify this function to set up. This function is similar to **QmaxFun ;** only the ouputs change.

```
function [F] = dispatchconstr(x)
data ;
[PFmax,Fmax] = quadprog(-2*A, -b, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
Fmax = -Fmax ;
[PFmin,Fmin] = quadprog(2*A, b, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
[PEmax,Emax] = quadprog(-2*D, -e, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
Emax = -Emax ;
[PEmin,Emin] = quadprog(2*D, e, [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
[Plmax,Lmax] = quadprog(-B, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
Lmax = -Lmax ;
[Plmin,Lmin] = quadprog(B, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
[PRmax,Rmax] = quadprog(-R, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
Rmax = -Rmax ;
[PRmin,Rmin] = quadprog(R, zeros(n,1), [],[],ones(1,n),PD+PL,zeros(n,1),Inf*ones(n,1)) ;
Relmax = exp(-lambda*(PD+PL)) ; Relmin = 0 ;
gS = (exp(-lambda*ones(1,n)*x)-Relmin)/(Relmax-Relmin) ;
f = OmegaR(1,1)*((x'*A*x+b'*x - Fmin)/(Fmax - Fmin)) ;
e = OmegaR(2,1)*((x'*D*x+e'*x - Emin)/(Emax - Emin)) ;
l = OmegaR(3,1)*((x'*B*x - Lmin)/(Lmax - Lmin)) ;
r = OmegaR(4,1)*((x'*R*x - Rmin)/(Rmax - Rmin)) ;
gR = f + e + l + r ;
F = f ; %gR ; % -gS ; %1/gS ; %q*gR-gS ; f ; e ; l ; r
```