Why is difficult to make decisions under multiple criteria?

F. Della Croce
Dipartimento Automatica Informatica
Politecnico di Torino
dellacroce@polito.it

A. Tsoukiàs,
LAMSADE - CNRS
Université Paris Dauphine
tsoukias@lamsade.dauphine.fr

P. Moraïtis
Dept. of Computer Science
University of Cyprus
moraitis@ucy.ac.cy

Abstract

The paper makes a survey of the principal difficulties the multiple criteria decision making introduces with a particular emphasis on scheduling problems. Two types of difficulties are considered. The first is of conceptual nature and has to do with the difficulty of defining the concept of optimality in presence of multiple criteria and the impossibility to define universal preference aggregation procedures. The second difficulty is of more technical nature and concerns the increasing computational complexity of multiple criteria decision making problems. A number of examples are introduced in order to explain these issues.

Introduction

In this paper, decision making is referred to an agent (artificial or human) who has to act within a given context, with a given amount of resources and time in order to pursue one or more goals. The decision process is expected to be characterised by a form of rationality (possibly bounded) and to be represented in a formal way (the agent has preferences expressed either under a value function or more simply as a binary relation on the set of consequences of his/her actions). This is the frame of operational research and/or decision theory, possibly under Simon's (Simon 1979) bounded rationality variant.

In real life, making decisions under multiple criteria is the standard situation: there are always different consequences to consider, there always more objectives and goals to satisfy, there are always more opinions to take in account. Under this point of view the presence of multiple criteria it should be considered the general case, while single criterion optimisation should be considered as a special case. This is not what happened in the history of OR, where the first contributions on the use of multiple criteria appeared in the late 60s, early 70s (Roy 1968; Geoffrion 1968; Zeleny 1974; Keeney & Raiffa 1976).

The difficulty to make decisions under multiple criteria is twofold. The principal difficulty is conceptual. OR and decision theory are based on the idea of a rational decision process represented by a single objective function to optimise. Such an idea simply does not apply in the presence of multiple criteria. Further on, some other conceptual difficulties arise. Is it possible to substitute optimality with another concept? Are there universal procedures solving multiple criteria decision making problems? We explore these issues in section 2. The second difficulty is more technical and has to do with complexity. We confine ourselves in scheduling problems in order to show that the presence of multiple criteria normally implies the increase of computational complexity of the problem also in apparently "easy" problems. We discuss this problem in section 3.

The paper is based on results which are well known in literature. The aim of the paper is to put together such results for a community such as the A.I. planning and scheduling one. Further on, we want to show the importance of an autonomous theory concerning decision making and support in presence of multiple criteria and the difficulties such an effort has to face.

The vanishing optimum

Can the concept of optimum vanish (Schärlig 1996)? Traditionally when we think about decision theory we think about optimisation: find the one best solution. From a strict mathematical point of view this is straightforward. Express your problem as a function F of your decision variables $x_1, \cdots, x_n$ and then find the minimum (or maximum) of the function. This is well defined since

$$\min(F(x_1 \cdots, x_n)) \Leftrightarrow F'(x_1 \cdots, x_n) = 0$$

where F' is the "derivate" of function F. But then, as soon as we consider more than one criteria (more objective functions) we have a set of functions $F_i, i = 1 \cdots, m$ and we should look for a solution X such that $\forall i\ F'_i(X) = 0$ and this is a problem since $\forall i\ F'_i(X) = 0$ can be an inconsistent sentence.

Example 0.1 *Consider two objective function F_1, F_2, both to minimise, such that $\min(F_1) = A = \max(F_2)$ and $\min(F_2) = B = \max(F_1)$. Clearly the sentence $\forall i\ F'_i(X) = 0$ is inconsistent.*

There is no way to guarantee that in presence of multiple criteria there exist feasible solutions such that all objective functions can be simultaneously optimised. What we learn from that?

Difficulty 0.1 *Unlike traditional optimisation, the presence of multiple criteria does not allow to establish an "objective" definition of "optimal solution".*

In other terms when we work using multiple criteria there is no mathematical definition of the solution. We have to introduce alternative concepts, less easy to define and moreover subjectively established. What are we allowed to establish in the frame of multiple criteria?

There is a set of feasible solutions which are the "natural" candidates for solving a multiple criteria decision making problem. These are the so-called Pareto solutions (or efficient solutions or non dominated solutions). We introduce the following notation:

$$\forall X, Y \; D(X, Y) \Leftrightarrow \forall i \; F_i(X) \leq F_i(Y) \wedge \exists k \; F_k(X) < F_k(Y)$$

We read: *solution X dominates solution Y, iff for all criteria X is at least as good as Y and there is at least one criterion where X is strictly better than Y.* It is clear that all feasible solutions which are **not** dominated are potentially solutions of our problem (a dominated solution is obviously not interesting). The problem is that the set of Pareto solutions can be extremely large (sometimes equal to the set of feasible solutions).

Example 0.2 *Consider three candidates A, B, C such that for criterion: 1 $A > B > C$, for criterion 2: $B > C > A$ and for criterion 3: $C > A > B$ ($>$ representing a preference). All three candidates are non dominated.*

What can we do? Roughly there are two ways to face the problem:
1. fix a function $\mathcal{F}(F_1, \cdots, F_m)$ and then try to optimise $\mathcal{F}$ (that is re-conduct the problem to a single criterion optimisation problem);
2. explore the feasible or the efficient set using a majority rule as this is conceivable in various voting procedures (that is, choose the Pareto solution preferred by the "majority" of criteria).

One single function

The basic idea is simple. Put together the different functions in such a way that we obtain one single value for each feasible solution. After all this is exactly what happens in all schools, university degrees, multi-dimensional indices, cost benefit analysis and hundred other examples of "more or less" simple aggregation functions where values expressed on different attributes are merged in one single value.

The interested reader can look in (Bouyssou *et al.* 2000) for a nice presentation of all the drawbacks and unexpected consequences of such an approach. We try to summarise.

- Such a global function does not always exist. To say it in other terms, the conditions under which such a function exist are not always possible to fulfill. First of all evaluation on the different objective functions have to commensurable. Provided it is the case, then it should be possible to compensate the values of one function with the values of another function. If this is possible then each subset of functions should be preferentially independent with respect to its complement (see (Keeney & Raiffa 1976) for a detailed presentation of this approach). Last, but not least, it is possible that the effort to adapt the information to these conditions results in a model which has nothing to do with the original problem.
- Fulfilling the conditions can be possible in principle, but impossible in practice. In the sense that the cost of obtaining the extra information (such as the trade-offs among the criteria, the trial-error protocol used in order to calibrate the global function etc.) can be simply to large with respect to the problem or even unattainable (see (Hobbs 1986; Svenson 1996; Mongin 2000) for a discussion on this issue, including the cognitive effort required for such an approach).
- In any case, even if such a function can be defined, further information is required in order to establish it. Such information concerns two non exclusive issues:
  - further preferential information (trade offs among criteria, ideal points in the criteria space etc.);
  - shape of the global function (additive, distance, non linear etc.).

  In human decision support usually is the client (or decision maker) who provides such information through a protocol of information exchange with the analyst. However, there is always some arbitrariness in this process since this information depends also on technical choices (for instance trade offs are necessary in an additive function, but not in the frame of scalarising constants; the reader can see (Steuer 1986; Vanderpooten 1989; Korhonen, Moskowitz, & Wallenius 1992) for more details).

  The problem is more difficult in the case of "automatic" decision support as with artificial agents. Either such an agent has to carry enough preferential information or it has to be able to support a dialog with a human providing such information. Moreover the agent should be aware of the technical knowledge necessary to define the global function. It is always possible to fix the global function (at least the shape) from the beginning, but then we impose a severe limitation to the agent's autonomy.

Let the criteria vote

Another option is to make the criteria vote as if they were parties in a parliament. The idea is simple. Given any two feasible solutions X and Y, X is better than Y if it is the case for the majority of criteria. Hundreds of parliaments, committees, boards, assemblies, use this principle of democracy.

The interested reader can again refer to (Bouyssou *et al.* 2000) for a critical presentation of the drawbacks and counterintuitive results such an approach presents. Again we summarise.

- There is no universal voting procedure. Since the 18th century we know that voting procedures are either manipulable (to some extend a minority can impose its will) or potentially ineffective (unable to find a solution) as can be seen in the following example (borrowed from (French 1988)).

Example 0.3 *Consider four candidates (A,B,C,D) and seven examiners (a,b,c,d,e,f,g). Each examiner gives a preference in decreasing order (1 is the best, 2 is the second best etc.). The following table is provided.*

	a	*b*	*c*	*d*	*e*	*f*	*g*
A	*1*	*2*	*4*	*1*	*2*	*4*	*1*
B	*2*	*3*	*1*	*2*	*3*	*1*	*2*
C	*3*	*1*	*3*	*3*	*1*	*2*	*3*
D	*4*	*4*	*2*	*4*	*4*	*3*	*4*

If we sum the ranks of each candidate we obtain $\sigma(A) = 15$, $\sigma(B) = 14$, $\sigma(C) = 16$, $\sigma(D) = 25$ *and clearly B is the winner. Suppose now that for some reason the candidate D could not participate to the selection. Being the worst one should expect that nothing changes. Unfortunately it is not the case. Recomputing the sum of the ranks we obtain* $\sigma'(A) = 13$, $\sigma'(B) = 14$, $\sigma'(c) = 15$ *and now A is the winner. This is tricky. On the other hand if we look on pure majorities we get that* $A > B$ *(five examiners prefer A to B),* $B > C$ *(five examiners prefer B to C) and* $C > A$ *(four examiners prefer C to A). There is no solution.*

Arrow (Arrow 1963) definitely solved the problem proving the following theorem.

Theorem 0.1 *When the number of candidates is at least 3, there exists no aggregation method satisfying simultaneously the properties of universal domain, unanimity, independence and non-dictatorship.*

where:
- universal domain means that there is no restriction on the preferences to aggregate;
- unanimity means that an aggregation procedure should not violate the unanimity;
- independence means that in order to establish if X is better than Y we consider only information concerning X and Y and nothing else;
- non-dictatorship means that there is no preference information which is more importante than others, such to impose its will.

The reader can see that although the conditions imposed by Arrow are very "natural" they are inconsistent. In other terms: there is no universal preference aggregation procedure. Either we choose for guaranteing a result and we take the risk of favouring a minority or we impose the majority rule and we take the risk not to be able to decide. Decision efficiency and democracy are incompatible.

- Suppose a voting procedure has been chosen. If it is manipulable then one should obtain the information necessary to control possible counterintuitive results. If it is a majority rule then the outcome could be an intransitive and/or incomplete binary relation. In such a case further manipulation is necessary in order to obtain a final result.

  As for the previous approach such further information is usually provided by the client (the decision maker) through a precise dialog. A number of guidelines apply here (see (Bouyssou *et al.* 2000)), but no structured methodological knowledge is available up today. In the case of automatic decision making things become much more difficult since an artificial agent should be able to understand the difference among several voting schemes and procedures.

  What did we learn from the above discussion?

Difficulty 0.2 *There is no way to establish an universal procedure for a multiple criteria decision making problem. Either further information has to be gathered or "extra-problem" procedures have to be adopted. Either the quality of the outcome can be poor (but we are sure to have an outcome) or we require a nice outcome knowing that it might be impossible to obtain it.*

The fact that we have such "negative" results should not induce the reader to consider that multiple criteria decision making problems are just a mess. In real world decision makers make every day sound decisions using multiple criteria. What we have to give up is the idea of **THE** solution of a multiple criteria decision making problem. We need to accept locally, bounded to the available information and resources, satisfying solutions.

There is still one more open question. Suppose that for a given problem we establish a model (and a concept of good or optimal solution). Suppose also that a precise procedure has been adopted in order to put together the preferences on the different criteria. How "complicated" is to reach a solution?

Complexity issues

Let us assume that a well defined multiple criteria optimisation model is available and, without loss of generality, let us consider scheduling problems. For a comprehensive analysis on multiple criteria scheduling we refer to (T'kindt & Billaut 2002). We will deal with the simplest scheduling environment, namely the static single machine environment. We use the notation given in (Chen & Bulfin 1993) that extends to multiple objective problems the so-called *three-field* $\alpha/\beta/\gamma$ classification of Lawler (Lawler *et al.* 1993).

Consider a set N of n jobs where each job j has a processing time p_j, a weight w_j and a due date d_j, respectively. Given a schedule, for each job j we denote with C_j its completion time, with $T_j = \max\{C_j - d_j, 0\}$ its tardiness. Also, let $T_{\max}$ denote the maximum tardiness of the schedule. Finally, let U_j denote the unit penalty for job j being tardy: namely, $U_j = 1$ if $T_j > 0$, else $U_j = 0$.

If we refer to mono-criterion problems, we already encounter all main classes of computational complexity (see (Garey & Johnson 1979) for details): for instance, the $1||\sum w_j C_j$, the $1||\sum U_j$ and the $1||T_{\max}$ are polynomially solvable, whereas the $1||\sum T_j$ is weakly NP-hard and the $1||\sum w_j U_j$ and the $1||\sum w_j T_j$ are strongly NP-hard.

Consider the simplest multiple criteria environment, namely the bi-criteria one and the two main general approaches indicated previously for putting together the two criteria (a specific case of the first approach is considered for presentation purposes):
(1) fix a function weighting the two criteria by means of

a lexicographic rule (one criterion is designated as primary and the other criterion is designated as secondary);
(2) generate the set of efficient solutions (to be then explored by some majority rule). Notice that an optimal solution of (1) always belongs to the set of efficient solutions described by (2).

In the three-field scheduling notation, γ denotes the performance measure. Let γ_1 and γ_2 be the two performance measures for the bi-criterion problem. Consider, now, the above general approaches with respect to single machine bi-criteria problems. In case (1), the objective γ_1 is lexicographically more important than objective γ_2 and the corresponding problem will be denoted as $1||(\gamma_2|\gamma_1)$. In case (2), where the set of non dominated solutions must be determined the corresponding problem will be denoted as $1||\gamma_1, \gamma_2$.

The following result proposed in (Chen & Bulfin 1993) links the complexity of a problem with single objective γ_1 to the complexity of bi-criteria problems involving objective γ_1.

Theorem 0.2 *If $1||\gamma_1$ is NP-hard, then $1||(\gamma_2|\gamma_1)$ and $1||\gamma_1, \gamma_2$ are NP-hard.*

Theorem 0.2 indicates that there is little hope to efficiently handle multiple criteria problems if any of the related mono-criterion problems is difficult.

There are actually a few special cases where the bi-criterion lexicographic problem is polynomially solvable when the secondary objective induces a mono-criterion NP-hard problem.

An example of this peculiar situation is given by the $1||(\sum T_j|\sum C_j)$ problem. The $1||\sum T_j$ problem is known to be NP-hard in the ordinary sense, whilst the $1||\sum C_j$ is known to be optimally solved in polynomial time by sequencing the jobs in nondecreasing order of their processing times, the so-called SPT rule. In the $1||(\sum T_j|\sum C_j)$ problem, in order to optimise the primary objective, the SPT rule must be respected. However there may be ties, namely jobs with identical processing times. Only for these jobs it is possible to optimise the secondary criterion. But this is equivalent to solve a special case of the $1||\sum T_j$ problem with all identical processing times, this latter problem being optimally solvable in polynomial time by sequencing the jobs in nondecreasing order of the due dates (the well known EDD rule). Hence, the $1||(\sum T_j|\sum C_j)$ problem is polynomially solvable.

Analogously there are a few special cases where the bi-criterion lexicographic problem is pseudo-polynomially solvable when the secondary objective induces a mono-criterion strongly NP-hard problem. An example is $1||(\sum w_j T_j|\sum w_j C_j)$ problem which is NP-hard in the ordinary sense though the $1||(\sum w_j T_j)$ problem is NP-hard in the strong sense. These are the only *relative good news* we have.

The following theorem also proposed in (Chen & Bulfin 1993) links the complexity of cases (1) and (2).

Theorem 0.3 *If $1||(\gamma_2|\gamma_1)$ is NP-hard, then $1||\gamma_1, \gamma_2$ is NP-hard.*

Theorem 0.3 indicates that case (2) is at least as difficult as case (1). Let then focus on bi-criteria problems handled by means of a lexicographic approach. We have here pretty bad results as bi-criteria problems involving polynomially solvable mono-criterion ones are often already NP-hard.

For instance, consider the $1||(\sum w_j C_j|T_{\max})$ problem. Both the $1||\sum w_j C_j$ problem and the $1||T_{\max}$ problem are polynomially solvable. The $1||(\sum w_j C_j|T_{\max}$ problem, however, is NP-hard in the strong sense as shown in (Hoogeveen 1992). This is due to the fact that the primary objective $T_{\max}$ induces a constraint in the secondary objective of the type $T_j \leq T_{\max}\ \forall j$, that can be written as $C_j \leq d_j + T_{\max}\ \forall j$. By introducing a deadline $\overline{d_j} = d_j + T_{\max}$, we obtain $C_j \leq \overline{d_j}\ \forall j$. Hence, the above $1||(\sum w_j C_j|T_{\max})$ problem is equivalent to the $1|\overline{d_j}|\sum w_j C_j$ problem which is known to be NP-hard in the strong sense.

What happens is that the lexicographic weighting of criteria (that we have seen to be generally easier than the generation of the efficient solutions) induces a further constraint (well defined as the primary objective is polynomially solvable) in the solutions space: this nearly always induces untractable bi-criteria problems that are polynomially solvable when only the secondary criterion is considered. This is what occurs in terms of pure computational complexity.

Also in practice, however, the structural properties of the problem defined on the secondary criterion tend to be destroyed when the primary objective is introduced as constraint.

An example of this is given by the $1||\sum(T_j|T_{\max})$ problem. By the same approach applied previously, this problem can be shown to be equivalent to the $1|\overline{d_j}|\sum T_j$ problem. But the presence of the deadlines kills the nice decomposition structure (leading to a pseudo-polynomial dynamic programming algorithm) of the $1||\sum T_j$ problem as shown in (R. Tadei). At the present state of the art the $1||\sum T_j|T_{\max})$ problem is open with respect to the weakly or strongly NP-hardness status.

What did we learn then in terms of complexity?

Difficulty 0.3 *Even when we deal with the easiest well defined multiple criteria problems, we immediately fall into NP-hard problems. There is very little hope to derive polynomial algorithms for multiple criteria problems whatever is the complexity status of the corresponding mono-criterion problems.*

So, also in terms of computational complexity, we face pretty negative results. Rather than being discouraged by this situation (as for NP-hard mono-criterion problems several high quality meta-heuristics exist for multiple objective problems), we need to precise very carefully the goals of our decision making: for instance, there is nonsense in searching for the complete set of efficient solutions if such set has huge cardinality.

As an example, consider problem $1||\sum w_j C_j, \sum h_j C_j$ where each job j has two weights (w_j and h_j). It is possible to derive the set of all efficient solutions by means of an ϵ-constraint approach and each solution can be computed in polynomial time. However the $1||\sum w_j C_j, \sum h_j C_j$ problem is NP-hard in the ordinary sense as the number of efficient solutions may not be polynomially bounded as shown in (Hoogeveen 1992).

Conclusions

In this paper we analyse the conceptual and technical difficulties associated to decision making problems in presence of multiple criteria. Three difficulties are discussed:
- the impossibility to introduce an “objective” definition of solution;
- the impossibility to define “universal” preference aggregation procedures;
- the increasing computational complexity even when each single criterion corresponds to an “easy” problem.

Despite the apparent negative nature of the above results we claim that the development of precise preference aggregation procedures, of heuristics adapted to the presence of multiple criteria, allow for a given decision making problem to find satisfying solutions. What we should keep in mind is that:
- it makes no sense to look behind “optimality”, in any way it might be defined;
- the method which is going to be used in order to solve a multiple criteria decision making problem is part of the model of the problem and is not defined externally.

References

Arrow, K.J. 1963. *Social choice and individual values*. Wiley, New York, 2nd edition.

Bouyssou, D.; Marchant, Th.; Pirlot, M.; Perny, P.; Tsoukiàs, A.; and Vincke, Ph. 2000. *Evaluation and decision models: a critical perspective*. Kluwer Academic, Dordrecht.

Chen, C.H., and Bulfin, R.L. 1993. Complexity of single machine, multi-criteria scheduling problems. *European Journal of Operational Research* 70:115–125.

French, S. 1988. *Decision Theory*. Ellis Horwood, Chichester.

Garey, M., and Johnson, D. 1979. *Computers and Intractability*. Freeman and Company, New York.

Geoffrion, A. 1968. Proper efficiency and the theory of vector optimisation. *Journal of Mathematical Analysis and Application* 22:618–630.

Hobbs, F. 1986. What can we learn from experiments in multiobjective decision analysis? *IEEE Transactions on Systems Man and Cybernetics* 16:384–394.

Hoogeveen, J.A. 1992. *Single machine bicriteria scheduling*. Ph.D. Dissertation, CWI, Amsterdam.

Keeney, R.L., and Raiffa, H. 1976. *Decisions with multiple objectives: Preferences and value tradeoffs*. Wiley, New York.

Korhonen, P.; Moskowitz, H.; and Wallenius, J. 1992. Multiple criteria decision support - a review. *European Journal of Operational Research* 63:361–375.

Lawler, E.L.; Lenstra, J.K.; Kan, A.H.G. R.; and Shmoys, D.B. 1993. Sequencing and scheduling: Algorithms and complexity. In Graves, S.C.; Kan, A.H.G. R.; and Zipkin, P., eds., *Handbooks in Operations Research and Management Science: Logistics of Production and inventory*. North-Holland, Amsterdam.

Mongin, Ph. 2000. Does optimisation implies rationality? *Synthese* 124:73–111.

R. Tadei, A. Grosso, F. D. C. Finding the pareto optima for the total and maximum tardiness single machine scheduling problem. *Discrete Applied mathematics* forthcoming.

Roy, B. 1968. Classement et choix en présence de points de vue multiples: Le méthode electre. *Revue Francaise d'Informatique et de Recherche Opérationnelle* 8:57–75.

Schärlig, A. 1996. The case of the vanishing optimum. *Journal of Multicriteria Decision Analysis* 5:160–164.

Simon, H.A. 1979. Rational decision making in business organizations. *American Economic Review* 69:493–513.

Steuer, R.E. 1986. *Multiple criteria optimisation: Theory, computation, and application*. Wiley, New York.

Svenson, O. 1996. Decision making and the search for fundamental psychological regularities: what can we learn from a process perspective? *Organisational Behaviour and Human Decision Processes* 65:252–267.

T'kindt, V., and Billaut, J.C. 2002. *Multicriteria Scheduling*. Springer Verlag, Berlin.

Vanderpooten, D. 1989. The interactive approach in mcda: a technical framework and some basic conceptions. *Mathematical and Computer Modelling* 12:1213–1220.

Zeleny, M. 1974. *Lineat Multiobjective programming*. LNEMS 95, Springer Verlag, Berlin.