Multi-Objective Optimization

Carlos A. Coello Coello

ccoello@cs.cinvestav.mx
CINVESTAV-IPN
Evolutionary Computation Group (EVOCINV)
Computer Science Department
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D.F. 07360, MEXICO

Patna, India, December, 2016

LECTURE 7

Introduction

An aspect to which relatively little attention has been paid in EMOO is the incorporation of user's preferences.

Many researchers working with MOEAs disregard the fact that the solution of a multi-objective optimization problem actually involves three stages: measurement, search and decision making.

Finding PF_{true} does not completely solve a multi-objective problem since, in practice, only one or a few nondominated solutions can be implemented, so it is required to allow the user (or decision maker) to select such (preferred) solutions.

Also, in many real-world applications, the user wants to bias the search towards certain specific regions of the Pareto front.

Introduction

From the Operations Research (OR) perspective, there are two main lines of thought regarding MCDM [Huylenbroeck, 1995]:

1. The French school, which is mainly based on the **outranking** concept [Vincke, 1986], and
2. The American **Multi Attribute Utility Theory** (MAUT) school [Keeney, 1993].

Outranking

The French school is based on an outranking relation which is built up under the form of pairwise comparisons of the objects under study.

The main goal is to determine, on the basis of all relevant information for each pair of objects, if there exists preference, indifference, or incomparability between the two. For this purpose, preference or dominance indicators are defined and compared with certain threshold values.

The main disadvantage of this approach is that it can become very expensive (computationally speaking) when there is a large number of alternatives.

Also, some authors consider the use of outranking methods as complementary to other techniques (e.g., MAUT) and are therefore intended for problems that present certain characteristics (e.g., at least one criterion is not quantitative) [Roy, 1996].

Multi-Attribute Utility Theory

MAUT is based, in contrast, on the formulation of an overall utility function, and its underlying assumption is that such a utility function is available or can be obtained through an interactive process.

When this utility function is not available, the task is then to identify a set of nondominated solutions. In this case, strong preference can only be concluded if there exists enough evidence that one of the vectors is clearly dominating the vector against which it is compared.

Weak preference (modeled as weak dominance), on the other hand, expresses a certain lack of conviction. **Indifference** means that both vectors are "equivalent" and that it does not matter which of them is selected.

It is important to distinguish this "indifference" from the "incomparability" used with outranking methods, since the second indicates vectors with strong opposite merits [Huylenbroeck, 1995].

Multi-Attribute Utility Theory

MAUT does not work when there are intransitivities in the preferences, which is something that frequently arises when dealing with "incomparable" objects using an outranking approach [Zeleny, 1977].

There are a few issues related to MAUT deserving some discussion. First, it is important to distinguish between global and local approaches to MAUT. Despite the fact that it is common practice to assume a global approach to MAUT in which an overall utility function that expresses the DM's global preferences is assumed, operations researchers tend to favor local approaches. In local approaches, the utility function is decomposed into simple utility functions (e.g., single attribute functions) that are easier to handle [Geoffrion, 1972].

Second, it is important to mention that a utility function does not really reflect the DM's inner (psychological) intensity of preference. It just provides a model of the DM's behavior [Mossin, 1973]. This is an important distinction, since behavior should then be consistent (i.e., it should not originate intransitivities), according to MAUT's practitioners.

Multi-Attribute Utility Theory

The main criticism towards MAUT is its inability to handle intransitivities. There are, however, reasons for not dealing with intransitivities in MAUT [Luce, 1957]:

- MAUT is only concerned with behavior of the DM, and behavior is transitive.
- The transitive is often a “close” approximation to reality.
- MAUT’s interest is limited to “normative” or “idealized” behavior.
- Transitive relations are far more mathematically tractable than intransitive ones

From these arguments, it is often assumed that the main reason to support MAUT is in fact the mathematical tractability of utility functions [Starr, 1977].

Operational Attitude of the Decision Maker

The French and American schools of thought lead to three types of operational attitude of the DM [Roy, 1977]:

1. Exclude incomparability and completely express preferences by a unique criterion. This leads to an aggregating approach in which all the criteria are combined using a single utility function representing the DM's global preferences. An example of this approach is the technique called "maximim programming" [Dyson, 1980].
2. Accept incomparability and to use an outranking relation to model the DM's preferences. In this case, the DM only has to model those preferences that is capable of establishing objectively and reliably, using outranking only when such preferences cannot be established. In this case, the DM is asked to compare all criteria two by two; each objective is assigned a weight derived from the eigenvector of the pairwise comparison matrix [Huylenbroeck, 1995]. It is important to be aware that these pairwise comparisons can lead to intransitive or incomplete relations. One example of this approach would be ELECTRE in its different versions [Roy, 1978].

Operational Attitude of the Decision Maker

3. Determine, through an interactive process, the different possible compromises based on local preferences. In this case the DM experiments with local preferences at each stage of the search process, which allows exploration of only a certain region of the search space.

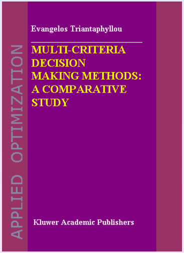
   These local preferences can be expressed in different ways (e.g., a ranking of objectives, an adjustment of aspiration levels, or even detailed trade-off information) [Evans, 1984].

   The main issue here is that the DM is not asked preliminary (specific) preference information. Such preferences are derived from the behavior exhibited by the DM through the search process.

   When further improvement is no longer necessary or is impossible then a compromise has been reached.

   This can be seen as a local optimum relative to an implicit criterion. An example of this approach is the STEP Method (STEM) [Benayoun, 1971].

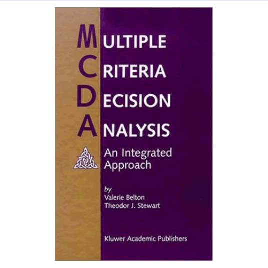
Multi-Criteria Decision Making



When to Get the Preference Information?

A very important issue in MCDM is the moment at which the DM is required to provide preference information. There are three ways of doing this [Evans, 1984, Horn, 1997]:

1. Prior to the search (**a priori** approaches).
2. During the search (**interactive** approaches).
3. After the search (**a posteriori** approaches).



When to Get the Preference Information?

There is a considerable body of work in OR involving approaches performing prior articulation of preferences (i.e., *a priori* techniques).

The reason for its popularity is that any optimization process using this *a priori* information becomes trivial. The main difficulty (and disadvantage of the approach) is finding this preliminary global preference information.

When to Get the Preference Information?

That is the reason why despite the popularity of *a priori* schemes to articulate preferences, interactive approaches (i.e., the progressive articulation of preferences) have been normally favored by researchers [Gardiner, 1994] for several reasons [Monarchi, 1973]:

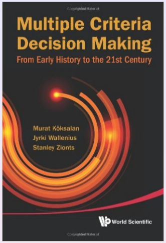
1. Perception is influenced by the total set of elements in a situation and the environment in which the situation is embedded.
2. Individual preference functions or value structures cannot be expressed analytically, although it is assumed that the DM subscribes to a set of beliefs.
3. Value structures change over time, and preferences of the DM can change over time as well.
4. Aspirations or desires change as a result of learning and experience.
5. The DM normally looks at trade-offs satisfying a certain set of criteria, rather than at optimizing all the objectives at a time.

When to Get the Preference Information?

However, interactive approaches also have some problems, mainly related to the preference information that the DM has to provide during the search [Starr, 1977].

For example, the DM can be asked to rank a set of solutions, to estimate weights or to adjust a set of aspiration levels for each objective. None of these tasks is trivial and very often DMs have problems providing answers that can guide the search in a systematic way towards a best compromise solution [Evans, 1984].

In fact, despite the existence of sophisticated algorithms to transform the information given by the DM to a mathematical model that can be used to guide the search, it has been shown that interactive approaches that adopt a simple trial-and-error procedure tend to be highly competitive [Wallenius, 1975].



When to Get the Preference Information?

This indicates that the preference information provided by a DM tends to be so contradictory and inconsistent that in some cases it can be even disregarded without significantly affecting the outcome of a decision-making algorithm.

That is the reason why MCDA emphasizes the decision-making process itself, as there are many factors that could contribute to this inconsistent behavior

Multi-Criteria Decision Making

When to Get the Preference Information?

The use of *a posteriori* approaches is also popular in OR field. The main advantage of these approaches is that no utility function is required for the analysis, since they rely on the use of a “more is better” assumption [Evans, 1984].

The main disadvantages of *a posteriori* approaches are the following [Evans84]:

1. The algorithms used with these approaches are normally very complex and tend to be difficult for the DM to understand.
2. Many real-world problems are too large and complex to be solved using this sort of approach.
3. The number of elements of the Pareto optimal set that tends to be generated is normally too large to allow an effective analysis from the DM.

When to Get the Preference Information?

It is also possible to combine two or more of these approaches. For example, one could devise an approach in which the DM is asked some preliminary information before the search, and then ask the DM to adjust those preferences during the search. This may be more efficient than using either of the two approaches independently.

Finally, it is worth mentioning that EMO researchers often disregard the importance of MCDM without taking into consideration that it normally requires a considerable amount of time (perhaps more than the search itself).

It is well known in the OR community that defining a good utility function or preference structure for a real-world problem is normally a complex task that may take days or even weeks [Moskowitz, 1978].

In fact, operations researchers distinguish a trade-off between spending more time working on a good utility function (or preference structure) and spending more time searching through a larger set of solutions [Evans, 1984].

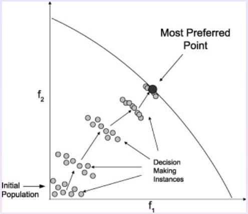
Incorporation of Preferences in MOEAs

The previous classification of stages at which preferences can be provided by the DM (i.e., *a priori*, interactively, *a posteriori*) is also used with respect to MOEAs.

The current EMO literature indicates *a priori* approaches, i.e., aggregating approaches in which weights are defined beforehand to combine all the objectives into a single objective function, are very popular.

Specific *interactive* approaches are less common in the EMO literature, although several of the approaches reviewed in this chapter can be used interactively.

Additionally, several MOEAs *could* be used interactively if desired. From the current volume of research, however, one infers that most MOEA researchers assume an *a posteriori* incorporation of preferences. That is because the main research emphasis is in generating Pareto optimal solutions assuming no prior information from the DM.



Incorporation of Preferences in MOEAs

Regardless of the stage at which preferences are incorporated into a MOEA, the goals are clear: the aim is to magnify (i.e., concentrate search on) a certain portion of the Pareto front by favoring certain objectives (or trade-offs) over others.

The goal of incorporating DM's preferences is to find a mechanism allowing a certain MOEA to generate only the portion of the Pareto front corresponding to the preferences expressed by the DM.

Definition of Desired Goals

Apparently, the earliest attempt to incorporate preferences from the user in a MOEA is the multicriteria decision support system developed by Tanaka and Tanino [1992] and further developed by Tanino et al. [1993].

In this work, a genetic algorithm was used to generate members of the Pareto optimal set (using selection based on nondominance) and an *interactive approach* was adopted to incorporate preferences from the DM. This system allowed the DM to express preferences in three different ways:

1. By choosing satisfactory and unsatisfactory solutions from the set presented to the DM.
2. By defining aspiration levels (i.e., goals) in objective space. Solutions far from these goals are then considered unsatisfactory and vice-versa.
3. By defining the worst acceptable levels for each objective. Any value below this level is then considered unsatisfactory.

Masahiro Tanaka and Tetsuzo Tanino, “**Global optimization by the genetic algorithm in a multiobjective decision support system**”, in *Proceedings of the 10th International Conference on Multiple Criteria Decision Making*, Vol. 2, pp. 261–270, 1992.

Multi-Criteria Decision Making

Definition of Desired Goals

Fonseca and Fleming [1993] made a similar proposal at about the same time as Tanaka & Tanino. The proposal consisted of extending the ranking mechanism of MOGA to accommodate goal information as an additional criterion. The goal attainment method [Gembicki, 1974] was used, so that the DM could supply goals at each generation of the MOEA, reducing in consequence the size of the set under inspection and learning, at the same time, about the trade-offs between the objectives.

It should be clear that this is an *interactive* approach (Fonseca and Fleming call it "progressive") since the DM must express preferences along the evolutionary process.

Carlos M. Fonseca and Peter J. Fleming, "**Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization**", in Stephanie Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423, Morgan Kaufmann Publishers, San Mateo, California, USA, 1993.

Definition of Desired Goals

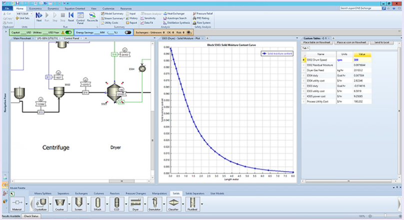
This work was extended in a further paper in which Fonseca & Fleming mathematically define a relational operator incorporating the preference information given by the DM [Fonseca, 1998].

The ranking performed by MOGA is then based on this operator. This operator relies on the use of priorities defined by the DM. In the absence of information all objectives are then given the same priority.

These mathematical definitions provide a much more flexible scheme to incorporate preferences, since their formulation can encompass different decision making strategies (e.g., goal programming and lexicographic ordering).

Carlos M. Fonseca and Peter J. Fleming, "**Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms–Part I: A Unified Formulation**", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **28**(1):26–37, 1998.

Multi-Criteria Decision Making



Definition of Desired Goals

Hinchliffe et al. [1998] used Fonseca & Fleming's approach to incorporate DM's preferences into a multiobjective genetic programming (MOGP) system applied to chemical process systems modeling.

His approach is also used *interactively*, since the goals are tightened as the search progresses.

Mark Hinchliffe, Mark Willis and Ming Tham, "**Chemical Process Systems Modelling using Multi-Objective Genetic Programming**", in John R. Koza et al. (Eds.), *Proceedings of the Third Annual Conference on Genetic Programming*, pp. 134–139, Morgan Kaufmann Publishers, San Mateo, California, USA, July 1998.

Definition of Desired Goals

Shaw and Fleming [1997] experimented with different approaches to incorporating preferences into a multiobjective scheduling problem (solved using MOGA).

Their experiments showed that the use of preferences expressed *a priori* (in the form of attainable or desirable goals provided in a matrix) by the DM was better than using a problem-specific heuristic. They concluded that over constraining the preferences has a significant impact on the performance of the MOEA and therefore, the use of flexible schemes is encouraged.

Since interactive approaches provide some extra flexibility (particularly as it is assumed that certain events may change the DM's preferences over time), an implication of these experiments is that interactive approaches may be more appropriate in real-world applications.

K.J. Shaw and P.J. Fleming, "**Use of Rules and Preferences for Schedule Builders in Genetic Algorithms Production Scheduling**", in *Proceedings of the AISB'97 Workshop on Evolutionary Computation*, Springer-Verlag, Lecture Notes in Computer Science No. 1305, 1997.

Definition of Desired Goals

Another approach based on the definition of desired goals was proposed by Tan et al. [2003]. In this case the DM can define goals that are then used to modify the MOEA's ranking assignment process. The approach also allows the use of both soft and hard constraints.

The approach ranks separately those individuals that satisfy the goals from those that do not satisfy them. This scheme is more flexible than the original proposal by Fonseca and Fleming [1993] because it allows the use of the logical operators AND & OR (these operators were defined using concepts from fuzzy logic), and also allows the specification of "don't care" priorities and non-attainable goals. This aims to reduce the human intervention in the decision-making process.

Although the authors suggest a possible interactive use of the approach they seem to have used it only as an *a priori* technique.

K.C. Tan, E.F. Khor, T.H. Lee and R. Sathikannan, "**An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization**", *Journal of Artificial Intelligence Research*, Vol. 18, pp. 183–215, 2003.

Definition of Desired Goals

Sait and Youssef [1999] proposed a technique in which the DM defines a set of goals (or acceptable limits) for each objective function. Using these goals, the selection mechanism of a MOEA can be modified such that only those solutions that are “nearest” to all the individual goals established are selected. The definition of “nearest” is, in this case, also made using fuzzy logic.

Shibuya et al. [1999] proposed an approach in which the DM has to provide preferences interactively during the evolutionary process. The DM expresses preferences through pairwise comparisons of images in a computer-generated animation application. The approach then sorts the solutions available based on the DM’s preferences.

Sadiq M. Sait, Habib Youseff and Hussain Ali, “**Fuzzy Simulated Evolution Algorithm for Multi-objective Optimization of VLSI Placement**, in *1999 IEEE Congress on Evolutionary Computation*, IEEE Press, pp. 91–97, Washington, D.C., July 1999.

Hajime Kita, Mitsuhiro Shibuya and Shigenobu Kobayashi, “**Integration of multi-objective and interactive genetic algorithms and its application to animation design**”, in *Proceedings of IEEE Systems, Man and Cybernetics*, Vol. III, pp. 646–651, IEEE Press, 1999.

Definition of Desired Goals

Deb [1999] proposed a technique to transform goal programming problems into multiobjective optimization problems which are then solved using a MOEA.

In goal programming, the DM has to assign targets or goals that wishes to achieve for each objective, and these values are incorporated into the problem as additional constraints. The objective function then attempts to minimize the absolute deviations from the targets to the objectives. Deb's approach is used only to perform the transformation from goals to objectives, but it could also be used for incorporating preferences into a MOEA.

The use of weights (i.e., an utility function), deviations from ideal goals (e.g., min-max method) or the direct use of priorities (e.g., the lexicographic method) are all good candidates to incorporate preferences in an approach of this kind (in fact, these three techniques have been coupled to goal programming in the past [Miettinen, 1999].

Kalyanmoy Deb, "**Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms**", in *1999 IEEE Congress on Evolutionary Computation*, pp. 77-84, IEEE Press, Washington, D.C., July 1999.

Multi-Criteria Decision Making

Definition of Desired Goals

Barbosa and Barreto [2001] proposed a co-evolutionary genetic algorithm with two populations: a population of solutions to the problem (i.e., individuals that encode coordinates of all vertices of a graph, since the application is a graph layout problem) and a population of weights (i.e., individuals that contain, each one, a set of weights to be applied on the different aesthetic objectives imposed on the problem).

The decision maker is then presented a set of nondominated solutions and asked to rank them based on subjective preferences. This ranking is then used to determine fitness of the population of weights.

The process is *interactive*, since these preferences are expressed at each iteration of the system. The subjective criteria adopted by the decision maker can be seen as goals that the system has to achieve.

Helio J.C. Barbosa and André M.S. Barreto, "**An interactive genetic algorithm with co-evolution of weights for multiobjective problems**", in Lee Spector et al. (Eds), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 203–210, Morgan Kaufmann Publishers, San Francisco, California, July 2001.

Definition of Desired Goals

Borges and Barbosa [2001] proposed an approach that is similar to goal attainment.

The decision maker is required to provide a constant vector that contains the limit values for each objective function. These limit values represent the minimum (or maximum) values that are acceptable by the decision maker. Then, the original objective functions are penalized based on the values attained (if the values are within the requirements of the decision maker, there is no penalty and the original objective function values are adopted). The form of the penalty function proposed is similar to compromise programming.

This is clearly an *a priori* approach, because the decision maker needs to have prior knowledge of the behavior of each objective function.

Carlos C.H. Borges and Helio J.C. Barbosa, "**Obtaining a Restricted Pareto Front in Evolutionary Multiobjective Optimization**", *Foundations of Computing and Decision Sciences*, Vol. 26, No. 1, pp. 5–21, 2001.

Definition of Desired Goals

Kato et al. [2005] proposed an **interactive fuzzy satisficing method**. In this approach, fuzzy goals are defined and a membership function is elicited from the decision maker for each objective function of the problem to be solved. The method relies on the concept of M-Pareto optimality [Sakawa, 1993] in which an aggregating function is adopted to represent the degree of satisfaction or preference of the decision maker for all the (fuzzy) goals of the problem.

Under this approach, there is one membership function for each objective function. All the membership functions are aggregated into a single expression that uses a minimax formulation, and the decision maker expresses his/her aspiration levels using the so-called reference membership levels. The approach then finds a solution which is as near as possible to the aspiration levels from the decision maker (or even better if such aspiration levels are attainable).

Kosuke Kato, Cahit Perkgoz and Masatoshi Sakawa, "**An Interactive Fuzzy Satisficing Method for Multiobjective Integer Programming Problems through Genetic Algorithms**", in Yaochu Jin (editor), *Knowledge Incorporation in Evolutionary Computation*, Springer, pp. 503–523, Berlin Heidelberg, 2005, ISBN 3-540-22902-7.

Definition of Desired Goals

Yun et al. [2004] proposed the use of **generalized data envelopment analysis** (GDEA) [Yun, 1999] with aspiration levels for choosing desirable solutions from the Pareto optimal set.

This is an *interactive* approach in which a nonlinear aggregating function is optimized by a genetic algorithm in order to generate the Pareto optimal solutions of the multiobjective optimization problem. The decision maker must define his/her aspiration levels for each objective, as well as the ideal values for each of them (these "ideal" values may be, but not necessarily are the components of the ideal vector). Then, the aspiration levels are adopted as constraints during the optimization, so that the Pareto optimal solutions are filtered out and those closest to the aspiration levels are given the highest fitness.

Y.B. Yun, H. Nakayama and M. Arakawa, "**Multiple criteria decision making with generalized DEA and an aspiration level method**", *European Journal of Operational Research*, Vol. 158, No. 3, pp. 697–706, November 1, 2004.

Utility Functions

Fonseca and Fleming [1993] also proposed the use of an expert system to automate the task of the DM. Such an expert system uses built-in knowledge obtained from the preferences expressed *a priori* by the (human) DM.

In this case, a utility function continuously evaluated through the evolutionary process is used.

Carlos M. Fonseca and Peter J. Fleming, "**Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization**", in Stephanie Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423, Morgan Kaufmann Publishers, San Mateo, California, USA, 1993.

Multi-Criteria Decision Making

Utility Functions

Tanaka et al. [1995] experimented with an *interactive* approach using a utility function.

In this case the DM expresses preferences and the system uses a normalized radial basis function network to bias the selection pressure of a genetic algorithm towards those regions that better approximate the preferences.

Masahiro Tanaka, Hikaru Watanabe, Yasuyuki Furukawa and Tetsuzo Tanino, "**GA-Based Decision Support System for Multicriteria Optimization**", in *Proceedings of the International Conference on Systems, Man and Cybernetics*, Vol. 2, pp. 1556–1561, IEEE Press, Piscataway, New Jersey, USA, 1995.

Utility Functions

Hu et al. [1996] and Greenwood et al. [1997] used elements of *imprecisely specified multi-attribute value theory* (ISMAUT) to perform imprecise ranking of attributes.

The idea is that the DM has to rank a set of solutions to the MOP instead of explicitly rank the attributes of the problem (this is implicitly done by the approach). Preference information is also incorporated into the survival criteria used by the MOEA in order to bias the search towards the region of main interest by the DM. Using as a basis the preferences expressed by the DM, the approach derives a set of equations that are solved to find the values of the weights of a utility function that bias the ranking procedure used by the MOEA. This approach is then a compromise between using no preference information of the problem and a utility function (the weights of each attribute are not provided by the DM). This is also an *a priori* approach, since the DM has to express preferences before the search begins.

Garrison W. Greenwood, Xiaobo Sharon Hu and Joseph G. D'Ambrosio, "**Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings**", In Richard K. Belew and Michael D. Vose (Eds.), *Foundations of Genetic Algorithms 4*, pp. 437–455, Morgan Kaufmann Publishers, San Mateo, California, USA, 1997.

Utility Functions

An interesting approach called “Guided Multi-Objective Evolutionary Algorithm” also exploiting the concept of utility function is proposed by Branke et al. [2001].

The idea is to express the DM’s preferences in terms of maximal and minimal linear weighting functions, corresponding directly to slopes of a linear utility function. The authors determine the optimal solution from a population using both of the previously mentioned weighting functions. Those individuals are given rank one and are considered the borderline solutions (since they represent extreme cases of the DM’s preferences). Then all nondominated vectors are evaluated in terms of these two linear weighting functions. After that, all solutions that have a better fitness than either of the two borderline individuals (according to at least one of the two linear weighting functions) are assigned the same rank (these are the individuals preferred by the DM). These solutions are removed from the population and a similar ranking scheme is applied to the remaining individuals.

Utility Functions

It should be clear that this approach is a biased version of the NSGA. In a more recent version [Branke, 2005], this approach was coupled to the NSGA-II.

Jürgen Branke and Kalyanmoy Deb, "**Integrating User Preferences into Evolutionary Multi-Objective Optimization**", in Yaochu Jin (editor), *Knowledge Incorporation in Evolutionary Computation*, Springer, pp. 461–477, Berlin Heidelberg, 2005, ISBN 3-540-22902-7.

Utility Functions

Meneghetti et al. [1999] use a technique called LUTA (**Local UTility Approach**).

In this approach, the DM is asked which solution prefers (out of a set of nondominated vectors previously generated). No other specific information is required (e.g., to perform pairwise comparisons or to justify a certain choice).

The algorithm then proceeds in two stages. In the first stage, it checks any possible inconsistencies in the DM's preferences (e.g., intransitivities) and advises the DM to remove them. In the second stage, it proposes a composite utility function made up of the sum of a set of piecewise linear utility functions, one for each objective function under consideration. The DM's preferences are then stated as inequalities between the utility functions and the solutions available. In other words, these inequalities define a feasible search space for the algorithm, so that the definition of the composite utility function can be modified such that the search is constrained to solutions that lie within the feasible region of the problem.

Utility Functions

This is really an *a posteriori* approach, although some further (local) search may be required to refine the final solution that is presented to the DM. This search is, however, performed with a hillclimber and not with the MOEA used to generate the elements of the Pareto optimal set.

G. Meneghetti, V. Pediroda and C. Poloni, "**Application of a Multi Objective Genetic Algorithm and a Neural Network to the Optimisation of Foundry Processes**", in Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki and Jacques Périaux (Eds), *Evolutionary Algorithms in Engineering and Computer Science*, Chapter 23, pp. 457–470. John Wiley & Sons, Ltd, Chichester, UK, 1999.

Preference Relations

Cvetković and co-workers [2002, 2005], proposed the use of binary preference relations that can be expressed qualitatively (i.e., using words such as "less important"). These preferences are translated to quantitative terms (i.e., weights) to narrow the search of a MOEA.

The weights generated can be used with a simple aggregating approach (i.e., a sum of weights) or with Pareto ranking. In the second case, the weights are used to modify the definition of nondominance used by the ranking scheme of the MOEA.

This approach has some resemblance with the Surrogate Worth Trade-Off method [Haimes, 1974], but unlike that method, binary preference relations can find concave portions of the true Pareto front.

Dragan Cvetkovic and Ian C. Parmee, "**Preferences and their Application in Evolutionary Multiobjective Optimisation**", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp. 42–57, February 2002.

Multi-Criteria Decision Making

Preference Relations

This is also an *a priori* approach since the weights are assumed constant throughout the optimization process, but nothing in the approach really precludes its use in an *interactive* way.

However, there may be some practical issues to take into account if the approach is used interactively, since the DM is asked a considerably high number of questions to make it possible to translate qualitative preferences into quantitative values. This could become too expensive (computationally speaking) if done repeatedly along the evolutionary process.

Dragan Cvetkovic and Carlos A. Coello Coello, "**Human Preferences and Their Applications in Evolutionary Multi-Objective Optimization**", in Yaochu Jin (editor), *Knowledge Incorporation in Evolutionary Computation*, Springer, pp. 479–502, Berlin Heidelberg, 2005, ISBN 3-540-22902-7.

Preference Relations

The direct use of weights to estimate the importance of solutions that have been already identified as Pareto optimal has been suggested by other researchers in the evolutionary computation community in the past.

For example, Bentley and Wakefield [1997]. define a property called "importance" which provides additional information regarding the sort of solutions that are preferred by the DM.

This property is then coupled to a GA using ranking, but no Pareto ranking. The authors sort the fitness values of each objective separately and then rank solutions based on their ordering; the average ranking of each individual is then used as its fitness. This ranking scheme is very similar to the one previously proposed by Vemuri and Cedeño [1995, 1996] and does not quite correspond to the concept of Pareto ranking normally adopted by EMO researchers.

P.J. Bentley and J.P. Wakefield, "**Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms**", In P.K. Chawdhry et al. (Eds), *Soft Computing in Engineering Design and Manufacturing*, Part 5, pp. 231–240, London, June 1997.

Preference Relations

Once individuals are ranked, the "importance" that the DM assigns to each objective can be used as a weight and the fitness of an individual is now a weighted sum of its average ranking. Since the weights are assigned by the DM before actually performing the search, this is also an *a priori* technique. Obviously, it could also be used interactively, but apparently was never tested that way.

It is important to mention that the focus of the research conducted by Cvetković [2000] is broader and more related to some OR work.

Also, it is worth mentioning that this is one of the few attempts to develop a formal decision making model explicitly for evolutionary multiobjective optimization algorithms.

Dragan Cvetkovic, "Evolutionary Multi-Objective Decision Support Systems for Conceptual Design", PhD thesis, School of Computing, University of Plymouth, Plymouth, UK, November 2000.

Preference Relations

Drechsler and co-workers [2001] proposed the use of satisfiability classes to model preferences from the decision maker.

The approach consists of defining a relation “favor”, whose concept is similar to Pareto dominance, but not equivalent (mathematically speaking, the relation “favor” is not a partial order, because it is not a transitive relation).

In this case, the search space is divided into several categories (e.g., superior, very good, good, satisfiable, and invalid). Solutions generated are then analyzed in terms of their “quality” (defined in terms of the priorities of the decision maker) and divided into several satisfiability classes (i.e., solutions of similar quality belong to the same satisfiability class).

Nicole Drechsler, Rolf Drechsler and Bernd Becker, “**Multi-objective Optimisation Based on Relation *favour***”, in Eckart Zitzler et al. (Eds.), *First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 154-166. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

Preference Relations

After sorting the satisfiability classes with respect to their quality, a ranking of the solutions is obtained. Preference relations are modeled in this case using a directed graph that is recomputed at each generation without any human intervention.

In this approach, incomparable solutions may be placed in different satisfiability classes, even if the decision maker wants them to be within the same class. Also, goals are considered in parallel and not relative to each other.

Preference Relations

Another interesting proposal to directly modify the ranking procedure of an evolutionary algorithm using preference information provided by the user is made by Hughes [2000, 2001].

In this case, the author proposes the use of expressions that incorporate information about constraint violation and priority satisfaction into the formulas used to compute domination probabilities (i.e., this information is used to alter the ranking procedure of the population). The approach is simple and elegant, and the author finds it particularly useful when dealing with noisy fitness functions.

Evan J. Hughes, "**Evolutionary Multi-objective Ranking with Uncertainty and Noise**", in Eckart Zitzler et al. (Eds), *First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 329–343. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

Outranking

Rekiek and co-workers [2000] proposed the use of an outranking method called PROMETHEE II [Brans, 1994] combined with a MOEA.

PROMETHEE II computes a net flow for each individual and this value is used to rank the population (it imposes a complete preordering of preferences). Weights assigned to each objective by the DM (called preference indexes) have an impact in the computation of the net flows and impose an ordering on the solutions found.

The approach is used in an *interactive* way since the DM has to adjust the weights along the evolutionary process. An interesting aspect of this work is the use of what the authors called "branching on population" which basically consists of creating intermediate states based on the preferences expressed by the DM interactively. This allows the evolutionary algorithm to restart from one of these intermediate states rather than from the very beginning.

Brahim Rekiek, Pierre De Lit, Fabrice Pellichero, Thomas L'Eglise, Emanuel Falkenauer and Alain Delchambre, "**Dealing With User's Preferences in Hybrid Assembly Lines Design**", in *Proceedings of the MCPL'2000 Conference*, 2000.

Outranking

Guimarães Pereira [1995, 1996] uses ELECTRE I coupled with a genetic algorithm to generate alternative routes between two geographical locations.

Routes generated are compared in a pairwise fashion for each of the objectives under consideration. An index describing the outranking relation (i.e., the DM's preferences) of a pair of individuals is calculated based on these pairwise comparisons. Then, an aggregating formula is used to combine the different indexes generated from these comparisons. Thus, this aggregating formula is used in an *interactive* way to narrow the search of the GA.

Angela Guimaraes Pereira, "**Generating alternative routes by multicriteria evaluation and a genetic algorithms**", *Environment and Planning B: Planning and Design*, **23**:711-720, 1996.

Outranking

In further work, Guimarães Pereira [1997] and Haastrup & Guimarães Pereira [1997] couple PROMETHEE to a MOEA.

In this case, at each generation of the MOEA, a set of alternatives (i.e., nondominated vectors) are generated and pairwise comparisons of alternatives are performed. The different alternatives available are then ranked in terms of preferences defined over certain suitability values defined by the DM. An aggregating approach is used to measure if a certain alternative is preferred over another one. Actually, the author computes differences between the intensity of preference (this is defined as a membership function, using fuzzy logic) on each objective for each pair of alternatives under consideration.

This approach is applied in an *interactive* way during the evolutionary process.

Angela Cristina Martinho Guimaraes Pereira, "**Extending Environmental Impact Assessment Processes: Generation of Alternatives for Siting and Routing Infraestructural Facilities by Multi-Criteria Evaluation and Genetic Algorithms**", PhD thesis, New University of Lisbon, Lisbon, Portugal, 1997.

Outranking

Massebeuf et al. [1999] also used PROMETHEE II combined with a MOEA (in this case, a technique called “Diploid Genetic Algorithm” [Perrin, 1997]).

The DM is asked to express preferences for each pairwise comparison of alternatives (e.g., preference or indifference). Additionally, the DM has to assign weights that express relative importance of the objectives. This information is used to compute a global concordance index and a discordance index (as in ELECTRE III [Roy, 1993]).

Using these indexes, the authors generate outranking degrees for every pair of alternatives. This allows to rank the alternatives so that a final recommendation can be made to the DM.

Silvere Massebeuf, Christian Fonteix, Laszlo N. Kiss, Ivan Marc, Fernand Pla and Kazimierz Zaras, “**Multicriteria Optimization and Decision Engineering of an Extrusion Process Aided by a Diploid Genetic Algorithm**”, in *1999 IEEE Congress on Evolutionary Computation*, pp. 14–21, IEEe Press, Washington, D.C., USA, July 1999.

Outranking

Parreiras and Vasconcelos [2005] also used PROMETHEE II combined with a MOEA.

In this case, the authors adopted a Gaussian preference function and a (normalized) aggregating function for computing the global preferences. The implementation developed by the authors is called *Smart* and is applied in an *a posteriori* way, to nondominated solutions generated by the NSGA-II.

Roberta O. Parreiras and João A. Vasconcelos, "**Decision Making in Multiobjective Optimization Problems**", in Nadia Nedjah and Luiza de Macedo Mourelle (editors), *Real-World Multi-Objective System Engineering*, pp. 29–52, Nova Science Publishers, New York, 2005.

Multi-Criteria Decision Making

Fuzzy Logic

Voget [1996] and Voget and Kolonko [1998] used a fuzzy controller that automatically regulates the selection pressure of a MOEA by using a set of pre-established goals defining the "desirable" behavior of the population.

A set of fuzzy rules is used to modify the selection mechanism of the MOEA when it is deviating from the goals defined by the DM.

Although the approach is used only to keep diversity in the population, it could easily be extended to incorporate preferences of the DM. The idea is similar to goal attainment, except that in this case membership functions are used to express goals in vague terms (i.e., it allows uncertainties). This is an *a priori* approach, but it could also be used interactively.

Stefan Voget and Michael Kolonko, "**Multidimensional Optimization with a Fuzzy Genetic Algorithm**", *Journal of Heuristics*, **4**(3):221-244, September 1998.

Fuzzy Logic

Similar fuzzy controllers were proposed by Esbensen & Kuh [1996], and Lee & Esbensen [1997] but in this last case, on-line and off-line performance of the MOEA are used to guide the search, so that the following conditions are satisfied [Lee, 1997]:

- Maximize the diversity of the nondominated vectors in the population.
- Maximize the number of nondominated vectors in the population.
- Maximize the bounding volume of the set of nondominated vectors.
- Make the center of gravity of the final solution set close to the origin.

All of these conditions aim to promote diversity and bias the MOEA towards the true Pareto front of the problem solved. Additionally, the DM's preferences are incorporated using a simple utility function (a weighted sum). The approach is used *a priori*, but the authors suggest its possible use as an *interactive* method.

M. A. Lee and H. Esbensen, "**Fuzzy/Multiobjective Genetic Systems for Intelligent Systems Design Tools and Components**", In Witold Pedrycz (Ed.), *Fuzzy Evolutionary Computation*, pp. 57-80. Kluwer Academic Publishers, Boston, Massachusetts, 1997.

Fuzzy Logic

Pirjanian [1998] used fuzzy rules to generate weights that would narrow the search of a traditional multiobjective optimization technique. This work is extended in Pirjanian & Matarić [1999] where, instead of adopting the usual fuzzy behavior-based control, where fuzzy rules are combined using standard fuzzy inferencing, the authors use multicriteria decision making for optimizing the behavior of a robot: an action is chosen which maximizes the objective corresponding to different behaviors.

The process operates in four stages:

1. Find feasible actions, based on physical and other hard constraints.
2. Find Pareto optimal solutions.
3. Find satisficing actions from the Pareto optimal solutions incorporating subjective knowledge.
4. Find the most preferred action, using other criteria.

Fuzzy Logic

In [Pirjanian, 1999], a single robot is used with the task of reaching a certain target. However, in [Pirjanian, 2000], the method is generalized to several robots (in the experiments reported, two robots are used) with the task of surrounding and capturing the target. The authors use weighted sums to generate the Pareto front (by varying the weights) and adopt lexicographic ordering, goal programming and interval criterion weights for finding satisficing actions [Pirjanian, 2000]. In all cases, the approaches seem to be applied *a posteriori*.

Paolo Pirjanian and Maja Matarić, "**Multi-robot target acquisition using multiple objective behavior coordination**", in *Proceedings of the International Conference on Robotics and Automation (ICRA'2000)*, pp. 2696–2702, San Francisco, California, USA, 2000.

Fuzzy Logic

Jin & Sendhoff [2002] proposed an approach for converting fuzzy preference relations into interval-based weights which are then combined with a dynamic weighted aggregation method proposed by the same authors [Jin, 2001].

This approach is very similar to the proposal of Cvetković & Parmee [2002], but instead of converting the fuzzy preferences into single-valued weights, they are converted to interval-based weights. So, the authors adopt preference matrices and real-valued preference relation matrices. One interesting side-effect of using intervals is that the dynamic weighted aggregation method cannot properly work with non-convex Pareto fronts in this case, because in such situations the movements of the individuals cannot be controlled based on the fuzzy preferences. However, this approach could obviously be used with an alternative multi-objective evolutionary algorithm that does not have this limitation.

Yaochu Jin and Bernhard Sendhoff, "**Incorporation of Fuzzy Preferences into Evolutionary Multiobjective Optimization**", in E.Cantú-Paz et al. (Eds), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pp. 683, Morgan Kaufmann Publishers, San Francisco, California, USA, July 2002.

Fuzzy Logic

Wang and Terpenny [2005] proposed an approach based on the use of a fuzzy set-based aggregating function to express the preferences from the user. This aggregating function relies on two things: (1) a set of weights that express the importance of the design attributes and (2) a degree of compensation among these design attributes. By changing the weights and the compensation factors, different portions of the Pareto front can be obtained.

This approach is used in an *interactive* way within an agent-based system in which the user expresess his/her preferences and a set of agents produce subsolutions that are later refined based on the constraints and the fuzzy preference aggregating function previously indicated. An interesting aspect of this work is that the actual parameters of the fuzzy preference aggregating function are really *learnt* using a neural network that attempts to minimize the cumulative error of the network.

Jiachuan Wang and Janis P. Terpenny, "**Interactive Preference Incorporation in Evolutionary Engineering Design**", in Yaochu Jin (editor), *Knowledge Incorporation in Evolutionary Computation*, Springer, pp. 525–543, Berlin Heidelberg, 2005, ISBN 3-540-22902-7.

Fuzzy Logic

Farina and Amato [2002, 2003, 2004] analyzed the limitations and drawbacks of the Pareto optimality definition when dealing with problems that have more than 3 objectives. The three main reasons that the authors provide for considering the definition of Pareto optimality as unsatisfactory are the following:

1. It does not consider the number of improved or equal objective function values.
2. It does not consider the (normalized) size of the improvements.
3. It does not consider preference among objectives.

Based on this analysis, they proposed three alternative definitions of optimality that aim to generalize the definition of Pareto optimality: k-optimality, k_F-optimality (with fuzzy numbers) and fuzzy optimality.

M. Farina and P. Amato, "**On the Optimal Solution Definition for Many-criteria Optimization Problems**", in *Proceedings of the NAFIPS-FLINT International Conference'2002*, pp. 233–238, IEEE Service Center, Piscataway, New Jersey, June 2002.

Fuzzy Logic

These definitions cope with the previous limitations of the definition of Pareto optimality. An important aspect of this work is the fact that in this case fuzzy logic is not adopted for the treatment of the user's preferences, but for modeling the size of the improvements done in each objective.

Thus, in this model, all objectives are given the same importance as traditionally done with the Pareto optimality definition. The approach, however, is adopted to extract subsets of solutions from the Pareto optimal set, but instead of directly expressing preferences as membership functions, the approach allows the definition of fuzzy tolerances for the objectives.

M. Farina and P. Amato, "**Fuzzy Optimality and Evolutionary Multiobjective Optimization**", in Carlos M. Fonseca et al. (Eds.), *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pp. 58–72, Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal, April 2003.

Compromise Programming

Deb [1999, 2001] suggested a variation of compromise programming to bias the sharing procedure of the NSGA.

Deb uses a normalized Euclidean distance between objective vectors (as normally used to compute sharing distances in the NSGA), but introduces unequal weights such that different importance can be assigned to each objective. This allows one to bias the niche-formation procedure of a MOEA, but does not produce a single final solution as normally done with multi-criteria decision making techniques. That means that further intervention of the DM is still required. Therefore, this is an *a posteriori* approach.

Kalyanmoy Deb, "**Nonlinear goal programming using multi-objective genetic algorithms**", *Journal of the Operational Research Society*, Vol. 52, No. 3, pp. 291–302, 2001.

Multi-Criteria Decision Making

Compromise Programming

In more recent work, Branke & Deb [2005] proposed a biased crowding measure that is applied to the NSGA-II.

The idea is that the user provides his/her preference as a direction vector, which is really a central linearly weighted utility function. The crowding measure will be biased based on this direction vector, so that only a fraction of the Pareto front is produced. This crowding measure has the advantage of being easily scalable to any number of objectives and to work well even in the presence of non-convex portions of the Pareto front.

It is worth indicating that the version of the NSGA-II adopted in this work uses a modified crowding distance that is more appropriate to deal with problems of higher dimensionality than the original proposal.

Jürgen Branke and Kalyanmoy Deb, "**Integrating User Preferences into Evolutionary Multi-Objective Optimization**", in Yaochu Jin (editor), *Knowledge Incorporation in Evolutionary Computation*, Springer, pp. 461–477, Berlin Heidelberg, 2005, ISBN 3-540-22902-7.

Compromise Programming

Sakawa et al. [1994] also suggested an approach based on compromise programming.

In this case, the DM establishes goals for each objective using membership functions (i.e., fuzzy logic). A *minimum* operator is defined so that it can integrate all the preferences from the DM into a single membership function. Several expressions inspired on compromise programming are then used to guide the search of the MOEA.

These expressions are used as online performance measures having a direct impact on the selection process. The approach was applied to multidimensional versions of the knapsack problem.

Masatoshi Sakawa, Masahiro Inuiguchi, Hideaki Sunada and Kazuya Sawada, "**Fuzzy Multiobjective Combinatorial Optimization Through Revised Genetic Algorithms**", *Japanese Journal of Fuzzy Theory and Systems*, **6**(1):77-88, 1994.

Suggested Readings

Carlos A. Coello Coello, "**Handling Preferences in Evolutionary Multiobjective Optimization: A Survey**", in *2000 IEEE Congress on Evolutionary Computation*, Vol. 1, pp. 30–37, IEEE Press, Piscataway, New Jersey, USA, July 2000.

Matthias Ehrgott, José Rui Figueira and Salvatore Greco (Editors), **Trends in Multiple Criteria Decision Analysis**, Springer, International Series in Operations Research and Management Science, 2010, ISBN 978-1-4419-5903-4.

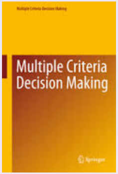
Tobias Wagner and Heike Trautmann, "**Integration of Preferences in Hypervolume-Based Multiobjective Evolutionary Algorithms by Means of Desirability Functions**", *IEEE Transactions on Evolutionary Computation*, Vol. 14, No. 5, pp. 688-701, October, 2010.

Suggested Readings

Tobias Wagner, Heike Trautmann and Dimo Brockhoff, "**Preference Articulation by Means of the R2 Indicator**", in Robin C. Purshouse et al. (Eds), *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013*, pp. 81–95, Springer. Lecture Notes in Computer Science Vol. 7811, Sheffield, UK, March 19-22, 2013.

Salem F. Adra, Ian Griffin and Peter J. Fleming, "**A Comparative Study of Progressive Preference Articulation Techniques for Multiobjective Optimisation**", in Shigeru Obayashi et al. (Eds), *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pp. 908–921, Springer. Lecture Notes in Computer Science Vol. 4403, Matshushima, Japan, March 2007.

Slim Bechikh, Marouane Kessentini, Lamjed Ben Said and Khaled Ghédira, "**Preference Incorporation in Evolutionary Multiobjective Optimization: A Survey of the State-of-the-Art**", in Ali R. Hurson (editor), *Advances in Computers*, Chapter 4, pp. 141–207, 2015, Elsevier, ISBN 978-0-12-802132-3.



Issues Deserving Attention

Regardless of the approach used to handle DM's preferences in an MOEA, there are several issues that should be kept in mind:

- Preserving dominance.
- Transitivity.
- Scalability.
- Group decision making.

Each of them will be briefly discussed next.

Preserving Dominance

It is important to make sure that the preference relationships introduced in the MOEA preserve existing dominance relationships. Although this may seem straightforward, this is not always the case.

If our proposed preference relationships do not preserve dominance, the search would be biased towards undesired regions of the search space.

In spite of the fact that this property can be easily preserved in most cases (e.g., [Greenwood, 1997]), it should be nevertheless kept in mind when proposing approaches that incorporate preferences into a MOEA.

Transitivity

The use or lack of intransitivities has been the subject of much debate in the OR field [Munda, 1993]. It has been argued by some researchers that human beings tend to define preferences that are not necessarily transitive, and there are several examples in the OR literature in which intransitivities of preferences easily occur (e.g., [Huylenbroeck, 1995]).

The main argument against allowing intransitivities is that their absence considerably simplifies the modeling of preferences; intransitivities can lead to contradictions that are much more difficult to handle. Also, by leaving intransitivities out, the decision model becomes mathematically tractable.

However, the issue remains open, and the French school of MCDM prefers to use outranking procedures that allow intransitivities. However, outranking procedures have been combined with MOEAs by only a few researchers (as seen in the previous section) and utility functions seem to be preferred.

Scalability

Some early researchers indicated that MAUT was sound only when few attributes were considered [Zeleny, 1977].

MOEAs in general are victim of the “dimensionality curse”, because they tend to become cumbersome or even useless as the number of objectives is increased (the so-called many-objective problems).

Some of the approaches reviewed here, such as preference relations, are very sensitive to the number of objectives and to changes in the order of the questions asked to the DM. Therefore, they are most likely impractical in applications with a large number of objectives.

Interestingly, aggregating approaches are less sensitive to scalability. An aggregating approach can easily manipulate a large number of objective functions and preferences, because they are integrated into a single scalar value (e.g., [Cardon, 2000] where up to 500 objectives are considered at a time). However, the effect of aggregation is diluted as the number of factors to be mixed is increased.

Multi-Criteria Decision Making

Group Decision Making

It is not trivial to get a DM to express preferences in a consistent way for an arbitrary problem. If this task is by itself difficult, incorporation of preferences from a group of DMs is even more complicated. Unfortunately, the use of group preferences is not an uncommon situation in real-world applications but its introduction raises additional questions.

If there is a group of DMs, each of them probably has their own objectives and priorities. Therefore, some form of negotiation is necessary in order to reach a consensus. Normally, a moderator intervenes to solve the many conflicts that could arise from these situations.

The members of the group can express their preferences independently and leave it to the moderator to integrate them. Alternatively, they could be asked to debate and to establish a consensus regarding their priorities (even if this could take a considerable amount of time). In the latter case, the approach used to integrate preferences has to adapt to the ordering of preferences that represents the collective opinion of the group [Hwang, 1987].

Group Decision Making

The most common approach is the first, in which the preferences of every individual DM are aggregated into a single utility function that represents the unified preferences of the group.

However, the economist Kenneth J. Arrow [1951] showed that apart from some very special cases, utility functions cannot be used to aggregate individual preferences into a group utility function. The so-called **Arrow's Impossibility Theorem** has very important consequences in MCDM. To explain how it works, consider the following assumptions [Keeney, 1993]:

- **Complete Domain**: The utility function should be able to define an ordering for the group, regardless of the individual members' ordering.
- **Positive Association of Social and Individual Values**: If the group ordering indicates that alternative x is preferred to alternative y for a certain set of individual rankings, and (1) if there are no changes on the ordering of each individual, and (2) each individual's paired comparison against x remains unchanged or is modified in x's favor, then the group ordering must imply that x is still preferred to y.

Group Decision Making

- **The Independence of Irrelevant Alternatives**: If an alternative is eliminated and the preference relations for the remaining alternatives remain unchanged for all the individuals, then the new group ordering should remain the same as before.
- **Individual's Sovereignity**: For each pair of alternatives x and y, there is some set of individual orderings which causes x to be preferred to y.
- **Non-dictatorship**: It is impossible that the preferences of the group be always in agreement with the preferences of a single individual.

Group Decision Making

So, what Arrow's Impossibility Theorem says is that any joint decision process which is reasonably democratic and respectful of individuality (following the assumptions described before) is also irrational or unreliable.

It is likely to have at least one of the following problems: a) the order of the decisions affects the final outcome, b) the independence of its elements might not be respected, and c) the unanimous will of its elements might be ignored. A classical example of Arrow's Impossibility Theorem uses three candidates and three voters with the preferences indicated in the Table shown in the next slide.

Multi-Criteria Decision Making

Individual	Preferences	A vs. B	B vs. C	A vs. C
1	$A > B > C$ $A > C$	A	B	A
2	$B > C > A$ $B > A$	B	B	C
3	$C > A > B$ $C > B$	A	C	C
Group Preferences		**A > B**	**B > C**	**C > A**

Group Decision Making

In the previous slide, we show the voting preferences of three rational individuals on three candidates. $A > B$ means that A is preferred over B.

While each individual has a rational set of preferences, it is obvious that combining these to form a group utility function presents a problem (the group utility relation is cyclic, i.e., it is not transitive). In consequence, optimization for the group using this data is impossible.

Some authors such as Hazelrigg [1996] argue that this situation is not a rare case, but is in fact the norm and as greater detail about the preferences of individuals within a group are provided, the higher the chance of encountering this type of problem.

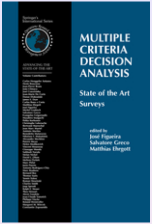
Some authors have shown that Arrow's conditions can be ignored in practical problems [Scott, 1999], but its mere existence has triggered a considerable amount of research in economics [Roy, 1973], and cannot be disregarded by EMO researchers.

Other Important Issues

Finally, it is important to establish a set of characteristics that the ideal scheme to incorporate preferences should have. In that regard, the seven prerequisites for a good MCDA approach, defined by Brans and Mareschal [1994] are provided next:

1. The approach should take into account the amplitude of the deviations between the alternatives.
2. Scaling should not be required, despite the fact that the criteria of a problem could be (and normally are) expressed in different units.
3. When comparing two alternatives *a* and *b*, the MCDA technique should arrive at one of the following conclusions:
   - *a* is preferred to *b*, or *b* is preferred to *a*.
   - *a* and *b* are indifferent.
   - *a* and *b* are incomparable.

Multi-Criteria Decision Making



Other Important Issues

4. The method should be understandable by the DM. Therefore, black-box effects should be avoided.
5. Parameters that have no economical significance should not be included in the approach.
6. The analysis of the conflicting aspects of the criteria must be available.
7. It is important to have a clear interpretation of the weights of the criteria.