Taibi Boumedyen [a]
a - PhD,*, Faculty of Economics, Business and Management, Saida University,*
didenmajor@yahoo.fr

Multiple objective combinatorial optimization problems

Abstract :

Many sectors are concerned with complex problems of great dimention that must be optimized. These optimization problems are seldom single-objective : usually, there are several contradictory criteria or objectives that must be satisfied simultaneously. Multi-objective optimization is a discipline centered in the resolution of this kind of problems. It has its roots in the 19th century in a work economy of Edgeworth and Pareto. Initially, it was applied to economic sciences and management, and gradually to engineering sciences.

Combinatorial optimization is a held extensively studied by many researchers. Due to its potential for application in real world problems.

In this paper, we present a general formulation of Multiple objective combinatorial optimization (MOCO) problems, describe the main characteristics of MOCO problems, and the most important properties and theoretical results for these problems. Also, we propose to enrich the surveys by providing an analysis of recent innovative approaches in this domain.

Key words :

Multiple objective, combinatorial optimization, multicriteria analysis, efficient solutions.

Introduction

Multiple objective combinatorial optimization (MOCO) has become a quickly growing field in multiple objective optimization, and has recently attracted the attention of researchers both from the fields of multiple objective optimization and from single objective integer programming [Ehrgott and Gandibleux (2000)].

Many real world decisions cannot be performed by comparing alternatives through a single value. Instead, it is more realistic to associate several values to each alternative, according to several criteria, points of view or states of the world. In the case of criteria, this gave rise to the subject of *multiple criteria decision making* (MCDM). MCDM mainly embraces *multiple criteria decision aid* (MCDA) and *multi-objective optimization* (MOO). While the former focuses on situations where the set of alternatives of a decision problem is defined in extension and is assumed to be of small cardinality, the latter faces a large, possibly infinite, set of feasible solutions. Therefore, most developments in MCDA focus on modeling issues, trying to determine and use in the best way the decision maker's preferences in order to formulate the most valuable recommendation regarding a practical decision problem. In contrast, MOO primarily faces computational issues arising from multiple and conflicting objective functions even when the single objective version of the problem is easily solved. MCDM in general finds numerous real world applications and so does MOO:[Ehrgott (2008)] details some of them in various fields such as finance, transportation, medicine, and telecommunication. Also [Clímaco and Pascoal (2012)] briefly survey recent applications in routing problems and telecommunication networks.

Several practically efficient algorithms have been proposed from the end of the seventies to enumerate efficient solutions in multi-objective combinatorial optimization (MOCO). Most proposals first addressed the bi-objective case which has interesting and exclusive properties compared to higher dimensional problems. The generalization of these algorithms to more than two objective functions started to produce good practical results during the last decade. It turns out that most exact multi-objective algorithms fall into one of the two following categories.

The first category corresponds to implicit enumeration, that is, iterative partitioning of the instance and elimination of dominated subinstances. Thus this kind of algorithm primarily acts in the decision space. Two main approaches are concerned: multi-objective branch and bound (MOBB) and dynamic programming [Bazgan et al. (2009)] for the implementation of these two methods in the general multi-objective case.

The second category contains algorithms that primarily act in the objective space, attempting to identify small zones that are individually explored. The two-phase method [Ulungu and Teghem (1995)] for the original method and [Przybylski et al. (2010)] for a generalization to more than two objective functions) falls into this category as well as several methods based on the resolution of budget constrained integer programs that use the integer programming formulation of the underlying problem [Sylva and Crema, (2004)].

1- Multiple Objective Combinatorial Optimization Problems

The feasible set of a (multiobjective) combinatorial problem is defined as a subset $X \subseteq 2^A$ of the power set of a finite set $A = \{a_1, . . ., a_n\}$. For example, consider the minimum spanning tree problem. $G = (V, A)$ is a graph with node set V and the edge set A, the feasible set is the set of spanning trees of G and $X = \{S \subseteq A : S \text{ is a spanning tree of } G\}$.

Typically, in combinatorial optimization two types of objective functions are considered, namely the sum and the bottleneck objective:

$$z\ (S) = \Sigma\ w(a),\ or$$
$$z\ (S) = max\ w(a),$$

Where $S \in X$ and $w : A \rightarrow \mathbb{Z}$ is some weight function.

A combinatorial problem can also be formulated in terms of binary variables.

For this purpose we introduce a variable x_i for each element $a_i \in A$. Then, a feasible solution

$S \in X$ can be represented by a binary vector $x \in \{0,1\}^n$ if we define

$$Xi = \begin{cases} 1 & e_i \in S \\ 0 & else. \end{cases}$$

With this de_nition S = $\{ai : xi = 1\}$. It is therefore equivalent to speak about feasible solutions as subsets of A or about their representations by binary vectors. Accordingly X will be represented by a subset of $\{0,1\}^n$.

In terms of the feasible set, this definition comprises (multiobjective versions of) the shortest path, minimum spanning tree, assignment, knapsack, travelling salesperson, or set covering problems, to mention only a few.

In a multicriteria combinatorial problem several weight functions $wj : A \rightarrow \mathbb{Z}$ are given, yielding several objective functions z^j , $j = 1, . . .,Q$ (usually of the sum or bottleneck type). The problem is then to solve

$$\text{“ Min”} (z^1(S), . . ., z^Q(S))$$

Most often the minimization in (MOCO) is understood in the sense of efficiency (or Pareto optimality). A subset $S \in X$ is called efficient if there does not exist another feasible solution $S' \in X$ such that $z^j(S') \leq z^j(S)$ for all $j = 1, . . .,Q$ with strict inequality for at least one of the objectives. The corresponding vector $z(S) = (z^1(S), . . . , z^Q(S))$ is called nondominated. The set of Pareto optimal (efficient) solutions of (MOCO) will be denoted by E.
However, besides efficiency, there are other definitions of the "min" term in the formulation of (MOCO). For example, one could consider lexicographic minimization, when objective vectors are compared lexicographically: $z(S_1) <_{lex} z(S_2)$ *if* $z^j(S_1) < z^j(S_2)$, where j is the smallest index such that $z^j(S_1) \neq z^j(S_2)$. This could be done with respect to one, or all permutations of the objective functions z^j.
Another possibility is to minimize the worst objective function, i.e.

$$min\ max\ z^j(S).$$

We call this the max-ordering problem in order to distinguish it from the single objective ottleneck problem [Ehrgott and al (1999)].
A combination of the latter two is the lexicographic max-ordering problem, where the vector of objective values $z(S)$ is first resorted in a nonincreasing order of its components, and the resulting vectors are compared lexicographically [Ehrgott (1996)].

In a real world decision context, finally a compromise has to be made among the many efficient solutions that (MOCO) may have. This is the reason why often the existence of a utility function is implicitly or explicitly assumed. A utility function assigns each criterion vector $z(S)$ a scalar overall utility. Then methods are developed to find a solution of maximum utility.

Closely related to combinatorial problems are multiobjective integer programming problems.

These can be formulated as follows.

$$\begin{array}{ll} & \text{" Min "} \quad Cx \\ \textit{subject to} & Ax = b \\ & x_i \geq 0 \quad i = 1, \ldots, n \\ & x_i \ \textit{integer} \quad i = 1, \ldots, n \end{array} \qquad \text{(MOIP)}$$

Here C is a $Q \times$ n objective matrix, A is an $m \times n$ constraint matrix, and $x \in IR^n$. There is a considerable amount of literature on these problems. We refer to some surveys that exist but will not consider the literature in detail. [Climaco and al (1997)] provide surveys of techniques to find efficient solutions for (MOIP), [Teghem and Kunsch (1986)] gives an overview of interactive methods for (MOIP), and [Rasmussen (1986)] surveys (MOIP) with binary variables.

In general, combinatorial optimization problems can be considered as special cases of integer (in particular binary) programming. A MOCO problem is distinguished by a specific set of constraints, that provides a structure to the problem. We focussed on such problems and do not intend to review literature on general multiobjective binary or integer programming.

When the set of feasible solutions is an explicitly given finite set, e.g. $X = A$. In this case, all problems discussed above are efficiently solvable. Algorithms can be found in [Ehrgott (1998)].

To summarize, (MOCO) is a discrete optimization problem, with n variables x_i, $i = 1, \ldots, n$, Q objectives z^j, $j = 1, \ldots, n$ and a specific constraint structure defining the feasible set X.

2- Properties of Multiobjective Combinatorial Optimization Problems

In this section we discuss some of the properties of MOCO problems. It is in order to mention here that there is a considerable number of erroneous statements, even in papers published in international standard refereed journals [Ehrgott and Gandibleux (2000)] . We will point out the most important of these throughout the paper, in the appropriate places.

By its nature, multiobjective combinatorial optimization deals with discrete, non continuous problems, although the objectives are usually linear functions. An essential consequence of this fact when trying to determine the set of all efficient solutions (or nondominated vectors in objective space) is, that it is not sufficient to aggregate the objectives through weighted sums.

It is long known that for multiobjective linear programming problems

$$\text{Min } \{Cx : Ax = b , x \geq 0\}$$

The set of efficient solutions is exactly the set of solutions that can be obtained by solving LP's

$$\text{Min } \{\Sigma \lambda_j c^j x : Ax = b, x \geq 0\},$$

Where $\Sigma \lambda_j = 1$, $\lambda_j > 0$, $j = 1, \ldots, n$. But the discrete structure of the MOCO problem makes this result invalid. Thus there usually exist efficient solutions, which are not optimal for any weighted sum of the objectives. This is true even in cases where the constraint matrix is totally unimodular, contrary to a proposition in [Kouvelis and Carlson (1992)]. These solutions are called nonsupported efficient solutions NE, whereas the remaining are called supported efficient solutions, SE. In early papers referring to MOCO, NE was usually not considered. Most authors focussed on scalarizing the objectives by means of weighting factors λ_j.

Nevertheless, the set NE is important. Usually there are many more nonsupported than supported efficient solutions, see e.g. [Visée and al (1998)] for numerical results. Moreover, the nonsupported solutions contribute essentially to the difficulty of MOCO problems. Below, we shall briey discuss the concepts of computational complexity of (MOCO). For introductions to the theory of *INP*-completeness and *#IP*-completeness we refer to [Garey and Johnson (1979)]. These notions deal with the difficulty of finding a, respectively counting the number of solutions of a (MOCO).

In order to transfer the notions of *IP*, *INP* and *#IP* to MOCO we first introduce a decision problem related to (MOCO) in a straightforward manner:

Given constants $k_1, \ldots, k_Q \in \mathbb{Z}$, does there exist a feasible solution $S \in X$ such that D(MOCO)

$$z^j(S) \leq k_j,\ j = 1, \ldots, Q$$

The corresponding counting problem is:

How many feasible solutions $S \in X$ do satisfy

$$z^j(S) \leq k_j,\ j = 1, \ldots, Q$$

#(MOCO)

It turns out that the respective versions of (MOCO) in the sense of finding or counting efficient solutions are in general *INP*- and *#IP*-complete, respectively. This is true even for problems which have efficient algorithms in the single objective case. We refer to [Emelichev and Perepelitsa (1991)] and [Ehrgott (2000)] for results in this respect. Therefore the development of heuristics with guaranteed worst case performance (bounded error) is interesting. However, not much is known in this regard: [Ehrgott (2000)] gives some general results on approximating the efficient set by a single solution, [Prasad (1998)] uses a Tchebycheff metric to measure the error, and [Safer and Orlin (1995)] consider the existence of such algorithms.

Another aspect related to the difficulty of MOCO is the number of efficient solutions. It turns out that it may be exponential in the problem size, thus prohibiting any efficient method to determine all efficient solutions. Such results are known for the spanning tree, matroid base, shortest path, assignment, and travelling salesperson problem, see [Hamacher and Ruhe (1994] for details). Consequently such problems are called intractable. Even the size of the set SE may be exponential. However, numerical results available on the knapsack problem [Visée and al (1998)] show the number of supported solutions grows linearly with the problem size, but the number of nonsupported solution grows following an exponential function. As far as the other definitions of optimality in (MOCO) are concerned, we note that the max-ordering problem with sum objectives is *INP*-hard in general (see [Chung and al (1993)]), but can be reduced to a single objective problem in the case of bottleneck objectives [Ehrgott (1997)]. Bounds and heuristic methods for the former problem have been investigated in [Punnen and Aneja (1995)]. At least one solution of the max-ordering problem is always efficient, but possibly nonsupported. Similarly, a lexicographic max-ordering solution, although always efficient and optimal for the max-ordering problem may be nonsupported.

For lexicographic optimization it is known that a lexicographically optimal solution is always efficient, and even a supported efficient solution, see [Hamacher and Ruhe (1994)]. Lexicographic optimization can also be viewed as a special case of algebraic optimization.

In view of the new trend to apply metaheuristics and local search in MOCO problems, it is interesting to consider the issue of neighbourhoods of feasible solutions, and their relations to effcient solutions. Using a neighbourhood corresponding to Simplex basis pivots for the shortest path problem and exchanges of one edge for the spanning tree problem it was shown in [Ehrgott and Klamroth (1997)] that the set of efficient solutions can be an unconnected subset of X with respect to the neighbourhood. So it is possible that local search methods (in principle) cannot find all efficient solutions.

3- Applied techniques of combinatorial optimization

In this section, we briefly explain mathematical programming and simulated annealing

3-1 Mathematical programming

Let us first define a *linear program* (LP): Given an $m \times n$ Matrix A, an m-vector b, and an n-vector c, minimize $c^T \cdot x$ subject to $A \cdot x \geq b$, $x \geq 0$, with $x \in R^n$. Generally, an LP can be solved in polynomial time. Most commonly the *simplex algorithm* is applied. Although this theoretically may require exponential time, it solves LPs with hundreds of thousands of variables in practice [Haunert (2007)].

By replacing the continuous variables $x \in R^n$ in this definition by integer variables $x \in Z^n$, we define an *integer linear program* (ILP) or simply *integer program* (IP). Many combinatorial optimization problems can be formulated as IP. Though the definitions of IP and LP are very similar, the computational complexity of solving an IP is much higher. In fact the problem is NP-hard. However, several algorithms have been developed for the solution of IPs, which have been found out to be useful for applications. A *mixed-integer program* (MIP) is a combination of an LP and an IP, i.e., it may contain continuous as well as integer variables. Basically, a MIP can be solved with the same techniques as an IP.

3-2 Simulated annealing

The techniques described in the pervious section restrict to objectives and constraints that can be expressed by linear combinations of variables. Even in case that such a formulation of a problem is found, the branch-and-cut technique can turn out to be inefficient and therefore inappropriate for application. However, it is often not necessary to insist on finding the globally optimal result. Therefore, *heuristic* techniques have been developed. Generally, these attempt to find relatively good solutions in reasonable time. Two different types of heuristics need to be distinguished: Heuristics that are designed for a specific problem and those that offer solutions for a very general class of problems (*meta-heuristics*).

To explain simulated annealing, let us first consider a *hill-climbing* method: Starting from a feasible solution, hill climbing iteratively moves to a solution which is cheaper according to a cost function *c*, e.g. it selects the best solution in a defined neighborhood of the current solution. The problem with the hill-climbing approach is that it usually gets stuck in local optima. The simulated annealing approach is to occasionally accept moves to worse solutions, in order to escape these local optima. For this, a *temperature T* is introduced, which controls the probability of accepting worse solutions. Initially, *T* is high, meaning that it is likely that worse solutions are accepted. During the simulation *T* is decreased according to a defined *annealing schedule*. Commonly, a multiplier $\alpha \in [0, 1]$ is introduced for this. The following algorithm defines the common simulated annealing approach [Haunert (2007)]:

1. Find an initial feasible solution *s* and define the temperature by $T \leftarrow T_0$.
2. Randomly select a solution s' in the neighborhood of *s*.
3. If $c(s') \leq c(s)$, set $s' \leftarrow s$, else, set $s' \leftarrow s$ with probability $\exp\left(-\frac{c(s') - c(s)}{T}\right)$.
4. Reduce the temperature, i.e., set $T \leftarrow \alpha \cdot T$.
5. Proceed with 2 until the temperature falls below a threshold T_E.

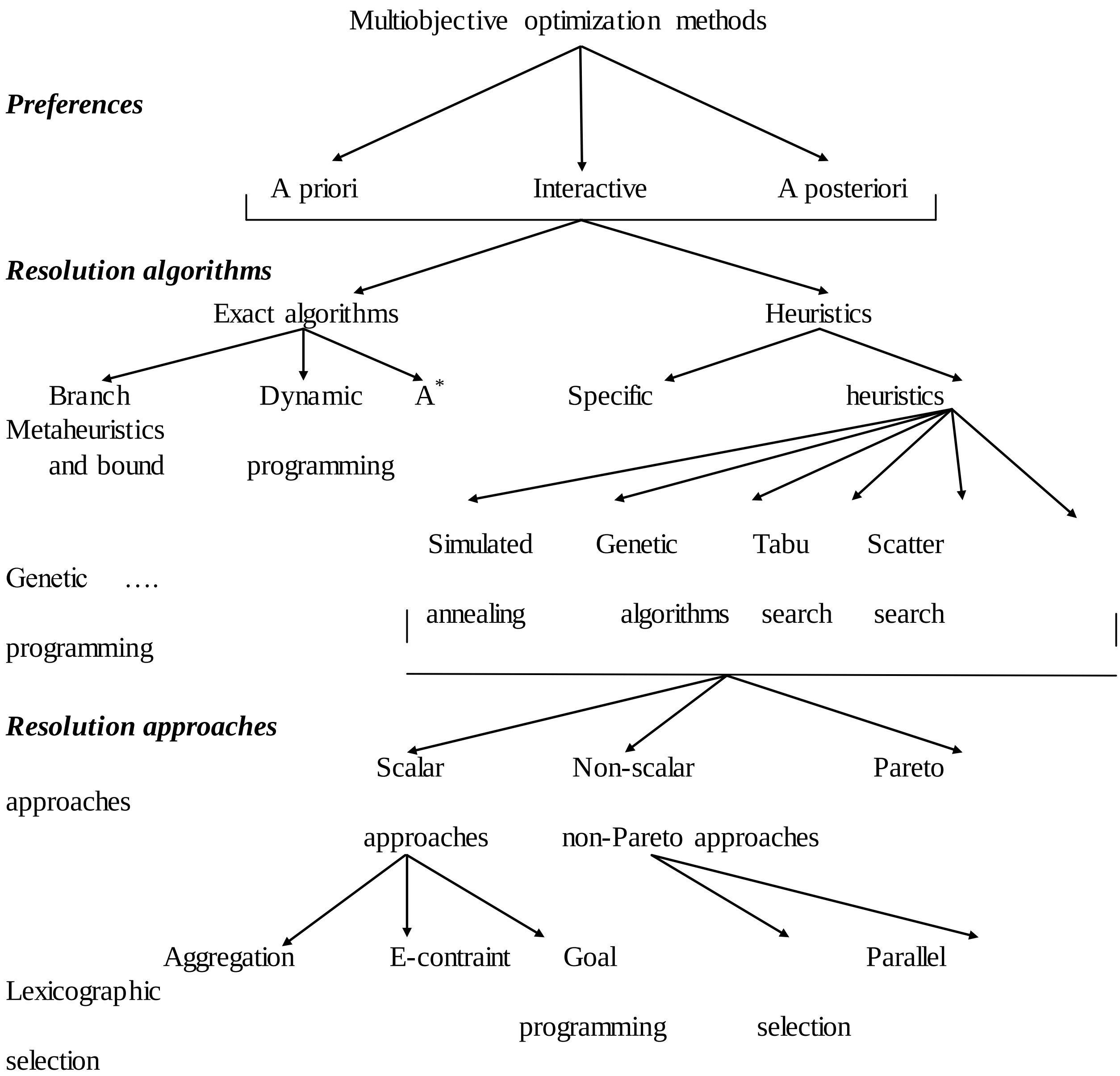
4- Classification of multi-objective combinatorial optimization methods

The approaches used for MCOPs resolution can be classified in three main categories [Basseur and al (2006)]

- Scalar approches : these methods imply the transformation of MCOP inti a single-objective problem. This class of approches includes those alghorithms based on aggregation, wich combine the various cost functions f_i into only one objective function F. These techniques require for the decision maker to have a good knowledge of its problem.
- Pareto approches : they are based on directly using the concept of Pareto optimality in their search. The process of selection of the generated solutions is based on the concept of non-dominance.

- Non-Pareto and non-scalar approaches : these approaches do not transform the MCOP into a single-objective problem ; on the contrary, they use operators to treat the various objectives separately.

Figure 1 : Classification of multi-objective combinatorial optimization methods.



Conclusion

Multi-Objective Combinatorial Optimization (MOCO) explores a finite search space of feasible solutions and finds the optimal ones that balance multiple (often conicting) objectives simultaneously. MOCO is a fundamental challenge in many design and development problems in engineering, in economic sciences and other domains.

Combinatorial optimization problems are often too complex to be solved within reasonable time limits by exact methods, in spite of the theoretical guarantee that such methods will ultimately obtain an optimal solution. Instead, heuristic methods, which do not offer a convergence guarantee, but which have greater flexibility to take advantage of special properties of the search space, are commonly a preferred alternative. The standard procedure is to craft a heuristic method to suit the particular characteristics of the problem at hand, exploiting to the extent possible the structure available. Such tailored methods, however, typically have limited usefulness in other problems domains.

An alternative to this problem specific solution approach is a more general methodology that recasts a given problem into a common modeling format, permitting solutions to be derived by a common, rather than tailor-made, heuristic method. Because such general purpose heuristic approaches forego the opportunity to capitalize on domain-specific knowledge, they are characteristically unable to provide the effectiveness or efficiency of special purpose approaches. Indeed, they are typically regarded to have little value except for dealing with small or simple problems.

References:

- **Bazgan C., Hugot H. and Vanderpooten D. (2009)**, "Solving efficiently the 0–1 multi-objective knapsack problem", *Computers & Operations Research* 36.1, pp. 260–279.
- **Chung S., Hamacher H.W., Maffioli F., and Murty K.G. (1993)**, "Note on combinatorial optimization with max-linear objective functions", Discrete Applied Mathematics 42, p.p139-145.
- **Clímaco J. and Pascoal M. (2012)**, "Multicriteria path and tree problems: discussion on exact algorithms and applications", *International Transactions in Operational Research* 19.1-2, pp. 63–98.
- **Climaco J., Ferreira C. and Captivo M.E. (1997)**, "Multicriteria integer programming: An overview of the different algorithmic approaches", Multicriteria Analysis, Springer-Verlag, Berlin, p.p 248-258.
- **Ehrgott M. and Gandibleux X. (2000),** " A survey and annotated bibliography of multiobjective combinatorial optimization", OR Spektrum 22, pp. 425–460.
- **Ehrgott M. (1998)**, " Discrete decision problems, multiple criteria optimization classes andlexicographic max-ordering", In T.J. Stewart and R.C. van den Honert, editors, Trends in Multicriteria Decision Making, volume 465 of Lecture Notes in Economics and Mathematical Systems, Springer Verlag, Berlin, p.p 31-44.
- **Ehrgott M. (2008)**, "*Multiple Objective Programming and Goal Programming: Theoretical Results and Practical Applications*", Lecture Notes in Economics and Mathematical Systems, France: Springer 618 , pp. 267–282.
- **Ehrgott M. and Gandibleux X. (2007)**, "Bound sets for multi-objective combinatorial optimization problems", *Computers and Operations Research* 34.9, pp. 2674–2694.
- **Ehrgott M. and Klamroth K. (1997),** "Connectedness of eÆcient solutions in multiple criteria combinatorial optimization", European Journal of Operational Research 97, p.p 159-166.
- **Ehrgott M., Nickel S., and Hamacher H.W. (1999)**, "Geometric methods to solve maxordering location problems", Discrete Applied Mathematics 93, p.p3-20.
- **Emelichev V.A. and Perepelitsa V.A. (1991),** "Complexity of vector optimization problems on graphs", Optimization 22, p.p 903-918.
- **Hamacher H.W. and Ruhe G. (1994),** "On spanning tree problems with multiple objectives", Annals of Operations Research 52, p.p 209-230.
- **Haunert J-H (2007),** "Optimization methods for area aggregation in land cover maps", In *Proceedings of the 23rd International Cartographic Conference (ICC'07)*.
- **Kouvelis P. and Carlson R.C. (1992),** "Total unimodularity applications in bi-objective discrete optimization", Operations Research Letters 11, p.p 61-65.
- **Matthieu B., El-Ghazali T. and Antonio N., Enrique A. (2006)**, "Metaheuristics for multiobjective combinatorial optimization problems : Review and recent issues", Laboratoire d'Informatique Fondamentale de Lille, INRIA Lille - Nord Europe, p.p 9-10.

- **Prasad S.Y. (1998),** "Approximation error analysis in bicriteria heuristics", Journal of Multi-Criteria Decision Analysis 7(3), p.p155-159.
- **Przybylski A., Gandibleux X., and Ehrgott M. (2010)**, "A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives", *Discrete Optimization* 7.3, pp. 149–165.
- **Punnen A.P. and Aneja Y.P. (1995),** "Minmax combinatorial optimization", European Journal of Operational Research 81, p.p 634-643.
- **Rasmussen L.M. (1986)**, "Zero-one programming with multiple criteria", European Journal of Operational Research 26, p.p 83-95.
- **Safer H.M. and Orlin J.B. (1995)**, "Fast approximation schemes for multi-criteria combinatorial optimization", Technical report, Sloan School of Management, MIT, Cambridge.
- **Sylva J. and Crema A. (2004)**, "A method for finding the set of non-dominated vectors for multiple objective integer linear programs", *European Journal of Operational Research* 158.1, pp. 46–55.
- **Teghem J. and Kunsch P.L. (1986),** "Interactive methods for multi-objective integer linear programs", Largescale modelling and interactive decision analysis, volume 273 of Lecture Notes in Economics and Mathematical Systems, Springer Verlag, Berlin, p.p 75-87.
- **Teghem J. and Kunsch P.L. (1986)**, "A survey of techniques for finding efficient solutions to multi-objective integer linear programming", Asia-Pacific Journal of Operational Research 3, p.p 95-108.
- **Ulungu E.L. and Teghem J. (1995)**, "The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems", *Foundations of Computing and Decision Sciences* 20.2, pp. 149–165.
- **Visée M., Teghem J., Pirlot M., and Ulungu E.L. (1998),** "Two-phases method and branch and bound procedures to solve the bi-obective knapsack problem", Journal of Global Optimization 12, p.p 139-155.