Pareto Simulated Annealing

Piotr Czyżak[1] and Andrzej Jaszkiewicz[1]

[1] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland

Abstract. The paper presents a multiple objective metaheuristic procedure - Pareto Simulated Annealing. The goal of the procedure is to find in a relatively short time a "good" approximation of the set of efficient solutions of a multiple objective combinatorial optimization problem. The procedure uses a sample of generating solutions. Each of the solutions explores its neighborhood in a way similar to that of classical simulated annealing. Weights of the objectives are set in each iteration in order to assure a tendency to approach the efficient solutions set while maintaining a uniform distribution of the generating solutions over this set.
Keywords. Multiple objective combinatorial optimization, metaheuristic procedures.

1. Introduction

Many real-life problems are combinatorial, i.e. they concern a choice of the best solution from a finite but large set of feasible solutions [18]. It is also a well known fact that the solutions of real-life problems are often evaluated from several points of view which may be described by different objectives [20], [26]. These facts were reasons for significant research efforts in the fields of *combinatorial optimization* (CO) and *multiple objective decision making* (MODM). The research resulted in many practical applications of methods developed in each of the fields. Nevertheless, surprisingly few theoretical works concern *multiple objective combinatorial optimization* (MOCO) problems [27]. Some exceptions are *multiple objective shortest path* problems [5] and *multiple objective project scheduling* problems [22], [24]. As a consequence, very few practical applications of MODM methods for combinatorial problems are reported in the OR literature [28].

In opinion of the authors, the relatively small number of applications of MOCO is not due to the fact that combinatorial problems rarely require multiple objectives but due to the notable difficulty of such problems. Indeed, different objectives are often used in particular classes of combinatorial problems. For example, in vehicle routing the typically used objectives are total cost, distance, number of vehicles, travel time [1], and in project scheduling the typical objectives are net present value, project completion time, mean weighted delay,

G. Fandel et al. (eds.), *Multiple Criteria Decision Making*

number of delayed tasks, mean weighted flow time, total resource utilisation [22, [24]. The objectives, however, are usually used separately or they are combined into a single objective.

It is the authors belief that the MOCO problems create one of the most difficult classes in the field of OR. This difficulty results from the following two factors:

- solving a MOCO problem requires intensive co-operation with the decision maker (DM); this results in especially high requirements for robust tools used to generate efficient solutions,
- many combinatorial problems are hard even in single objective versions; their multiple objective versions are frequently more difficult.

A MODM problem is ill-posed from mathematical point of view, because, except of trivial cases, it has no unique solution. It is usually assumed that the DM should select one of the *efficient* solutions [19]. The goal of MODM methods is to find an efficient solution most consistent with the DM's preferences, i.e. the *best compromise*. Interactive procedures, which became especially popular in recent years, generate efficient solutions in computational phases alternating with phases of decision. Such procedures, however, can only be used if sufficiently robust tools for finding efficient solutions are available.

Furthermore, many of single objective combinatorial problems belong to the class of NP-hard problems. Generation of efficient solutions in a MOCO problem is, of course, not easier than finding solutions optimizing particular objectives and, in many cases, is even harder. For example, the single objective shortest path problem is one of the simplest combinatorial problems while corresponding multiple objective problem is NP-hard [8]. So, even problems for which relatively efficient single objective exact methods are known, may be difficult if multiple objectives have to be considered.

Tools used for generation of efficient solutions in MOCO, alike single objective optimization methods, may be classified into one of the following categories:

- exact procedures,
- specialized heuristic procedures,
- metaheuristic procedures.

The main disadvantage of exact algorithms consists in their high computational complexity. In result, only limited class of real-life MOCO problems may be solved exactly.

The main weak point of specialized procedures, both heuristic and exact, is their inflexibility. This factor seems to be especially important in the case of MOCO. For example, many MODM methods start by presenting to the DM the ideal point, which is found by independent optimization of particular objectives. If the objectives have different mathematical form, their optimization may require different specialized procedures. Furthermore, MODM methods use various tools for generating efficient solutions. Some of them apply the Geoffrion's theorem (see e.g. [26]), others use penalty functions [3], [13], [17], utility functions [14], ε -Constraints (see e.g. [26]), or achievement scalarizing functions [28]. Of course, each of the tools, may require individual specialized procedure. In result, solving a given MOCO problem may involve several specialized procedures and it may be difficult to change the formulation of the model, the family of objectives or even the MODM method used to solve the problem.

Consequently, this is not only opinion of the authors (compare e.g. [27]) that the most promising practical approach to MOCO consists in generating efficient solutions with metaheuristic procedures.

In recent years several metaheuristic methods for single objective combinatorial problems were proposed. They allow finding nearly optimal solutions for a wide class of combinatorial problems in a relatively short time. The methods have been successfully applied to many problems that cannot be solved exactly in the polynomial time. Single objective metaheuristics, however, might not be robust enough for generation of efficient solutions, especially, if interactive exploration of efficient solutions set is used.

The paper proposes a metaheuristic procedure, called Pareto Simulated Annealing (PSA), for MOCO problems. The goal of the procedure is to find a set of solutions being a "good" approximation of the efficient solutions set. The set of solutions obtained by the procedure may then be presented to the DM. If the set is small enough, the DM may select the best compromise from it. Otherwise, the DM may explore the set of solutions guided by an interactive procedure.

The paper is organized in the following way. In the next section a formal statement of the MOCO problem is presented. In the third section classical single objective metaheuristic procedures are considered as tools for generation of efficient solutions. Some previous proposals of multiple objective metaheuristic procedures and their weaknesses are described in the fourth section. In the fifth section the PSA procedure is presented. Evaluation of multiple objective metaheuristic procedures is discussed in the sixth section. Finally, main features of the procedure are summarized and some possible directions of further research are described in the seventh section.

2. Problem statement

The general MOCO problem is formulated as:

$$\max\{f_1(\mathbf{x}) = z_1, \ldots, f_J(\mathbf{x}) = z_J\}$$

s.t.

$$\mathbf{x} \in D,$$

where: *solution* $\mathbf{x} = [x_1, \ldots, x_I]$ is a vector of discrete *decision variables*, D is the set of feasible solutions.

Solution $\mathbf{x} \in D$ is *efficient* (*Pareto-optimal*) if there is no $\mathbf{x}' \in D$ such that $\forall_j f_j(\mathbf{x}') \geq f_j(\mathbf{x})$ and $f_j(\mathbf{x}') > f_j(\mathbf{x})$ for at least one j. The set of all efficient solutions is denoted by N.

3. The use of single objective metaheuristic procedures for generation of efficient solutions

Single objective metaheuristic procedures were proposed and became popular relatively recently. They include such methods as: simulated annealing (SA) [2] [15], [16], tabu search [9] and genetic algorithms [10]. The methods are called metaheuristics, because they define only a "skeleton" of the optimization procedure that have to be customized for particular applications. The popularity of the methods increases due to the following advantages:

- generality, i.e. they may be used for solving various classes of problems,
- robustness, i.e. they allow to find solutions close enough to the optimal one in acceptable time for many real-life problems (please note, however, that the efficiency strongly depends on the way of customization for a given application),
- flexibility, i.e. they are relatively insensitive to changes in the problems formulation,
- simplicity, i.e. they can be relatively easily implemented.

The above advantages make the procedures especially useful for OR practitioners.

As most of the MODM methods use single objective optimization as a tool for generating efficient solutions it seems rational to apply the classical single objective metaheuristic procedures in MOCO. Yet such procedures may appear too inefficient for practical applications. As was mentioned in the first section, finding the best compromise requires a co-operation with the DM and gathering preference information from him/her. Taking into account the moment of collecting preference information with respect to computational process, MODM methods can be classified into [12], [23]:

- methods with a priori articulation of preferences,
- interactive methods, in which preferences are expressed progressively during the search over the non-dominated set,
- methods with a posteriori articulation of preferences.

Methods from the first class start by collecting the preference information from the DM, which is then used to build a model of his preferences. The model is then exploited in order to find the best compromise. In many cases the preference model takes the form of a real-valued function, e.g. utility function [14], penalty function [3], [13], [17] or scalarizing function [28]. Exploitation of such a model consists in optimizing the function on the feasible set. So, single objective metaheuristic procedures can be used in this step.

Methods with a priori articulation of preferences may, however, be only used if the preferences of the DM are well established at the beginning of the solution process. Yet even in this case the DM may be unable or unwilling to deliver all the information required to build the model of his preferences. Furthermore, functional models do not allow incomparability of some solutions which is often observed in practice [20]. Some comparative experiments also indicate that methods of this class perform relatively poor in the case of multiple objective mathematical programming [4].

Interactive methods consist of computational phases alternating with phases of decision. In each computational phase a solution or a sample of solutions, usually efficient, is generated. Single objective optimization methods are usually used as tools for generating these solutions. This allows for using single objective metaheuristic procedures in the case of interactive procedures applied to MOCO problems.

Interactive methods, however, may only be used if the time needed for the computational phase is acceptable for the DM. Although, metaheuristic procedures are relatively robust tools for combinatorial optimization, they might not be robust enough for the use in interactive procedures, especially in the case of methods generating samples of solutions.

Methods with a posteriori articulation of preferences are often criticized for their high computational complexity. This is clearly the case if the whole set of efficient solutions is to be generated. To this class, however, belong also methods finding approximations of the set of efficient solutions N. This approach seems to be especially interesting in the case of MOCO problems, because of mentioned above prohibitive computational complexity of exact methods. Please note, however, that finding the whole efficient solutions set or its approximation does not always completes the solution procedure. This set may be too large for the DM to analyze it and to select the best compromise without further support. So, the DM may decide to use one of procedures for interactive exploration of a finite but large set of alternatives.

4. Review of existing multiple objective metaheuristic procedures

Several multiple objective metaheuristic procedures have already been proposed. The goal of the procedures is to find a sample of feasible solutions being a "good" approximation of the efficient solutions set. Such procedures may be used in the first phase of methods with a posteriori articulation of preferences or in computational phases of interactive procedures generating samples of solutions.

Schaffer [21] (see also [10]) and Srinivas and Kalyanmoy [25] proposed methods based on genetic algorithms Although, genetic algorithms may be used in the case of combinatorial problems the methods are designed for continuous case. Usefulness of these procedures in the case of MOCO is yet to be tested.

Fortemps, Teghem and Ulungu [7] have proposed an algorithm based on simulated annealing and suggested their use in MOCO. The algorithms of their proposals are very close to that of the single objective SA. The general scheme of the procedures is given below:

Select a starting solution $\mathbf{x} \in D$
Update set M of potentially efficient solutions with $\mathbf{x}$
$T := T_0$
<u>**repeat**</u>
    Construct $\mathbf{y} \in V(\mathbf{x})$
    Update set M of potentially efficient solutions with $\mathbf{y}$
    $\mathbf{x} := \mathbf{y}$ (accept $\mathbf{y}$) with probability $P(\mathbf{x}, \mathbf{y}, T, \Lambda)$

if the conditions of changing the temperature are fulfilled **then**
decrease T

until the stop conditions are fulfilled

where: $V(\mathbf{x}) \subseteq D$ is the neighborhood of solution **x**, i.e. the set of feasible solutions that may be reached from **x** by making a simple move, T - temperature, $\Lambda = [\lambda_1, \ldots, \lambda_j]$ is a vector of weights.

Unlike single objective SA these procedures result not in a single solution but in a set of *potentially efficient solutions*, i.e. the set composed of solutions efficient with respect to all generated solutions. The other difference is in the way of calculating the probability of accepting a new solution, denoted by $P(\mathbf{x}, \mathbf{y}, T, \Lambda)$. In the case of single objective SA a new solution is accepted with probability equal to one if it is not worse than the current solution. Otherwise, it is accepted with probability less than one. In the case of multiple objectives one of the following three situations may appear while comparing a new solution **y** with the current one **x**:

- **y** may dominate (weakly dominate) **x**,
- **y** may be dominated by **x**,
- **y** may be nondominated with respect to **x**.

In the first situation the new solution may be considered as not worse than the current one and accepted with probability equal to one. In the second situation the new solution may be considered as worse than the current one and accepted with probability less than one. Fortemps, Teghem and Ulungu [7] have proposed several multiple objective rules for acceptance probability which in different way treat the third situation. Some characteristic rules are described below.

Rule C may be seen as a local aggregation of all objectives with an achievement scalarizing function based on the Chebyshev metric with the reference point **x**. It is defined by the following expression:

$$P(\mathbf{x}, \mathbf{y}, T, \Lambda) = \min\left\{1, \exp\left(\max_j \left\{\lambda_j \left(f_j(\mathbf{x}) - f_j(\mathbf{y})\right) / T\right\}\right)\right\}.$$

Rule SL may be seen as a local aggregation of all objectives with a weighted sum of the objectives. It is defined by the following expression:

$$P(\mathbf{x}, \mathbf{y}, T, \Lambda) = \min\left\{1, \exp\left(\sum_{j=1}^{J} \lambda_j \left(f_j(\mathbf{x})_j - f_j(\mathbf{y}) / T\right)\right)\right\}.$$

Rule W is defined by the following expression:

$$P(\mathbf{x}, \mathbf{y}, T, \Lambda) = \min\left\{1, \exp\left(\min_j \left\{\lambda_j \left(f_j(\mathbf{x}) - f_j(\mathbf{y})\right) / T\right\}\right)\right\}.$$

Please note, that the higher is the weight associated with a given objective the lower is the probability of accepting moves that decrease the value on this objective and the greater is the probability of improving value on this objective. So, controlling the weights one can increase or decrease the probability of improving values of the particular objectives.

The authors have tested the above approach on some MOCO problems. It was observed that the procedure works good enough for relatively small problems only. In the case of larger problems the set of potentially efficient solutions may represent a small region of set N. This was the reason for developing the PSA procedure which tends to generate a "good" approximation of the whole set N even for relatively large MOCO problems.

5. Description of the Pareto Simulated Annealing

PSA uses some ideas known from two existing single objective metaheuristic procedures: genetic algorithms [10] and simulated annealing [15], [16]. These ideas and their origin may be summarized as follows:

- genetic algorithms
  - The use of a sample (population) of interacting solutions
- simulated annealing
  - The exploration of a neighborhood of the considered solution
  - The acceptance of a new solution with some probability depending on a parameter called the temperature
  - The scheme of decreasing the temperature

The general scheme of the PSA procedure may be summarized as follows:

```
Select a starting sample of generating solutions S ⊂ D
Update set M of potentially efficient solutions with S
T := T_0
repeat
    for each x ∈ S do
        Construct y ∈ V(x)
        Update set M with y
        Select solution x' ∈ S closest to x and nondominated
        with respect to x
        if there is no such solution x' or it is the first iteration
        with x then
            Set random weights such that:
            ∀_j λ_j ≥ 0 i Σ_j λ_j = 1
        else
            for each objective f_j
                λ_j = { α λ_j^x   , if f_j(x) ≥ f_j(y)
                      { λ_j^x / α , if f_j(x) < f_j(y)
        x := y (accept y) with probability P(x, y, T, Λ)
    if the conditions of changing the temperature are fulfilled then
        decrease T
until the stop conditions are fulfilled
```

where: $\Lambda^{\mathbf{x}} = \left[\lambda_1^{\mathbf{x}}, \ldots, \lambda_J^{\mathbf{x}}\right]$ is the weighting vector used in the previous iteration for solution **x**, $\alpha > 1$ is a constant close to one (e.g. $\alpha = 1.05$), $P(\mathbf{x}, \mathbf{y}, T, \Lambda)$ is one of the multiple objective rules for acceptance probability described above.

In each iteration of the procedure a sample of solutions, called generating sample, is used. The main idea of PSA is to assure a tendency for approaching the set of efficient solutions as well as an inclination for dispersing the solution constituting the generating sample over the whole set N. In result each solution tends to investigate a specific region of set N.

The tendency for approaching the set of efficient solutions is assured by using one of the mentioned above multiple objective rules for acceptance probability. The inclination for dispersing the solutions from the generating sample over the whole set N is obtained by controlling the weights of particular objectives used in these rules. For a given solution $\mathbf{x} \in S$ the weights are changed in order to increase the probability of moving it away from its closest neighbor in S denoted by **x'**. This is obtained by increasing weights of the objectives on which **x** is better than **x'** and decreasing weights of the objectives on which **x** is worse than **x'**.

Please note that the algorithm of PSA is essentially parallel because calculations required for each point from S, i.e. construction of a new solution form its neighborhood, setting the weights and accepting the new solution, may be done on different processors.

It is also worth mentioning that one of the crucial points of the procedure from the point of view of its effectiveness is updating of the set of potentially efficient solutions. A data structure called Quad Tree allows for very effective implementation of this step [6], [11].

PSA is not a complete method for solving MOCO problems but just a tool for generating approximation of the set of efficient solutions. It is proposed to use the following three phase method for finding the best compromise of a MOCO problem:

Phase I. Finding an approximation M of set N with PSA
Phase 2. Selection of the best solution $\overline{\mathbf{x}}$ from the set M
Phase 3. Searching for a solution $\hat{\mathbf{x}}$ dominating $\overline{\mathbf{x}}$

The second phase is the only one that requires co-operation with the DM. In this phase the DM learns of the possible trade-offs as well as of his/her preferences and selects the best compromise. As was mentioned before, he/she may require some support in this phase. One of the interactive methods for multiple criteria problems with a large but finite set of alternatives may be used to support the DM.

In the third phase solution $\overline{\mathbf{x}}$ may be used as a reference point of a scalarizing function. This function can be optimized with a single objective metaheuristic procedure. This phase may result in a solution $\hat{\mathbf{x}}$ dominating $\overline{\mathbf{x}}$. So, this phase consists in an attempt at improving $\overline{\mathbf{x}}$ with respect to all objectives.

6. Evaluation of multiple objective metaheuristic procedures

Since the goal of multiple objective metaheuristic procedures is to find a "good" approximation of the set N, it is important to have some evaluation technique allowing for comparison of different approximations.

Solution obtained by a single objective optimization method may be evaluated by comparison with another solution obtained in a different way or with some reference solution, e.g. global optimum or the best solution known so far. Analogously in the multiple objective case a set of potentially efficient solutions may be compared with another such set obtained in a different way or compared with some reference set R of solutions, e.g. the set of efficient solutions or the best approximation known so far. The quality metric described below concern the latter case.

The proposed quality metric is defined as:

$$Dist = \max_{\mathbf{y}\in R}\left\{\min_{\mathbf{x}\in M}\left\{\max_{j=1,\dots,J}\left\{0, w_j\left(f_j(\mathbf{y}) - f_j(\mathbf{x})\right)\right\}\right\}\right\}.$$

In other words, for each solution $\mathbf{y}$ from reference set R the closest solution from set M is found. The distance between two solutions is measured by the scalarizing function based on the weighted Chebyshev metric with the reference point $\mathbf{y}$. So, the closest solution to $\mathbf{y}$ is the one that minimizes the maximal weighted deviation from $\mathbf{y}$ on each objective. The weights used in the above expression are first set as:

$$w_j = \frac{1}{\Delta_j}$$

where: Δ_j is range of objective f_j in the reference set, and then normalized such that:

$$\sum_{j=1}^{J} w_j = 1.$$

Some results of computational experiments with the procedure will be described during the conference presentation. Application of the procedure to the multiple objective cell formation problem is presented in another paper presented at the Conference.

7. Summary and conclusions

A multiple objective metaheuristic procedure for combinatorial problems has been presented. The procedure tends to generate a "good" approximation of the efficient solutions set in a relatively short time. The main advantages of the procedure are as follows:

- it finds representation of the whole set of efficient solutions for relatively large problems,
- it naturally allows for a parallel implementation.

The following directions of further research may be considered:

- adaptive setting of the size of he generating sample which is a new parameter introduced in PSA,
- the use of concepts from other single objective metaheuristic procedures, e.g. tabu search.

Acknowledgment

This research was supported by KBN grant no. 8 S503 016 016.

Bibliography

[1] Assad, A.A.: "Modelling and implementation issues in vehicle routing". In: B.L. Golden, A.A. Assad (eds.): *Vehicle routing: Methods and Studies*, North-Holland, Amsterdam 1988, pp. 7-46.

[2] Cerny V.: A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm, Report, Comenius University, Bratislava, Czechoslovakia 1982.

[3] Charnes A. and Cooper W.W. (1977), Goal programming and multiple objective optimizations, part 1. Eur. J. Opl Res. 1 (1977).

[4] Corner J.L., J.T. Buchanan: Experimental consideration of references in decision making under certainty: further evidence. XIth International Conference on Multiple Criteria Decision Making, Coimbra, Portugal, 1-6.08.1994, Coimbra 1994.

[5] Current J., M. Marsh: Multiobjective transportation network design and routing problems: taxonomy and annotation. Eur. J. Opl Res. 65 (1993) 1-15.

[6] Finkel R.A., J.L. Bentley: Quad Trees, A data structure for retrieval on composite keys. Acta Informatica 4 (1974) 1-9.

[7] Fortemps P., J. Teghem, B. Ulungu: Heuristics for multiobjective combinatorial optimization by simulated annealing, XI-th International Conference on Multiple Criteria Decision Making, Coimbra, Portugal, 1-6.08.1994, Coimbra 1994.

[8] Garey M., D. Johnson: *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, San Francisco 1979.

[9] Glover F.: Tabu Search - Part I, ORSA J. Comput. 1 (1989) 190-206.

[10] Goldberg D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Massachuset etc. 1988.

[11] Habenicht W.: Quad Trees, A datastructure for discrete vector optimization problems. In: P Hansen (ed.): *Essays and Surveys on Multiple Criteria Decision Making*. Proc. of the Fifth International Conference on MCDM, Mons, Belgium, August 9-13, 1982, *Lecture Notes in Economics and Mathematical Systems*, **209**, Springer-Verlag, Berlin etc. 1982, pp. 136-145.

[12] Hwang C.-L., S.R. Paidy, K. Yoon, A.S.M. Masud: Mathematical programming with multiple objectives: A tutorial. Comput. Oper. Res. 7 (1980) 5-31.

[13] Ignizio J.P.: *Goal programming and Extensions.* Lexington Books, Massachusetts 1976.

[14] Keeney R.L., H. Raiffa: *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley & Sons, New York 1976.

[15] Kirkpatrick S., C.D. Gelatt, M.P. Vecchi: Optimization by simulated annealing, Science 220 (1983) 671-680.

[16] Laarhoven P.J.M., E.H.L. Aarts: *Simulated Annealing: Theory and Applications,* Reidel, Dordrecht 1987.

[17] Lee S.M.: *Goal programming for decision analysis.* Auerbach Publishers, Philadelphia, Pennsylvania 1972.

[18] Nemhauser D.L., L.H. Wolsey: *Integer and Combinatorial Optimization*, Wiley, Chichester 1988.

[19] Rosenthal R.E.: Principles of multiobjective optimization. Decision Sciences 16 (1985) 133-152.

[20] Roy B., D. Bouyssou: *Aide Multicritčre ŕ la Décision: Méthodes et Cas.* Economica, Paris 1993.

[21] Schaffer J.D.: *Some experiments in machine learning using vector evaluated genetic algorithms.* Doctoral dissertation, Vanderbilt University, Electrical Engineering, Tenessee 1984.

[22] Słowiński R.: Multiobjective network scheduling with efficient use of renewable and non renewable resources. Eur. J. Opl Res. 7 (1981) 265-273.

[23] Słowiński R.: Overview of multiple objective mathematical programming methods. Part I (In Polish). Przegląd Statystyczny XXXI (1984) 47-63.

[24] Słowiński R.: "Multiobjective project scheduling under multiple-category resource constraint". In: R. Słowiński, J. Węglarz (eds.): *Advances in Project Scheduling*, Elsevier, Amsterdam 1989, pp. 151-167.

[25] Srinivas N., K. Deb: Multiobjective optimization using nondominated sorting in genetic algorithms. Proceedings of IEEE World Congress on Computational Intelligence, 26.06-02.07.1994, Orlando, Orlando 1994.

[26] Steuer R.E.: *Multiple Criteria Optimization - Theory, Computation and Application*, Wiley, New York 1986.

[27] Ulungu E.L., J. Teghem: Multi-ojective Combinatorial Optimization Problems: A Survey. Journal of Multi-Criteria Decision Analysis 3 (1994) 83-101.

[28] White D.J.: A bibliography on the applications of mathematical programming multiple-objective methods. J. Opl Res. Soc. 41 (1990) 669-691.

[29] Wierzbicki A.P.: On the completeness and constructiveness of parametric characterization to vector optimization problems. OR Spectrum 8 (1986) 73-87.