Control and Cybernetics

vol. **29** (2000) No. 3

Experiments concerning hashing in the multiobjective tabu search method TAMOCO

by

Michael Pilegaard Hansen

Pilegaard Planning
Olof Palmes Gade 3, 5 tv., 2100 Copenhagen, Denmark
E-mail: mph@imm.dtu.dk

Abstract: This paper examines the use of hashing in the multiobjective tabu search, TAMOCO. The hashing method was suggested by Woodruff and Zemel (1993) as a method of avoiding return to the already examined solutions in the standard single-objective tabu search. While the traditional tabu list is capable of insuring this, it can normally only be used for cycles of a moderate length. The hashing method, however, can efficiently avoid cycles over a much larger number of iterations and must be considered a natural component in the tabu search tool-box. We report from two experiments on practical models where the hashing component has been included into the TAMOCO-procedure; with two different outcomes.

Keywords: tabu search; hashing; multiobjective combinatorial optimization (MOCO).

Introduction

As noted by Woodruff and Zemel (1993), the tabu-list of tabu search often serves two purposes, and we quote from their paper:

(1) *Avoidance of cycling — In order to escape a local minimum, the search must be prevented from "falling back" to a recently visited solution. Unless randomness is used in move selection, it is easy to see that if a solution can be revisited, the algorithm may cycle infinitely.*

(2) *Trajectory — By making certain move attributes tabu, an attribute list often prevents the "reversal" of moves. This results in exclusion of many solutions that have not yet been visited. In many instances this is desirable because it forces the search to explore new regions of X but the aspiration criterion precludes the avoidance of any excellent solutions.* (X being the

The above distinction is useful because the necessary length of the tabu list in order to accommodate these two consideration might be quite different. The resulting length of the tabu list is therefore a trade-off between these two consideration — and the third consideration that we in general want a short tabu-list in order to have a fast and aggressive search.

Woodruff and Zemel suggest to isolate the problem of cycling and deal with it by keeping a very long list of solutions which have been visited in the search, possibly storing all solutions. A return to an already visited solution is then regarded as tabu. It is easy to see that if we avoid the return to the already visited solutions, no cycling can occur.

Because the solutions for efficiency reasons must normally be coded using a non-bijective hashing function, collisions can occur when two different solutions are mapped into the same hashing function value. Therefore, it makes sense to apply aspiration criteria on the neighbors which are labeled as tabu by the hashing method. For a deeper discussion of hashing functions, the probability of collisions, etc., the reader is referred to Woodruff and Zemel (1993).

Note, however, that unless the neighborhood function in itself induces trajectories in the search space, the hashing method, used alone as a local search procedure, can not do any better than preventing the return to the already visited solutions. This may cause the search to "pseudo-cycle" the neighborhood of a local optima and lead to a very low diversity in the search and a poor performance. It may therefore be advantageous to use combinations of the standard tabu-search and the hashing component.

This combination is attempted in this paper on a multiobjective combinatorial optimization problem. The aim here is to generate an approximation to the non-dominated set of a practical decision problem already examined in the literature.

After this short introduction to the hashing method, Section 1 will briefly outline the most basic parts of the multiobjective tabu search, TAMOCO. Section 2 describes the two test problems. Section 3 describes the specific usage of the hashing method, the chosen method of measurement and presents the computational result of the two test problems. Final remarks conclude on the experiments and give directions for further investigations.

1. Multiobjective tabu search

The general multiobjective combinatorial optimization (MOCO) problem is often stated as:

$$\text{"maximize" } f(x)$$
$$\text{subject to } x \in S$$

where S is a discrete, finite set of feasible solutions to the problem and f is the K-dimensional objective function to be "maximized", $f(x) = \{f^1(x), f^2(x), \ldots$

The term maximization is written in quotation marks since there generally does not exist any single solution, which simultaneously provides the optimal value on all K objectives. Instead, we may seek to generate the set of efficient (or Pareto optimal) solutions to the MOCO problem, where each solution is not worse than any other solution of S on all objectives and better on at least one objective. The image of the efficient set in the objective space is called the non-dominated set, and it is in fact often this set that we are interested in generating an approximation to. Unfortunately, locating just one efficient or non-dominated solution is very often an NP-hard task (see Serafini, 1987) and we may in practice have to settle for an approximation to a non-dominated solution or to the non-dominated set.

Methods for constructing such approximations have been devised for a selection of particular problems (see e.g. a survey by Ulungu and Teghem, 1994) using a variety of techniques. Adaptations of general applicable heuristics (metaheuristics) to MOCO have also been suggested within the scope of simulated annealing (Serafini, 1992, Fortemps, Teghem and Ulungu, 1994, and Czyżak and Jaszkiewicz, 1996, 1998), of genetic algorithms (Schaffer, 1985, Horn and Nafpliotis, 1993, Fonseca and Fleming, 1993 and 1995, Horn, Nafpliotis and Goldberg, 1994, and Srinivas and Deb, 1995) and tabu search (Gandibleux, Mezdaoui and Freville, 1996, and Hansen, 1998).

The multiobjective tabu search, TAMOCO, as described by Hansen (2000) is a general framework for adapting tabu search (Glover, 1989, Glover and Laguna, 1997) to multiple objectives. The general aim is to generate an approximation to the entire non-dominated set, as opposed to e.g. the approach of Gandibleux, Mezdaoui and Freville (1996), which optimizes towards the ideal point. While the single-objective tabu search works with only one current solution, TAMOCO works with a set of current solutions which are optimized towards the non-dominated frontier while at the same time seeking to diversify over this frontier. Each current solution maintains its own tabu-list and the current solutions take turns in applying one neighborhood move according to the tabu search metaheuristic. Whenever a solution is found which improves the set of potentially non-dominated solutions (ND-set), the ND-set is updated with the objective function values of this solution. In the end, the ND-set will contain the approximation to the non-dominated set. For a more detailed description on TAMOCO and its extension, the reader is referred to Hansen (2000) or Hansen (1998) and, for a deeper introduction to multiobjective optimization, to Steuer (1986).

2. Problem formulation

2.1. Experimental model 1: A budgeting problem

Czyżak and Jaszkiewicz (1996) present the following MOCO problem: A comof 50 potential building sites has been located and each site has been evaluated for its potential contribution on 3 criteria: short term profit, long term profit and negative environmental impact. Each site will require a given investment cost and the company posseses a limited budget for these investments. The problem is then to select a subset from the 50 potential sites which satisfies the budget constraint and optimizes the 3 objectives. This budgeting problem is equivalent to a multiobjective knapsack problem. Czyżak and Jaszkiewicz used the Pareto simulating annealing method to generate an approximation to the non-dominated set of solutions, one of which the company eventually selects (by using in this case the Light Beam Search as decision support system).

A neighborhood function was established in the following way: From a feasible solution, remove randomly selected sites until there is free capital for including the most expensive, non-selected site. Then randomly insert non-selected sites in the solution until there is no free capital for even the least expensive non-selected site. In the tabu search, sites which have been removed are declared tabu and cannot be inserted until they are no longer tabu.

This same problem and neighborhood function were used in Hansen (2000) and are used in the experiments of this paper as well. In order for the tabu-search to locate a best neighborhood solution, the neighborhood is sampled at random. By varying the sample size, we vary the "steepness" attached to each move.

2.2. Experimental model 2: A power network design problem

Another MOCO problem appears in Matos and Ponce de Leão (1995). Here the design of a power network is considered, distributing electricity from a set of suppliers to a set of consumers via a network. The network consists of arcs which are already constructed or can be built prior to the time-period in which it will be first used. Each arc has a maximum capacity, a probability of failure and a resistance; the latter results in a voltage drop as well as a power loss (heat). Furthermore, a radial constraint is imposed on the network design in each time period (meaning that no cycles may occur in the active arcs of the network), and a maximum voltage drop to the consumers must be respected. We consider a real, but relatively small case consisting of 21 arcs, 13 nodes and 3 time periods.

To handle uncertainty, the consumption of each consumer as well as the production of each producer is forecast as a fuzzy set for each time period in the planning horizon. This results in fuzzy flows in the network, and a new interpretation must be given to some of the model constraints. In fact, the constraints involving fuzzy sets are replaced with two objectives describing robustness (the maximum possibility that no constraint will be violated) and severity (how bad is the constraint violation). Three other objectives are formulated, namely the investment costs (building of new arcs), the operation costs (due to power loss) functions and it has been chosen to remove the fuzziness as a mean of comparing the outcomes. Also, investment costs and operation costs are financially depreciated. Together with the two objectives expressing the fuzzy constraints, a solution is therefore evaluated using 5 scalar objectives.

As a neighborhood function, the following approach is used. A feasible solution will present a spanning tree in each time period due to the radial constraint. In each of the time periods, all non-used arcs are considered to be inserted into the tree, inducing a cycle, which again is broken by removing one of the arcs in the cycle. Which arc to remove is determined using a priority rule, giving tabu arcs (of that time period) lower priority, and, in case of using also hashing tabu, giving even lower priority if the resulting solution has already been evaluated. The solution with highest priority is then evaluated and the best solution is kept. When all inserted arcs and all time-periods have been evaluated, a best neighbor is determined, the move is made and the combination of time-period and inserted arc is made tabu so that the inserted arc cannot be removed in that time-period until it is no longer tabu. For more details on the specific model, the modeling of fuzzy sets and on the tabu search implementation, please refer to Matos and Ponce de Leão (1995) and Hansen (1997).

3. Experimental results

Experiments have been set up with the purpose of detecting whether or not the hashing component offers an improved performance of the TAMOCO implementation for each of the two experimental models. All computational experiments have therefore been conducted both with and without the hashing component. Before presenting the experimental results, the implementation of the hashing component as well as the chosen method of evaluating an approximation are described.

3.1. Implementation of the hashing component

While the TAMOCO procedure works with a set of current solutions, each carrying their own tabu list, we will for the hashing tabu component use a shared list which is valid for all of the current solutions. This is natural since we do not want a current solution to move into a solution, which has been visited already by another current solution.

In the examples of this paper we will consider a binary decision vector, x, with N elements, $x = (x_1, x_2, \dots x_N) \in X = [0;1]^N$, and we use the h_1 hashing function:

$$h_1(x) = h_0(x) \bmod [MAXINT + 1]$$

$$h_0(x) = \lceil \sum^{N} z_i x_i \rceil$$

Using 32-bit unsigned integers, the $MAXINT$ takes on the value $2^{32} - 1$. The values of z_i are selected at random from the set $\{1, 2, 3, 4, 5, \ldots, MAXINT\}$. The h_0 hashing function values are computed using long integers.

For many neighborhood functions, with y being the neighbor solution to x, $h_0(y)$ is quickly calculated as:

$$h_0(y) = h_0(x) + \sum_{i \in I} z_i - \sum_{i \in O} z_i,$$

where I is the set of incoming variables (indices, where x_i changes from 0 to 1) and O is the set of outgoing variables (changes from 1 to 0). The sets I and O will with many neighborhood function be very small sets. If this is not the case, other hashing-functions should be considered.

The h_1 hashing function value of a new current solution is inserted into a binary tree. However, we limit the number of elements in the binary tree in order to limit the memory consumption and to reduce the effect of collision. This is efficiently implemented in a cyclic list by keeping a thread through the tree from the oldest to the newest element. Insertion of a new element with removal of the oldest element can then still be performed within the complexity of $O(\log(M))$, where M is the number of elements in the tree. Finally, collisions will only result in temporary tabu.

Woodruff and Zemel (1993) also emphasize the usage of aspiration criteria in the hashing tabu method. In the single-objective case, a natural aspiration criterion can be to accept a new best solution, even if tabu. Directly converting this to the multiobjective context is to allow new tabu solutions whenever they contribute to the non-dominated set. Since in practical MOCO problems this set can be very large, this aspiration criterion is equally stronger.

In our experiments we use $M = 10000$ as the maximum number of elements in the hashing tabu-list. Thus, if the hashing function values in the hashing list are uniformly distributed in the 32-bit domain, the probability of collision is less than 3 in a million when the hashing tabu-list is full. The aspiration criterion consists in allowing neighborhood solutions whenever they contribute to the non-dominated set.

3.2. Evaluating an approximation

For evaluating the performance of an approximation, $A = \{a_1, a_2, \ldots, a_n\}$, we compare the approximation with a fixed reference set, $Z^* = \{z_1, z_2, \ldots, z_m\}$, where all a_i and z_j are K-dimensional vectors in the objective space, K being the number of objectives. We will assume that each vector in the approximation is either equal to one of the vectors in the reference set or dominated by one of the vectors in the reference set. While many evaluation measures can be estab-

(Wierzbicki, 1986) in the following way (maximization on all objectives):

$$S^*(A, Z^*, \lambda) = \frac{1}{m} \sum_{j=1}^{m} \min_{i=1..n} \{ \max_{k=1..K} \{ \lambda^k (z_j^k - a_i^k) \} \}.$$

This measure can be seen as the average of the λ-weighted Tchebycheff distances from each vector in the reference set to its projection on the set A. The λ-vector contains range equalization factors (Steuer, 1986) and is calculated as:

$$\lambda^k = \frac{1}{\text{Range}^k} \Big[\sum_{\kappa=1}^{K} \frac{1}{\text{Range}^\kappa} \Big]^{-1}, \text{ where } \text{Range}^k = \max_{j=1..m} (z_j^k) - \min_{j=1..m} (z_j^k).$$

3.3. Experiments on the budgeting problem

We use tabu lists of lengths 4 and 8 and sample sizes of 20 and 100, giving 4 cells in the experimental design, which are to be examined both with and without the hashing component. Each cell is composed of 10 replicates, giving a total of 80 experiments. In each experiment, 2 million neighbors are generated; a sample size of 20 therefore means that 100,000 moves are made, whereas a sample size of 100 means that 20,000 moves are made. From the 80 resulting approximations to the non-dominated set, a reference set Z^* is generated as all distinct points from the approximations, except for the points which are dominated by other points; or in other words, the best, dominant-free set. Table 1 shows the average of the S^* values and other statistics from the 10 repetitions for each design of the experiment, using the reference set Z^*.

	Without the hashing component				With the hashing component			
Tabu-list length	4	4	8	8	4	4	8	8
Sample size	20	100	20	100	20	100	20	100
Repetitions	10	10	10	10	10	10	10	10
Average	0.00031	0.00043	0.00024	0.00028	0.00032	0.00031	0.00027	0.00027
Minimum	0.00026	0.00032	0.00017	0.00021	0.00026	0.00024	0.00021	0.00021
Maximum	0.00034	0.00055	0.00027	0.00031	0.00041	0.00042	0.00034	0.00032
95% confidence-	0.00029	0.00038	0.00022	0.00026	0.00028	0.00027	0.00024	0.00024
interval for average	0.00033	0.00048	0.00026	0.00031	0.00035	0.00036	0.00030	0.00029

Table 1. Statistics on the S^* values from the budgeting problem

We notice that the hashing component apparently does not present any real difference. We will assign this to the randomness attached to the neighborhood function, which allows the search to break cycles before they can become problematic. Only for a high sample size and a short tabu-list does the hashing component have a significant positive influence. However, better results are obtained by increasing the length of the tabu-list than by introducing the hashing

3.4. Experiments on the power network design problem

In this problem, tabu-list lengths of 5, 10 and 15 have been used, both with and without the hashing component. Each cell of the 6 designs is composed of 10 replicates and in each of these 60 experiments, 40,000 neighborhood moves have been made. Again, the reference set Z^* is generated as the best dominant-free set of points. Table 2 shows the statistics.

	Without the hashing component			With the hashing component		
Tabu-list length	5	10	15	5	10	15
Repetitions	10	10	10	10	10	10
Average	0.0024	0.0020	0.0022	0.0018	0.0017	0.0017
Minimum	0.0019	0.0014	0.0016	0.0013	0.0013	0.0015
Maximum	0.0031	0.0025	0.0028	0.0021	0.0020	0.0020
95% confidence-	0.0022	0.0018	0.0019	0.0016	0.0016	0.0016
interval for average	0.0027	0.0023	0.0025	0.0019	0.0019	0.0018

Table 2. Statistics on the S^* values from the power network design problem

Here, on the other hand, we clearly improve all results by including the hashing component, even in the tabu search implementations with rather long tabu-lists. With a short tabu-list, the hashing component becomes increasingly necessary, indicating that the hashing component helps prevent cycles when the tabu list can not.

Final remarks

In knapsack problems, such as the budgeting problem of this paper, we will often be able to obtain very good results using yield per cost ratios and this can also be used for making better neighborhood function for ascent based methods, such as tabu search. The neighborhood functions used in this paper were chosen in order to make direct comparison of the results with earlier experiments. Due to the randomness, which to a large extent already reduces cycling, it does not exploit the potential of the hashing component. In the network design problem, however, a positive effect is observed.

In the single objective knapsack problems, the "real" problem is often to determine inclusion or not of a smaller subset of items, the so called core-items. This could be advantageously exploited in the hashing-mapping, reducing the number of actual collisions. The idea is not so futile in the multiobjective case, where the core-items will be different over the non-dominated frontier.

In multiobjective optimization, in general, we wish to search the objective space rather than the decision space. So, unless we wish to generate the efficient set (as opposed to the non-dominated set), we are not directly interested in finding more than one solution having the same objective function vector.

ing mapping. However, one must be aware that different solutions having the same objective function vectors in most cases will be parts of different search paths.

Finally, as a more general remark on the hashing method, it should be emphasized that while cycling is obviously undesired, revisiting a solution may not be, if it is entered with a different trajectory (which, to be precise, results in the selection of a different neighboring solution). Therefore, it may be possible to modify the hashing method to include both the solution and (parts of) the trajectory. This can in practice be implemented in many ways — for instance by including (the active parts of) the tabu list in the hashing function or by using a 1-step back-tracking mechanism. These ideas can be the subject of future research.

References

CZYŻAK, P. and JASZKIEWICZ, A. (1996) A multiobjective metaheuristic approach to the location of a petrol station by the capital budgeting model. *Control and Cybernetics*, **25**, 177-187.

CZYŻAK, P. and JASZKIEWICZ, A. (1998) Pareto simulated annealing — a metaheuristic technique for multiple objective combinatorial optimization. *Journal of Multiple Criteria Decision Analysis*, **7**, 34-47.

FONSECA, C. M. and FLEMING, P. J. (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. *Proceedings of the ICGA '93.* Morgan Kaufmann Publishers, 416-423.

FONSECA, C. M. and FLEMING, P. J. (1995) An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, **3**, 1-16.

FORTEMPS, P., TEGHEM, J. and ULUNGU, B. (1994) Heuristics for multiobjective combinatorial optimization by simulated annealing. *XI^{th} International conference on MCDM*, Coimbra 1-6.08.94.

GANDIBLEUX, J., MEZDAOUI, N. and FREVILLE, A. (1996) A tabu search procedure to solve multiobjective combinatorial optimization problems. To appear in: R. Caballero and R. Steuer, eds., *Proceedings volume of MOPGP '96.* Springer-Verlag.

GLOVER, F. (1989) Tabu Search — Part I. *ORSA Journal on Computing*, **1**, 3, 190-206.

GLOVER, F. and LAGUNA, M. (1997) *Tabu Search.* Kluwer.

HABENICHT, W. (1982) Quad trees. A data structure for discrete vector optimization problems. *Lecture Notes in Economics and Mathematical Systems*, **209**, 136-145.

HANSEN, M. P. (1997) Design of a power network (in preparation).

HANSEN, M. P. (1998) *Metaheuristics for multiple objective combinatorial optimization.* Ph.D. dissertation, IMM-PHD-1998-45, Institute of Mathemat-

HANSEN, M. P. (2000) Tabu search for multiobjective optimization: TAMOCO. *Control and Cybernetics*, this issue.

HORN, J. and NAFPLIOTIS, N. (1993) Multiobjective Optimization using the niched Pareto genetic algorithm. IlliGAL report No. 93005.

HORN, J., NAFPLIOTIS, N. and GOLDBERG, D. E. (1994) A niched Pareto genetic algorithm for multiobjective optimization. *Proceedings of the first IEEE conference on evolutionary computation*, Vol. 1, 82-87.

MATOS, M. A. and PONCE DE LEÃO, M. T. (1995) Electric distribution systems planning with fuzzy loads. *International Transactions of Operational Research*, **2**, 3, 287-296.

SCHAFFER, J. D. (1985) Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the ICGA '85.* Lawrence Erlbaum, 93-100.

SERAFINI, P. (1987) Some considerations about computational complexity for multiobjective combinatorial problems. *Lecture Notes in Economics and Mathematical Systems*, **294**, 222-232.

SERAFINI, P. (1992) Simulated annealing for multi objective optimization problems. *Proceedings of the X^{th} International conference on MCDM*, Tapei 19-24.07.92, 87-96.

SRINIVAS, N. and DEB, K. (1995) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, **2**, 3, 221-248.

STEUER, R. E. (1986) *Multiple Criteria Optimization: Theory, Computation, and Application.* John Wiley & Sons.

ULUNGU, E. L. and TEGHEM, J. (1994) Multi-objective combinatorial optimization problems: a survey. *Journal of Multiple-Criteria Decision Analysis*, **3**, 83-104.

WIERZBICKI, A. P. (1986) On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spektrum*, **8**, 73-87.

WOODRUFF, D. L. and ZEMEL, E. (1993) Hashing vectors for tabu search. *Annals of Operation Research*, **41**, 123-137.