Cooperative Simulated Annealing: How much cooperation is enough?

Oliver Wendt, Wolfgang König
Frankfurt University, Germany
{wendt | koenig}@wiwi.uni-frankfurt.de

This paper provides empirical evidence in support of the hypothesis, that a populational extension of simulated annealing with cooperative transitions leads to a significant increase of efficiency and solution quality for a given combinatorial optimization problem (and a neighborhood relation) if and only if the globally optimal solution is located "close" to the center of all locally optimal solution.

1. Introduction

Many well-structured problems encountered in business (and other domains) can be defined as optimization problems

\[ \text{P: } \max (\min) f(x) \]
\[ \text{s.t. } x \in S \]

where \( S \) represents the set of admissible solutions (satisfying all hard constraints) of an underlying search space \( X \). The function \( f:S \rightarrow \mathbb{R} \), where \( \mathbb{R} \) represents the set of real numbers, is often called goal function or objective function.

A large subclass of these problems is the class of combinatorial optimization problems (COPs). An optimization problem is called combinatorial,

"if \( X \) and \( S \) are combinatorial or discrete in some sense. Although there does not seem to be the formal common understanding of what is 'combinatorial', we consider here that \( X \) and/or \( S \) are 'combinatorial' if they are discrete sets of finite elements or countably infinite elements." [IBARAKI 87, S.3]
For example, all problems of finding the optimal assignment or permutation of any discrete "items" are COPs, no matter how their objective function looks like or what domain they belong to.

Many COPs modeled when solving real-world problems are NP-complete\(^1\). Even for moderate problem sizes the computational effort to find their optimal solution is far beyond any realistic bounds. For this reason we have to employ so-called heuristics which abandon our search for provably optimal solutions for the sake of search efficiency.

In this paper we discuss critical success factors for using Cooperative Simulated Annealing (COSA), a hybrid search method, combining the basic principles of two major paradigms of heuristic search:

- Simulated Annealing (developed in analogy to thermodynamics) and
- Genetic Algorithms (developed in analogy to the theory of biological evolution).

### 2. Paradigms of Local Search

For many different classes of problems, the simplest heuristics known are local search heuristics, starting with a random solution \(s_i\) from the solution space \(S\) and then successively trying to "modify" this solution until some better solution \(s_j\) is reached. After this, we focus on improving \(s_j\) in order to find a better "neighbor" and after a finite number of steps reach a locally optimal solution, better than or equal in quality to all its neighbors. Before we can do such a local search we need to define "locality" in the sense that there has to be a binary neighborhood relation \(N \subseteq S \times S\), telling the search process, whether \(s_j\) is in fact a direct neighbor to \(s_i\). Defining an appropriate neighborhood relation \(N\) for a given class of COPs is certainly the most critical success factor for any type of local search:

- If the number of direct neighbors of a given solution is too big, the quality of a specific solution does not tell much about the quality of its neighbors and thus the search process cannot "learn" by moving to promising "regions" of the search space. Imagine the extreme case of a complete neighborhood relation: When every solution is a direct

---

\(^1\) To be exact: Not the optimization problem itself is NP-complete but its corresponding "decision problem" which consists of telling whether there exists a solution with an objective function value better than a given value.
neighbor of every other solution in $S$, there are no "regions" at all and a local search of length $l$ is equivalent to a random sampling of $l$ solutions and picking the best one.

- On the other hand, if the number of direct neighbors of a given solution is too small, the neighborhood relation $N$ will not be exact [PAPADIMITRIOU 82]. Exactness means, that from any solution $s_i$ there is at least one finite path of (bilaterally) neighbored solutions $[s_i, s_{i+1}, s_{i+2}, \ldots , s^*]$ to a globally optimal solution $s^*$ that comes with a monotonically decreasing (assuming a minimization problem) sequence of objective function values $[f(s_i), f(s_{i+1}), f(s_{i+2}), \ldots , f(s^*)]$. If this is not the case, a local improvement search may get stuck in a locally optimal solution without ever getting to the global optimum.

2.1. Simulated Annealing

Since for many COPs it has been impossible to find small exact neighborhoods, the development of Simulated Annealing (SA), initiated by the seminal work of [KIRKPATRICK 83] is considered to provide an elegant escape from this dilemma: Rather than trying to improve on the structure of the search "landscape" induced by the neighborhood relation $N$, SA modifies the acceptance criterion for a neighbored solution in order to get out of local optimal again.

Instead of only accepting a neighbor $s_j$ of the current solution $s_i$ as the new starting point for further search steps when $f(s_j) \leq f(s_i)$, SA also accepts the new solution $s_j$ when its objective function value is worse than the old one’s, but in this case only with the so-called Metropolis probability

$$P_i^A(T) = e^{-\frac{f(s_i) - f(s_j)}{T}},$$

which is a decreasing function of the difference in objective function values and an increasing function of a virtual control variable $T$, called temperature. As long as this temperature is very high, SA accepts every new solution, thus yielding a random walk through the search space. On the other hand, with a temperature close to zero, only improvements are accepted. It has been shown that, if the SA process starts with a high temperature and cools down "slowly enough", a globally optimal solution will eventually be reached [HAJEK 88]. Unfortunately, the bounds for "slowly enough" do in fact depend on the properties of the search landscape and have to be exceeded for most practical cases. In this case, SA falls into the class of heuristics. In empirical tests SA has proven its superiority compared to classical heuristics, as far as solution quality is concerned. On the other hand, this quality often comes at the cost of very long computation time [GOLDEN 86].
2.2. Genetic Algorithms

As a (competing) paradigm, Genetic Algorithms\(^2\) (GA) are trying to find good solutions to combinatorial optimization problems by imitating natural evolution processes. The reasoning is quite simple: When the principles of biological evolution were able to produce very complex "solutions" to the optimization problem "How to survive and reproduce in a given environment?", why shouldn't they be able to find good solutions to (much simpler) artificial optimization problems?

After an initial population of (randomly generated) solutions to a given problem is generated, the "fitness" of these solutions (i.e. their objective function value \(f(s_i)\)) determines, how likely it is for a given solution to "crossover" with other solutions of the population in order to produce "children". These children inherit structural information of their parents and compete among each other for being chosen as parents to produce the third generation. From time to time, a random mutation of an individuals structure takes place to prevent the total loss of "alleles" from the population. Eventually, after a finite number of generations, fitness does not improve anymore and the best solution is taken as the best solution obtained to the given problem.

Therefore, the main difference to SA is the fact that the current state of the search process is not represented by a single solution \(s_i\) but rather by a population \(POP = \left(I_1 \in S, I_2 \in S, \ldots, I_{|POP|} \in S\right) \in S^{|POP|}\) of several 'individuals'. For each individual \(I_k \in S\) which is currently part of the population, there is a probability to undergo a so-called mutation, comparable to SA's replacement of a solution by one of its direct neighbors in \(N \subseteq S \times S\). But in GAs this probability does NOT depend on the solutions objective function value. While this mutation operator is only considered to be a "background operator", the primary search operator is the crossover operator, which implicitly defines a more complex ternary crossover neighborhood \(NX \subseteq S \times S \times S\), telling us, which solutions \(s_k\) within \(S\) may be reached from a single crossover of two individuals representing solutions \(s_i\) and \(s_j\).

Therefore, the set of solutions reachable from a given population within one generation is not only determined by the individuals own position in the search space but also by their relative positions to each other. Due to this complexity, there are no commonly accepted rules for selecting appropriate probability distributions for generating and accepting new solutions based on this ternary crossover neighborhood \(NX\).

---

\(^2\) An introduction to GAs can be found in [GOLDBERG 89]. For a theoretical analysis of GAs performance on different fitness landscapes refer to [HORN 95].
Although there is a huge number of publications available, praising the results obtained by genetic algorithms (often comparing them to constructive heuristics or pure local improvement search despite of SA), GAs suffer from the following problems:

The so-called schema theorem, forming the theoretical basis for using crossover as the primary search operator, only holds when the two neighborhoods \(NX\) and \(N\) allow every possible population to perform a sequence of transitions leading to the global optimum with a high enough probability. Otherwise, the whole population gets stuck in (only) locally optimal regions of the search space, a phenomenon called "premature convergence" by the GA community.

While it is already difficult to find good local neighborhood relations \(N\) for most combinatorial optimization problems, it is even more difficult to find good crossover neighborhoods \(NX\).

### 2.3. Cooperative Simulated Annealing (COSA)

After we ran into trouble with Genetic Algorithms (concerning the poor performance of known crossover operators for permutation problems) and Simulated Annealing (concerning the huge number of transitions necessary for satisfactory results) when solving vehicle routing problems we had to answer the following question:

Is there any alternative way of EXCHANGING information within a population apart from crossover operators, performing smaller "jumps" on the fitness landscape, thus needing less "repair information" and finding a more efficient way towards the global optimum?\(^3\)

As an answer to this question we developed COSA, a hybrid local search algorithm, combining features of both, GA and SA [WENDT 95].

COSA inherits the idea of population and information exchange from Genetic Algorithms but replaces the usual crossover by so-called cooperative transitions. The acceptance probability for the transition is controlled by the Metropolis function, depending on the virtual "temperature" and the difference in the solutions’ goal function values (as usually done in SA).

COSA thus implements a concurrent\(^4\) but synchronous run of multiple SA processes, loosely coupled by the cooperative transitions. These cooperative transitions replace the usual uniform probability distribution \(P_{ij}^G\) (the probability of generating neighbor \(s_j\) from solution \(s_i\)) by a

---

\(^3\) For individuals that are very “far away” from each other concerning their position in search space, crossover can be compared to the very unpromising effort to recombine animals of different species.

\(^4\) It is irrelevant whether this concurrency is only a logical parallelism or whether the processes are actually run on parallel hardware.
"skewed" distribution, which relates the direction of the (potential) search step to the current position of other individuals in the population. Analogous to a "gravitation force", moves reducing the distance to some other individual get assigned a higher probability than steps increasing this distance.

The following diagram gives a formal (domain independent) description of COSA:

```
procedure COSA

initialize size of population \( |POP| \)
generate initial population \( POP_0 = \{ I_1 \in S, I_2 \in S, \ldots, I_{|POP|} \in S \} \in S^{|POP|} \)
initialize neighborhood relation \( N \subseteq S \times S \)
initialize number of transitions \( K = n \cdot |POP| \) with \( n \in \text{NAT} \)
initialize temperature \( T_1 \in \text{REAL}_+ \)
and adaptive cooling factor \( \alpha \in ]0;1[ \subseteq \text{REAL}_+ \)

for \( k := 1 \) to \( \frac{K}{|POP|} \) do

for \( i := 1 \) to \( |POP| \) do

randomly select cooperator \( I_j \in POP_{k-1} \)
generate \( I_{\text{new}} := \text{cotrans}(I, I_j) \)

if \( \text{random}(0,1) \leq \frac{1}{e^{-\frac{f(I_{\text{new}}) - f(I_i)}{\alpha T_k}}}, f(I_{\text{new}}) < f(I_i) \)
then \( I_i := I_{\text{new}} \)

next \( i \)

\( POP_k := (I_1, I_2, \ldots, I_{|POP|}) \quad T_{k+1} := \text{update}(POP_{k-1}, POP_k, T_k, \alpha) \)

next \( k \)
end.
```

*Fig. 1: The COSA algorithm (for minimization problem)*
**Function update**

\[
\text{update}: \left( POP_k \in S^{\text{pop}}, \ POP_{k-1} \in S^{\text{pop}}, \ T \in \mathbb{R}_+, \ \alpha \in \mathbb{R}_+ \right) \rightarrow T \in \mathbb{R}_+
\]

\[
\Delta E := E(POP_k) - E(POP_{k-1})
\]

\[
T := \begin{cases} 
T, & \text{if } \Delta E < 0 \\
\alpha T, & \text{if } \Delta E \geq 0
\end{cases}
\]

return \( T \).

**Fig. 2:** Function "update" implements an adaptive decrease of temperature.

**Function cotrans**

\[
l_i \in S, I_j \in S \rightarrow I_{\text{new}} \in S
\]

select those neighbors of \( l_i \) which are closer to \( I_j \) than \( I_i \) itself

\[
\text{(where } d(x,y) \text{ is the distance measured as the minimal number of search steps to reach } y \text{ from } x)\]

\[
\text{CLOSER} := \left\{ s_k \in N(l_i) \mid d(s_k, I_j) < d(I_i, I_j) \right\}
\]

if \( |\text{CLOSER}| \geq 1 \)

then randomly select \( I_{\text{new}} \in \text{CLOSER} \)

else randomly select \( I_{\text{new}} \in N(l_i) \)

return \( I_{\text{new}} \).

**Fig. 3:** Function 'cotrans' returns a cooperatively generated neighbor

Function 'update' compares the mean energy (fitness) of the current population \( POP_k \) with the mean energy of the "ancestor population," \( POP_{k-1} \). Iff it increased, temperature \( T \) gets multiplied by the annealing factor (e.g. 0.999), otherwise \( T \) is kept constant. The rationale behind this procedure is the following: Whenever \( T \) gets decreased it takes some time for the population to reach the thermodynamic equilibrium determined by the new temperature. If the population is big enough a rising mean energy is a good criterion for having reached this equilibrium,... and now starting to randomly fluctuate around this equilibrium (although there may of course be stochastic rises in mean energy before having reached the equilibrium).

Function 'cotrans' gives the domain independent formal description of the cooperative transitions.
For the Capacitated Vehicle Routing Problems (CVRP) the implementation of the cotransoperator is slightly more tedious than the TSP version described above. But it basically follows the same gravitation mechanism [WENDT 95].

In the following sections of this paper we will

- illustrate how to implement cooperative transitions for specific COPs
- try to find the best ratio of cooperative and non-cooperative transitions for the specific problem class.

3. Performance of COSA on the landscapes of different COP

3.1. The Traveling Salesman Problem (TSP)

The search for the shortest hamiltonian cycle of a graph, or, less formally speaking, the shortest round-trip through a given number of cities, has been one of the most popular combinatorial optimization problems for half a century [LAWLER 85]. For symmetric TSPs, the so-called “2-change” is often chosen as the most simple local search operator defining a neighborhood $N$ on the solution space by neighboring two solutions $s_i$ and $s_j$ iff they share all edges except for two of them [e.g. $(1\ 2\ 3\ 4\ 5)$ and $(1\ 4\ 3\ 2\ 5)$ are 2-change neighbors since the second tour can be obtained by replacing the first tour’s edge $(1\ 2)$ by the edge $(1\ 4)$ and the first tour’s edge $(4\ 5)$ by $(2\ 5)$]. The following graph illustrates this neighborhood relation for the 12 tours of a symmetric 5-city-TSP. As we see, every solution has five direct neighbors and (in this simple case) the neighborhood is even exact.

![Fig. 4: 2-change neighborhood relation of a 5-city-TSP](image-url)
In contrast to classical SA, where both removed edges are chosen randomly in every transition, for a cooperative transition we only chose one random edge to be removed from an individual $I_i$. Assume we randomly select the dotted edge (fig. 5) to be removed from $I_i$. Instead of selecting the second edge randomly (thus implicitly determining both new edges, too), we rather determine one of the new edges first.

![Fig. 5: A cooperative transition for the symmetric TSP](image)

This is the point where cooperation enters the game: Why should we choose any random edge when there is plenty of information available in other individuals concerning replacement edges worth considering? We therefore randomly select a second individual $I_j$ from the population and "ask" it for a proposal, meaning we check which edge leaves one of $I_j$'s two cities needing a new edge in individual $I_i$. Assume that $I_j$ looks like the right tour in fig. 5 then the dotted edge might be one possible answer. Having chosen one edge to be deleted from $I_i$ (randomly) and one new edge that has to be inserted into $I_i$ (by cooperation) leaves no choice for the second edge to be deleted and the second to be inserted. Thus our new candidate $I_i'$ looks like the third tour in fig. 5.

Whether we actually accept this transition, i.e. replace $I_i$ by $I_i'$, solely depends (like in classical SA) upon the Metropolis probability. In case we accept, the individual $I_i$ got a little bit "closer" to $I_j$ by now sharing 5 common edges (compared to 4 common edges before undergoing the transition). Thus, although we only walk along the classical 2-change neighborhood and we have no such thing as a "recombination neighborhood" $NX$, the individuals positions in search space converge (like they do in genetic algorithms).

Compared to the simple structure of the COSA algorithm the results we obtained in our empirical tests were surprisingly promising:

On standard instances of the Traveling Salesman Problem we obtained a significant improvement of efficiency and solution quality (compared to classical TSP heuristics, pure GA and pure SA strategies). For example we ran COSA 25 times on a 51-city problem defined by Christofides and Eilon [CHRISTOFIDES 69] with a population size of 100 and a "total budget" of 500000 transitions (i.e. only 5000 transitions per SA-process) and we
reached the global optimum (428.87) in each of the 25 runs\textsuperscript{5}. The following figure 6 shows the tremendous positive effect of using cooperative transitions: As soon as we reduce the percentage of cooperative to non-cooperative transitions\textsuperscript{6}, we get slower convergence (without getting rewarded by better results).

![Graph showing results for different percentage of cooperative transitions for a 51-city-TSP](image)

*Fig. 6: Results for different percentage of cooperative transitions for a 51-city-TSP*

Even when we decide to "freeze" the system immediately, i.e. by choosing a starting temperature of zero force all 100 SA processes to only accept transitions leading to an improvement in the objective function, we get the same picture: The higher the percentage of cooperative transitions, the better our results (fig. 7).

\textsuperscript{5} with an initial temperature of T=3 and an adaptive cooling rate of $\alpha=0.99$

\textsuperscript{6} randomly deciding which one to take at every transition
Since the positive effect of cooperative transitions nicely scales up with problem size, we also succeeded to reproduce the solution of the TSP318 [LIN 73] which has been proven to be optimal by a (very problem specific) cutting plane approach [CROWDER 80]. For the printed circuit board drilling problem TSP442 [GRöTSCHEL 91] a solution quality of 50881 has been obtained.

An equally impressive picture we obtained when solving the Capacitated Vehicle Routing Problem (CVRP) with COSA. Despite of the 25-year development of heuristics for solving CVRP, COSA succeeded to improve the results of other (often very problem specific) heuristics on several standard benchmarks [WENDT 95].

### 3.2. Job Shop Scheduling Problems (JSSP)

Motivated by the success of COSA in the vehicle routing domain we tried to adapt the idea of cooperative transitions to other problem classes, notoriously hard to find good crossover operators for. JSSP belong to this type of problems. The objective is to minimize the makespan of a set of $n$ jobs which have processed on $m$ machines. Every job has a predefined processing order on the $m$ machines and a given processing time on each machine. The following figure 8 represents three possible solutions of different total length to a 4*3-JSSP.
The simplest neighborhood relation we could imagine for the JSSP is to exchange the execution order of two consecutive operations on a specific machine. In the example above, we may for example swap the order of the black and the dotted operations on machine 1 or we may do the same with the black and the gray operations. After rescheduling the remaining operations we may get a neighboring solution (probably with a different makespan when both operations belong to the critical path), which we may either reject or accept.

Based on this neighborhood relation there is a simple cooperative version of this exchange: First we choose a random operation on a random machine. As long as this operation is neither the first nor the last to be executed on this machine, it has to adjacent operations it can be swapped with. Assume that we chose the first machine and the dotted operation in figure 8. To answer the question whether it should be processed after the gray or before the black operation in $I_i'$, we again "consult" a second individual $I_j$ of the population. In this individual $I_j$, the dotted operation is either processed at an earlier position than in $I_i'$, at a later position or at the same position. In the first case, the dotted operation gets swapped with its left neighbor, in the second case with its right neighbor and in the third case the neighbor is chosen at random.

As illustrated in fig. 9 and 10, the picture is less favorable for COSA in the JSSP case: Although for “toy problems” like the 6*6 JSSP example from [FISHER 63] convergence and solution quality increases with an increase of cooperation rate, this effect vanishes with increasing problem size: For the 10*10 JSSP from [FISHER 63] we still get faster convergence but for the price of less solution quality reachable when the SA processes get into the low temperature regime at higher number of transitions. Of course, like premature convergence in Gas, this phenomenon may be fought by a bigger population size, but when doing this, it is still advantageous to keep the cooperation rate low to obtain best quality for a given number of transitions.
Fig. 9: Results for different percentage of cooperative transitions for a 6*6-JSSP

Fig. 10: Results for different percentage of cooperative transitions for a 10*10-JSSP
3.3. Selection of Communication Protocols

Our third class of problems chosen for the adaptation of COSA was the following: Imagine nodes of a communication system, having the choice to implement one or more different communication protocols (like humans learning different languages or offices buying fax machines or internet connectivity). On the one hand, implementing any protocol (from a fixed set of possibilities) comes with a given cost, but, on the other hand, communication cost (defined bilaterally between every pair of nodes as a function of a given protocol) may decrease significantly by this investment, especially, when many nodes chose the same standard and thus may use it for a high number of communication partners. The objective is to minimize the total cost given by the sum of communication and protocol implementation cost for all nodes of the communication system. A detailed overview over this field of protocol selection and standardization economics is given in [BUXMANN 96].

Since every solution may easily be represented by a bit matrix, indicating, which standard will be implemented on which node, the simplest neighborhood relation is given by flipping a single bit. The cooperative version only flips one of those bits of $I_i$ that differ from the bits in $I_j$. This is illustrated in fig. 11.

![Fig. 11: A simple cooperative transition for the protocol selection problem](image)

As we see from fig. 12 (communication system with 100 nodes and implementation and communication costs randomly sampled from the interval $[0; 1000]$) we get the same “premature conversion” problems here that we got with JSSP.
4. **Towards a domain independent theory of cooperative search**

Of course, it would be possible to extend the list of cooperative transitions for new problem classes to an arbitrary length, but of course it would be helpful to know in advance, which abstract properties of a search landscape (defined by $S$, $f$ and $N$) will be good indicators for success or failure of cooperative local search.

It is clear from the definition of COSA, that the more cooperative transitions we use, the faster the population of individuals converges. And since there is not only a “gravitation force”, but also a Metropolis probability, the population gradually converges to “promising” regions, i.e. those with a high number of local optima. The key question now is: Are these regions really promising or rather misleading. Or, in other words: Is the global optimum really located somewhere within the “cloud” of local optima? If this is the case, then cooperative transitions may tremendously increase the probability of finding it by increased search intensity within this “cloud” for the price of reduced intensity in “outer regions” of the landscape.
Although the illustration of these two cases (for a very simple grid neighborhood relation) in fig. 13 might look very plausible, we need measurable characteristics of a landscape and the “cloud of local optima” on which we could build a formal definition of our hypothesis.

Fig. 14 measures some properties of the four landscapes discussed in the chapters above (logarithmic scale!):

The rightmost group of bars stands for the average distance of two solutions picked randomly from the search space. (Distance is always measured in “number of edges on the shortest path(s) in the graph induced by the given neighborhood relation $N$.”) This average distance
increases with the number of solutions in $S$ (nodes in the graph) but decreases with the degree of the nodes, i.e. the number of neighbors.

For estimating the other properties we generated 1000 local optima by sampling 1000 random solutions and applying a local improvement search until we reached a locally optimal solution. As we see from the second group of bars (from the left), the average distance between two randomly chosen local optima is much smaller than it is for randomly chosen solutions which are not "pushed" into the next local optimum, i.e. the notion of a "cloud" (or several clouds) of local optima seems to be appropriate for all four problems.

The diameter is defined by the maximum of the distance of all pairs of local optima (within the sample of 1000 local optima). In order to determine the radius of the set of local optima we first have to find the center\(^7\) of all local optima and then measure the maximal distance from this center to any other local optimum.

The first group of bars tells us now, what we want to know, namely how far the globally optimal solution is away from this center of the set of local optima. For the 51-city-TSP and the 6*6-JSSP this distance is less than half the radius of all 1000 local optima, this is a totally different story for the 10*10-JSSP and especially for the protocol selection problem: The distance of the globally optimal solution to the center is that high, that its position must be somewhere in the "outer regions" of the cloud of local optima. This provides a good explanation for the failure of cooperative transitions in the these two cases.

5. Open Questions and Outlook

Apart from further empirical studies with different problem classes and instances it should be possible to set up a theoretical proof of our central hypothesis by modeling the COSA process as a markovian chain like this is done for the convergence proof of SA [HAJEK 88]. Unfortunately, the number of states of this markov chain blows up from $|S|$ to $|S|^{pop}$, thus limiting the computational tractability of the analysis to toy problems and very small population sizes.

Furthermore, it is not at all clear why cooperation has to imply and attraction force and thus foster convergence. We might also think of cooperative transitions which increase divergence by assigning higher probability to moves "further away" from other individuals of the population thus increasing exploration of the search space.

\(^7\) The center of a graph is the node (or one of the nodes) for which the maximum distance to any other node of the graph is minimized.
6. References


[IBARAKI 87] Ibaraki T.: *Enumerative Approaches to Combinatorial Optimization*; Basel (Baltzer).


[NAGEL 78] Nagel F.: MUDDY - Ein Verfahren zur Lösung großer Mehrdepot-Lieferplanprobleme; Kiel (Universität Kiel)
