Exercise 3.1
Design a guided local search algorithm for the quadratic assignment problem. Consider, in particular, the computation of the utility measure as it is required for guided local search.

Exercise 3.2
One general idea that can be exploited in SLS is to modify the input data associated to an instance slightly and to use the modified data to perform a local search on it. Such data perturbations we have seen in the lectures as an example in a complex perturbation scheme for an ILS algorithm for the TSP.

- Consider how such a data perturbation scheme can be applied to non-geometric problems such as the single-machine total weighted tardiness problem or the quadratic assignment problem.
- Consider how this idea of data perturbation can be exploited to design a general SLS method (e.g. based on perturbative local search).

Exercise 3.3
In almost all GRASP implementations, the iterations are independent of each other. In particular, GRASP does not use memory between the iterations and, thus, does not learn from the history of the solutions found. Which kind of memory could be introduced into GRASP to

- improve the solution quality;
- avoid redundant work.

Propose such strategies that make the iterations of GRASP dependent on each other. In particular, consider how elements from other SLS methods we have seen in the lectures could be adapted for usage together with the ideas underlying GRASP.

Exercise 3.4
The ant colony optimization (ACO) metaheuristic includes in its definition the possibility of exploiting local search. When local search is used, the ACO algorithm can be interpreted as a way of generating starting points for the local search procedure. Each ant produces a solution, say $s_{ant}$, which is then transformed into another solution, say $s_{ls}$, by the local search. Then the pheromones are updated. There are two ways of updating the pheromones:

- Reinforce the pheromones corresponding to the final solution $s_{ls}$. In other words, do as if the solution $s_{ls}$ was generated directly by the ant, without the help of the local search.
- Reinforce the pheromones corresponding to the intermediate solution $s_{ant}$.

What reasons would you give in support of one or the other of the two choices? In each of the two cases, how would you choose the amount of pheromone that needs to be deposited?

Exercise 3.5
The $N$-queens puzzle is the problem of putting $N$ chess queens on an $N \times N$ chessboard such that none of the queens attacks any of the other queens. Design SLS algorithms for this problem. To do so, consider the following questions.

- Develop an appropriate evaluation function for this problem.
• How are initial solutions generated?

• Define appropriate neighborhood structures for perturbative local search algorithms.

• Design metaheuristic algorithms for the $N$-queens puzzle. Consider the main features that have to be defined for the various methods we have seen in the lectures and define accordingly these algorithms. In particular, consider randomized iterative improvement, simulated annealing, tabu search, iterated local search, and iterated greedy.

Remark: The well–known min-conflicts heuristic was originally tested using the $N$-queens puzzle as an example problem. The min-conflicts heuristic is an SLS method for tackling constraint satisfaction problems (CSPs). The RII extensions of the min-conflicts and tabu search extensions of the min-conflicts heuristics have proven to yield very good performance on CSPs.

Task: Search information on the internet on the min-conflicts heuristic (e.g. Wikipedia). For a simulator of some algorithms for this problem see http://yuval.bar-or.org/index.php?item=9

Exercise 3.6

In the Generalized Assignment Problem (GAP) a set of tasks $I, i = 1, \ldots, n$ has to be assigned to a set of agents $J, j = 1, \ldots, m$. Each agent $j$ has a limited capacity $a_j$ and each task $i$ requires, when being assigned to agent $j$, an amount $b_{ij}$ of the agent’s capacity. $d_{ij}$ is the cost of assigning task $i$ to agent $j$. The goal of the problem is to find a feasible, cost minimizing assignment of agents to tasks. An assignment is feasible if the capacity constraints of all agents are satisfied. No splitting of tasks is allowed among agents.

• In all the following questions, state also how you would deal with infeasible candidate solutions (note that candidate solutions maybe infeasible due to the capacity constraints of the agents).

• Design a greedy constructive heuristic for GAP.

• Consider what would be an appropriate neighborhood structure for a perturbative local search for the GAP.

• Design metaheuristic algorithms for the GAP. Consider the main features that have to be defined for the various methods we have seen in the lectures and define accordingly these algorithms. In particular, consider simulated annealing, tabu search, dynamic local search, iterated local search, GRASP, iterated greedy, ant colony optimization, and memetic algorithms.