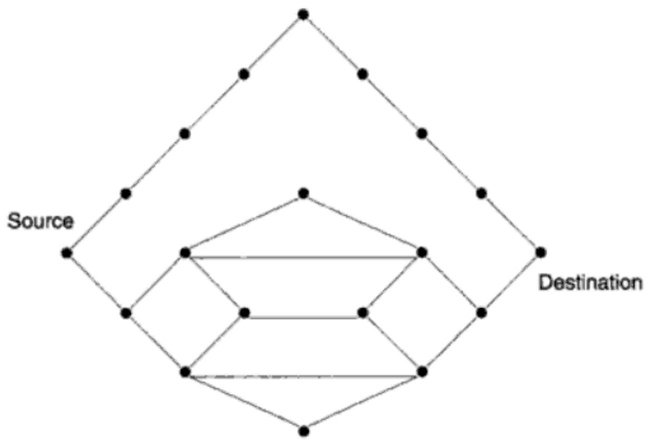
NAT-DL: Self-study questions on ACO etc. Set 4 (week 6)

1. The figure on the right shows an example from the ACO book by Dorigo and Stuetzle. What results do you expect for an ant colony algorithm that does not use taboo lists (except for inhibition of immediate return to the previous node)?



2. Discuss the application of ACO to the eight-queens puzzle. This puzzle is the problem of putting eight chess queens on an 8x8 chessboard such that none of them are able to capture any other using the standard chess queen's moves, cf. en.wikipedia.org/wiki/Eight_queens_puzzle.

3. **[Numerical exercise]** Run the standard ACO on the travelling salesperson problem with N cities. It is not difficult, but you can find code at www.aco-metaheuristic.org/aco-code/ or elsewhere, or use the code that will be become available shortly before the tutorial. Start with n_{ants}=N, α=1, β=2, rho=0.75. How can you influence the quality of the stationary solution? Consider the standard deviation of the tour length over the ants during one iteration.

4. Consider the following (very small) TSP:
   $d(A,B) = 2$, $d(A,C) = 3$, $d(A,D) = 5$, $d(B,C) = 3$, $d(B,D) = 3$, $d(C,D) = 4$.
   a. How many different tours are possible? What are the lengths of these tours?
   b. Which tour is most likely to be found by the ants?
   c. Compare the case of a single ant to a population of two or more ants on this problem.

5. Consider **one** of the following problems (or any other one that seems to be interesting) and explain how you would use ant colony optimisation to find an acceptable solution: Bin packing, graph colouring, the knapsack problem, the cutting stock problem, protein folding, the shortest common supersequence problem (for details cf. Wikipedia). Marco Dorigo has suggested approaching these problems by the following steps:
   a) Define a set of candidate solutions and the set of feasible solutions.
   b) Define a greedy construction heuristic:
      i) What are the solution components?
      ii) How do you measure the objective function contribution of addition a solution components
      iii) Is it always possible to construct feasible solutions?
      iv) How many different solutions can be generated with the constructive heuristic?
   c) Define a local search algorithm:
      i) How can local changes be defined?
      ii) How many solution components are involved in each local search step?
      iii) How do you choose which neighbouring solution to move to?
      iv) Does the local search always maintain feasibility of solutions?

Next question overleaf.

6. Recall the main algorithms that we have studied (i.e. GA, GP, ES, ACO, PSO, DE) and classify them according to Dorigo's criteria for the classification of metaheuristic optimisation algorithms. You may like to represent your answer to this question as a table.

a. Is the solution obtained by direct construction or by the use of local search?
b. Are population of solutions used or not?
c. Is a memory used within the search process or not?
d. Is the evaluation function fixed or is it modified during search?
e. Several neighbourhoods or only a single one (i.e. what topology)?
f. Inspired by biology, physics, or otherwise?
g. For what type of problems the algorithm can be expected to work well?

Can you think of new variants of the algorithms, by modification of any criteria or by recombination?