Homework 6
Ant Colony Optimization[1ex]

(Max useful score: 100 - Available points: 175)

15-382: Collective Intelligence (Spring 2019)

Instructions

Homework Policy

Homework is due on Autolab by the posted deadline. As a general rule, you have a total of 6 late days. No credit will be given for homework submitted after the late days. After your 6 late days have been used you will receive 20% off for each additional day late.

If you find solutions in any source other than the material provided, you must mention the source.

Submission

Create a zipped archive including: a PDF file with the answers to the provided questions (they can be hand-written, but in this case you must have / use a "readable" handwriting), files that have been used for dealing with the questions that require programming, a README file that specifies how to use / run the programming files. The zipped archive should be submitted to Homework 6 on Autolab.

Contents

1 Ant Colony Optimization (175 points)

A large *set partitioning problem* (SPP) has to be solved as a part of a people to table allocation scenario at a very large Indian wedding. We have invited P persons, that we index as $p = 1, \ldots, P$, that need to be seated. A number of tables of different sizes can be prepared, and at most we can use T tables in total. The tables are indexed as $t = 1, \ldots, T$. However, since some people dislike each other, while some people really want to be together, and some other people need to be together, partitioning the invited people in the tables is a difficult task. Based on the like/dislike/need information, each one of the tables that can be possibly set up can only host some subset of the people and has an associated cost measure, $c_t, t = 1, \ldots, T$. A binary coefficient matrix M is also given where the element m_{tp} of M is 1 if table t includes person p, 0 otherwise. In practice, the matrix M summarizes the match between people and the possible tables that can be prepared. The goal is to select a subset of the tables such that all invited people is seated and the total cost of the allocation is minimized.

Given the NP-hard nature of the problem, Ant Colony Optimization is selected as a meta-heuristic for tackling the resulting SPP problem.

1.1 Formulation of SPP (10 points)

Write the general mathematical formulation (as an integer programming problem) of the optimization problem to be tackled.

1.2 Pheromone model (35 points)

Since ACO has to be used to provide a heuristic solution to the problem, it is necessary to define an ACO algorithm for set partitioning problems. At this aim, the first step is to define a *pheromone model*, the pheromone variables $\boldsymbol{\tau}$, that is appropriate for the problem. The model has to be chosen to effectively guide the step-by-step solution construction of ant agents. You have to express the pheromone model in a formal way, clearly defining the mapping that represents it. You also have to compute the total number of pheromone variables that result from your choice as a function of the size (T, P) of the input problem.

Keep in mind that a pheromone model can't consist of a number of variables which is too large, otherwise it will be unfeasible to learn good values (an extremely large number of solution samples would be necessary). Also, note that the set partitioning is a subset selection problem, that differs from the ordering problems, such as the TSP, for which the pheromone model was (mostly) discussed in the class.

1.3 Heuristic variables (20 points)

Define what the heuristic variables $\boldsymbol{\eta}$ are in your ACO model and compute their total number as a function of the size (T, P) of the input problem.

1.4 ACO pseudo-code (50 points)

Provide a detailed pseudo-code (or a some equivalent description) for ACO for set partitioning. You have to write in explicit form the mathematical expressions that you will select for the decision policy, the ant-routing table, pheromone updating, and so on. Discuss the rationale behind your choices.

1.5 Integrating local search into ACO (60 points)

For routing problems (TSP, VRP, TOP) combining ACO with local search seems to be a very effective strategy. Can we do the same for set problems?

1. (40 points) Describe a possible local search procedure that would be appropriate for SPP (e.g., you need to define how the local search proceeds, based on the definition of an appropriate neighborhood for the problem, similarly to what has been shown in the class for 2-opt and 3-opt in the case of the TSP).
2. (20 points) Describe how you would combine the above local search procedure with ACO.