Department of Mathematics and Computer Science
University of Southern Denmark, Odense

November 27, 2008
Marco Chiarandini

DM812 - Metaheuristics

Assignment, Fall 2008

Exercise

Define the following components of ACO when applied to solve each of the three combinatorial optimization problems described below.

- Construction graph
- Constraints (how are they handled?)
- Pheromone trails (where are they set?)
- Heuristic information (where are they set and how are they computed?)

Single Machine Total Weighted Tardiness Problem Each of n jobs (numbered $1, ..., n$) is to be processed without interruption on a single machine that can handle no more than one job at a time. Job j ($j = 1, ..., n$) becomes available for processing at time zero, requires an uninterrupted positive processing time $p(j)$ on the machine, has a positive weight $w(j)$, and has a due date $d(j)$ by which it should ideally be finished. For a given processing order of the jobs, the earliest completion time $C(j)$ and the tardiness $T(j) = max\{C(j) - d(j), 0\}$ of job j ($j = 1, ..., n$) can readily be computed. The problem is to find a processing order of the jobs with minimum total weighted tardiness $\sum_{j=1,...,n} w(j)T(j)$.

Generalized Assignment Problem There are n kinds of items, x_1 through x_n, and m kinds of bins b_1 through b_m. Each bin b_i is associated with a budget w_i. For a bin b_i, each item x_j has a profit p_{ij} and a weight w_{ij}. An assignment is subset of items U to put in the bins. A feasible assignment is an assignment in which for each bin b_i the weights sum of assigned items is at most w_i. The assignment's profit is the sum of profits for each item-bin assignment. The goal is to find a maximum profit feasible assignment.

Mathematically the generalized assignment problem can be formulated as:

$$
\begin{array}{rl}
\text{maximize} & \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} \\
\text{subject to} & \sum_{j=1}^{n} w_{ij} x_{ij} \leq w_i \qquad i = 1, \ldots, m; \\
& \sum_{i=1}^{m} x_{ij} \leq 1 \qquad j = 1, \ldots, n; \\
& x_{ij} \in \{0, 1\} \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n;
\end{array}
$$

Set Covering Given a universe $\mathcal{U}$ and a family $\mathcal{S}$ of subsets of $\mathcal{U}$, a cover is a subfamily $\mathcal{C} \subseteq \mathcal{S}$ of sets whose union is $\mathcal{U}$. The set covering optimization problem asks to find a set covering which uses the fewest sets.