École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Computer Aided Decision Support
International Research Master 2
2013 - 2014

Operational research project draft

Ant colony optimization algorithm for a combined routing and scheduling problem

Supervisors
Jean-Charles BILLAUT
jean-charles.billaut@univ-tours.fr
Nicolas MONMARCHÉ
nicolas.monmarche@univ-tours.fr

Student
Thomas NOGUER
thomas.noguer@etu.univ-tours.fr

M2RI CADS 2013 - 2014

Université François-Rabelais, Tours

Version of January 8, 2014

Formalization of the problem

The problem can be defined from three differents subproblems :

- The two machines permutation flowshop. We have two machines M_1 and M_2 organized as a flowshop. A set of n jobs have to be scheduled into a sequence. Each job j has a value a_j and b_j that correspond to the time required to complete the job j on M_1 and M_2 respectively.
- The grouping of jobs to be delivered. When the jobs are completed for the two machines permutation flowshop we must form groups of jobs, where a group must contain at least one job. These groups are then used for the third subproblem.
- A traveling salesman problem. We consider one truck taking the group of jobs previously formed where each job has a destination k_j. The idea is to find the sequence of destinations in order to deliver every job from the group while minimizing the traveling distance. We have a matrix K $(m \times m)$ where m is the number of destinations and K_i^j is the distance between the destination i and j.

The objective function is to minimize the sum of lateness of the jobs $\sum_{j=1}^{j \leq n} L_j$. The lateness L_j of a job j is the difference between the completion time of the job c_j and its due date d_j: $L_j = c_j - d_j$. The completion time of a job is not equal to the completion time of the job for the flowshop problem but for the traveling salesman problem. c_j **is equal to the date at which the truck is coming back to the factory after its deliveries. We consider that the truck has no capacity limit, it can carry has many jobs has possible.**

The input data of our problem are the following :

- The number n of jobs,
- The values a_j, b_j and d_j of job j $(\forall j \in [1, n])$,
- The number of m destinations,
- The destination k_j of job j $(\forall j \in [1, n])$,
- The matrix K $(m \times m)$ of distances between each destination.

During the resolution of the problem we must find the sequence of jobs on the flowshop, the groups of jobs to be delivered and the sequence of delivery for each group.

We proposed a first way of encoding a solution : a table of size n containing the sequence of job for the flowshop sub-problem, a table of size $2n$ containing the groups and sequences for the travelling salesman sub-problem. The first cell of the second table contains the number of jobs for the first group, the following cells contain the number of jobs in the delivery sequence and so on. The figure 1.1 shows a solution for a simple instance of the problem with its encoding.

The sequence here is $\{1, 2, 3\}$ the first group G_1 of jobs is $G_1 = \{1, 2\}$ and 1 is delivered first, the second group G_2 only contains one job $G_2 = \{3\}$.

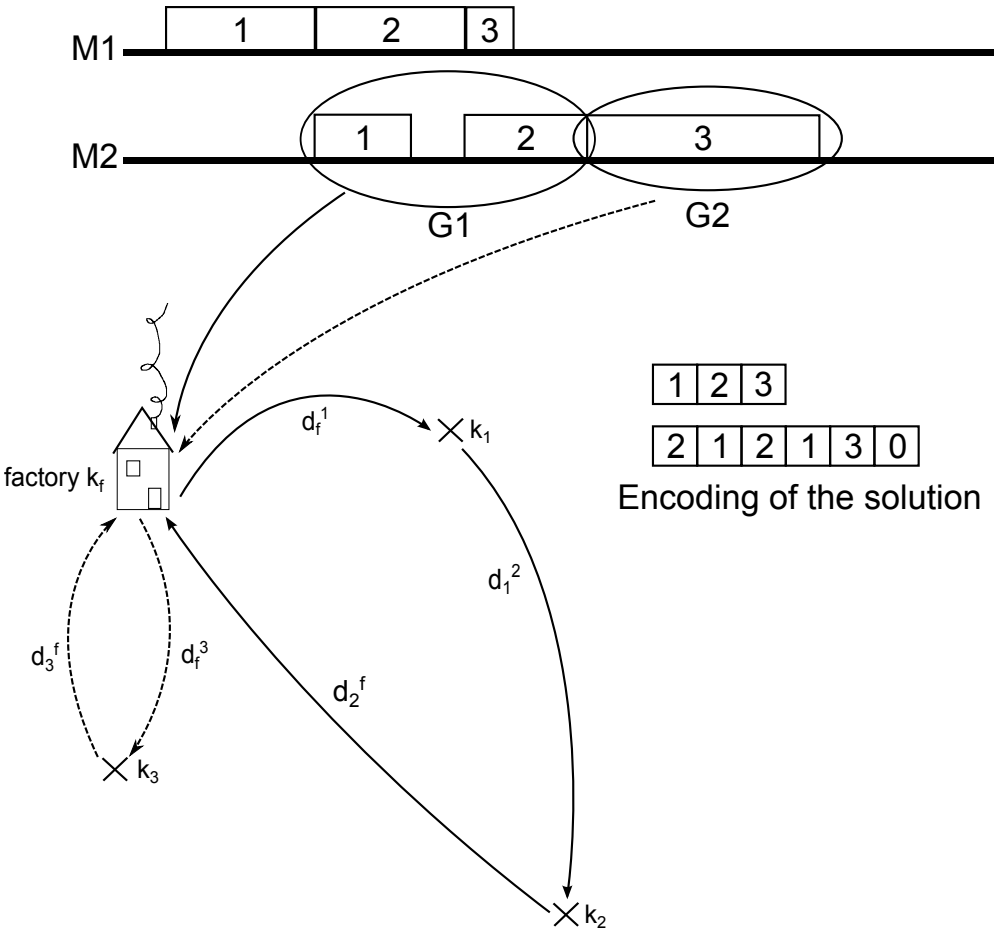

Figure 1.1: A solution of the problem for a simple instance

Ant colony optimization

2.1 General definitions

In this first part we will consider a travelling salesman problem in order to explain the mechanism of the ant colony metaheuristic. The graph representation of cities is an intuitive approach for such a method.

The classic algorithm of an ant colony heuristic is the following :

Algorithm 1 Ant colony

```
for k ← 1 to iterationsNumber do
    for each ant do
        Construct solution
        Local pheromone update
    end for
    Local search
    Global pheromone update
end for
return Best found solution
```

In order to construct the solution we need a probability formula based on the following :

$$P^m ij = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum\limits_{l \notin tabu_m} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta} & \text{if } j \notin tabu_m \\ 0 & \text{if } j \in tabu_m \end{cases}$$

Where P_{ij}^m is the probability of an ant m in city i to go to the city j, τij the intensity of the pheromone trail from i to j, η_{ij} the visibility of city j from city i (this is typically $1/d_{ij}$ where d_{ij} is the distance between i and j). α and β are parameters used to make the pheromone trail or the visibility more important from one another. And finally, the tabu list is used to prevent ants from going to forbidden cities like the city they come from.

The local pheromone update is used to emulate the evaporation phenomenon of pheromone. This evaporation is calculated from a formula based on the following :

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_{min}$$

Where ρ is the evaporation rate and τ_{min} the minimum value of the pheromone trail, which is also usually used as initial value. **It is important to consider that the evaporation function is only used for the paths the ants travelled through.**

The global pheromone update is used to emulate the deposit of pheromone from the ants. This deposit is calculated from a formula based on the following :

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}$$

Where $\Delta\tau_{ij}$ can be defined as follows :

$$\Delta\tau_{ij} = \begin{cases} 15 & \text{if } (i,j) \text{ is an edge in the best and second best,} \\ 10 & \text{if } (i,j) \text{ is an edge only in the best,} \\ 5 & \text{if } (i,j) \text{ is an edge only in the second best} \end{cases}$$

2.2 Application to the combined routing and scheduling problem

One main issue to overcome here is the fact that there is no simple graph representation of the complete problem. For the travelling salesman problem, it is easy to see how the pheromone trails and the visibility comes from. For our problem it is quite different. We can try to imagine a graph that an ant could visit and form a complete solution from it. For instance, a node could represent the fact that a job j is placed first in the flowshop, alone in a delivery group and delivered first. But we then have to consider every possibility knowing that the grouping and delivering problems are dependant on the flowshop. If we change the sequence of jobs in the flowshop we obtain a different grouping problem and then a different travelling salesman problem.

Even though we happen to be able to model such graph, its size would be unreasonably huge. It is not rational to chose such solution. **We must abandon the idea to apply the ant colony optimization all at once for our problem, we need to think of a different approach.**

Instead of trying to use the ant colony metaheuristic for the complete problem, we can consider using it for parts of it. **We can chose to use the ant colony on one or several sub-problem and use different method to construct the rest of the solution and then update the pheromone trails based on the results of the whole solution.**

Although is not clever to consider using the ant colony for the travelling salesman sub-problem since the problem is different each time you change the groups to be delivered or the sequence of jobs from the flowshop. The pheromone trails would be lost each time you change the solution in any sub-problem above.

We can still consider the other two sub-problems as potential candidate for the ant colony. Indeed, the flowshop problem and the grouping problem can both be considered central in our global problem. **We then have three possibilities, use the ant colony on the flowshop, on the grouping problem or both and use simple heuristics coupled with a surrogate mechanism to obtain the value of the objective function for our solution in order to guide the ants towards the best solution.** The surrogate mechanism could be very useful in order to save time computation, it would be used in the majority of cases and replaced with heuristics in the others in order to verify if the current solution is still a good one.

2.2.1 Definition of the Ant colony to the combined problem

Here we decided to use the ant colony metaheuristic on the grouping problem. It is easy to find heuristics to construct the rest of the solution when we have the groups of jobs. We have to specify how to implement the ant colony algorithm to our sub-problem. This typical visualisation we use for an ant building a solution of the travelling salesman problem is a complete graph. The nodes of the graph are cities, when the ant goes from city a to city b is means we have the sequence ab in our solution. The ant has to visit all nodes with a specific path. It chooses a path from the probability formula defined before.

We use a different construction of solution for our algorithm. We take the jobs as the nodes of the graph and two edges (a green and a red) connect each node together. The graph has then n nodes and

$n(n+1)$ edges. During the construction of a solution, and ant can come from a node a to a node b from two different edges: If it takes the green edge it means node a and b belong to the same group, if it takes the red edge it means the two nodes don't belong to the same group. The figure 2.1 shows how to construct the groups from the path travelled by an ant on our graph.

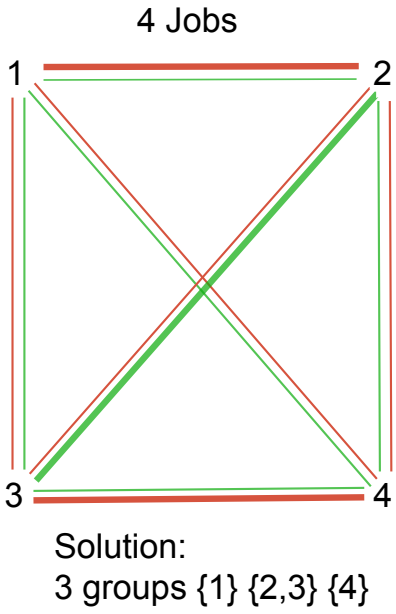

Figure 2.1: The construction of a solution by an ant.

We know how the ants construct the solution and how the graph is made. What we need is to define the visibility criterion for each arc of our graph. Let's define the visibility for the green edges, what is the formula that will make the ants form groups of jobs? We are handling a complex problem, we need to take into account the two other sub-problems: the flowshop, and the travelling salesman.

For the travelling salesman what we wish to do is group the jobs is their distance is small: $\frac{1}{\delta_{ij}}$ where δ_{ij} is the distance between the destination of job i and the destination of job j.

For the deliveries we know that we have to start from the factory and to end at the factory, the distance between the destinations of the jobs and the factory matters, when the jobs are far from the factory we tend to form groups: $\frac{\delta_{0i} \cdot \delta_{0j}}{\delta_{max}}$ where δ_{0i} is the distance between the factory and the destination of job i and δ_{max} is the maximum distance between the factory and any other destination. We divide by the maximum value in order to keep a value between 0 and 1.

We can then combine those two criteria and add a tuning parameter ω to make one criterion or the other more important:

$$\left(\frac{1}{\delta_{ij}}\right)^{\omega} \cdot \frac{\delta_{0i} \cdot \delta_{0j}}{\delta_{max}}$$

With this criterion, we will form groups of jobs the further their destinations are from the factory and the closer their destinations are from one another.

For the flowshop sub-problem we want to consider the due dates of the jobs. **When the due dates of two jobs are close to each other we will tend to put them into the same group**:

$$\frac{1}{|d_i - d_j|}$$

We can combine the criterion from both sub-problem, add tuning parameters and get the formula of the visibility η_{ij}:

$$\eta_{ij} = \left(\frac{1}{|d_i - d_j|}\right)^{\psi} \cdot \left[\left(\frac{1}{\delta_{ij}}\right)^{\omega} \cdot \frac{\delta_{0i} \cdot \delta_{0j}}{\delta_{max}}\right]^{\chi}, \text{ where } |d_i - d_j| \neq 0 \text{ and } \delta_{ij} \neq 0$$

We said before that this visibility was to be considered for the green edges of our graph. The visibility η'_{ij} for the red edges take the inverse criteria of the visibility of the green edges:

$$\eta'_{ij} = (|d_i - d_j|)^{\psi} \cdot \left[(\delta_{ij})^{\omega} \cdot \frac{\delta_{max}}{\delta_{0i} \cdot \delta_{0j}}\right]^{\chi}, \text{ where } \delta_{0i} \cdot \delta_{0j} \neq 0$$

Explain here the point of the roulette and who it is defined

2.2.2 Algorithms

Let us call our search graph G, the node set of this graph $\mathscr{N}$, the green edge set $\mathscr{G}$ and the red edge set $\mathscr{R}$.

Algorithm 2 Ant colony for the mixed optimisation problem

Initialize pheromone matrix with τ_0
Calculate the matrix of visibility
List *path*
for $k \leftarrow 1$ to *iterationsNumber* **do**
    for each *ant* **do**
        / Construction of the solution for the grouping problem*/*
        Select random node u from $\mathscr{N}$
        $i \leftarrow 1$
        repeat
            Pick a random number $p \in \{0, 1\}$
            if $p < P_0$ **then**
                From all incident edges of u get v the edge with the maximum value of P_{uv}
            else
                Select edge v from roulette
            end if
            if $v \in \mathscr{G}$ **then**
                $path.push(v)$
            end if
            $i \leftarrow i + 1$
        until $i = n$
        Each community of nodes linked with edged from *path* form a group
        Each remaining nodes form an individual group containing only this node
        / Construction of the rest of the solution */*
        Use heuristics to construct the rest of the solution
        Computation of $\sum_i L_i$
        if the new solution is better than the best known solution **then**
            Save the new solution as the new best known solution
        end if
        Local pheromone update
    end for
    Local search
    Global pheromone update
end for
return Best found solution

Ant colony optimization algorithm for a combined routing and scheduling problem

Computer Aided Decision Support
International Research Master 2
2013 - 2014

Operational research project draft

Abstract: abstract

Keywords: keywords

Supervisors
Jean-Charles BILLAUT
jean-charles.billaut@univ-tours.fr
Nicolas MONMARCHÉ
nicolas.monmarche@univ-tours.fr

Student
Thomas NOGUER
thomas.noguer@etu.univ-tours.fr

M2RI CADS 2013 - 2014

Université François-Rabelais, Tours