Université de Genève

Metaheuristics for optimization

TP 2: The Quadratic Assignement Problem

Author: Joao Filipe Costa da Quinta

E-mail: Joao.Costa@etu.unige.ch

October 17, 2021

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Département d'informatique

1 Introduction

The Quadratic Assignment Problem or (QAP) is a combinatorial optimization problem, which we will tackle during this TP. QAP is best described by the problem of assigning a set of facilities to a set of locations, where locations have a given distance from each other, and facilities a given flow. For example, it would be smart to have a car factory as close to the dealership as possible, so that the distance to deliver the cars manufactured is minimized. This is a NP-hard problem.

We must note that the number of facilities is the same as the number of locations. For this TP we will set n=12 the number of locations/facilities. and two matrices $D = [d_{rs}]$ where d_{rs} is the distance between locations r and s, as well as $W = [w_{ij}]$ where w_{ij} is the flow between facilities i and j.

Let $S(n)$ be the search space, which is the set of all permutations of size n! in our case 12!. We are looking for $v \in S(n)$ that minimizes a given fitness function. v is of size n, and vi corresponds to the location of facility i in the current solution $v \in S(n)$.

$$fitness\_function : I(v) = \sum_{i,j=1}^{n} w_{ij} \times d_{vi,vj}$$

2 Tabu Search

As we saw in the previous TP, with a deterministic hill climbing algorithm, we would easily get stuck in a local maximum. Tabu search (TS) has an approach that is quite like the deterministic hill climb, but when we are at a maximum, we don't stop, we keep going, and thanks to TS we are sure that we don't go back the mountain the same way. Basically, TS prevents us from choosing the same transition that we choose L moves ago. For example, let's assume L=1, if we are at an initial state s_0 and find a transition t_1 that brings us to s_1, then TS prevents us from going back to s_0 with t_2 for at least L=1 moves.

During deterministic hill climbing method we always choose the best neighbour of our state s, in TS we choose the best neighbour that is not forbidden by TS.

Like in the deterministic hill climb, we start by generating a random state s from the search space $s \in S(n)$, then we have to compute its fitness by using the formula described in the introduction section, $I(n) = fitness$.

Now in the deterministic hill climb we would simple compute $N(s)$ the neighbourhood of the state s and choose the highest fitted neighbour. However, because we are using the TS method, we have to take into account the forbidden transitions. Let TS_set be the set that contains all the forbidden transitions, we now compute all possible transitions from state s let's call it $T(s)$, now we simply compute the legal transition set $T^*(s)$ as follows:

$$T^*(s) = T(n) \setminus TS\_set$$

From $T^*(s)$ we easily compute all the legal neighbours $N^*(s)$, then just like in the deterministic hill climbing method we choose the neighbour that has the smallest/highest fitness depending on the problem.

After choosing a neighbour we had the respective transition to the Tabu List for a duratin of L moves/iterations. Tabu List is also known as a short term memory, as the last transitions choosen by the algorithm can't be used again in the near future.

The goal during this TP is to use TS method to solve QAP.

3 TS for QAP

Let's see how we will use TS to solve QAP.