Excercise 4
Implementing a centralized agent

Group №: 272257, 272709

November 3, 2020

1 Solution Representation

We formalize the pickup and delivery problem as a *constraint optimization problem*. The *COP* is formally a tuple $< X, D, C, f >$ where X is a set of variables describing a plan, D the domain of the variables $x_i \in X$, C a set of constraints that a plan needs to fulfill and f a cost function that we are trying to minimize. A plan P is given by an assignment of the variables in X (where the constraints in C are satisfied). We call N_v the number of vehicles and N_t the number of tasks.

1.1 Variables

Any plan can be described by a given assignment of the 4 following variables:

1. **nextTask_v** : the *nextTask_v* variable is an array of size N_v where each element $nextTask\_v[i]$ represents the first task that the i^{th} vehicle will perform.

   if $nextTask\_v[i] = j$ it means that the i^{th} vehicle will start it's route by delivering the j^{th} task

   if $nextTask\_v[i] = NULL$ it means that the i^{th} vehicle has no task to deliver in the given plan.

2. **nextTask_t** : the *nextTask_t* variable is an array of size N_t where each element $nextTask\_t[i]$ represents the next task that the vehicle that the vehicle will pickup

   if $nextTask\_t[i] = j$ it means that the vehicle that delivered the i^{th} task will deliver the j^{th} task next.

   if $nextTask\_t[i] = NULL$ it means that the vehicle that delivered the i^{th} has no task to deliver next.

3. **time** : the *time* variable is an array of size N_t where each element $time[i]$ represents the position of the i^{th} task in the plan of the vehicle delivering it in the plan (so if it is the first task delivered by some vehicle we would have $time[i] = 1$)

4. **vehicle** : the *vehicle* variable is an array of size N_t where each elements $vehicle[i]$ describes which vehicle will delivered the i^{th} task

1.2 Constraints

Not all possible variable assignments correspond to a valid plan, a valid plan is a plan that satisfies all constraints $c \in C$, the constraints are the following:

The next task after a given task t cannot be itself	$nextTask\_t[t] \neq t$
The time variable must be coherent	$nextTask\_v[i] = j \Rightarrow time(j) = 1$
	$nextTask\_t[i] = j \Rightarrow time(j) = time(i) + 1$
All tasks must be delivered	$\#$ *NULL values in nextTask_t*
	$=$ $\#$ *non-NULL values in nextTask_v*
From the definition of vehicle	$nextTask\_v(k) = j \Rightarrow vehicle(j) = k$
From the definitions of nextTask_t and vehicle	$nextTask\_t[i] = j \Rightarrow vehicle[j] = vehicle[i]$
From the definitions of nextTask_v and vehicle	$nextTask\_v[i] = j \Rightarrow vehicle[j] = vehicle[i]$
A vehicle cannot carry more than it's capacity	$load(i) > capacity(k) \Rightarrow vehicle(i) \neq k$

1.3 Cost function

Since we do not simply try to solve the constraint satisfaction problem but are actually looking for an optimal solution (in the sense that it minimizes a cost function) we need to define our cost function.

Given :

Distance between two cities c_1 and c_2:	$dist(c_1, c_2)$
Cost per kilometer for a given vehicle:	$C_{km}(v)$

We define the cost function $cost(P)$ as :

$$cost(P) = \overbrace{\sum_{v \in [1 \ldots N_v]} \left[\mathcal{C}_{km}(v) \cdot \sum_{path} dist(c_1, c_2) \right]}^{\textit{where } c_1 \textit{ and } c_2 \textit{ are two cites on vehicle } v\textit{'s path}}$$

2 Stochastic optimization

Because of the high computational complexity of the problem we search for a solution using a *Stochastic Local Search method (SLS)* that can be described by the following algorithm :

2.1 Stochastic optimization algorithm

Algorithm 1: Stochastic Local Search for Constraint Optimization Problem

S $\leftarrow$ GenerateInitialSolution($< X, D, C, f >$)
repeat
  $A_{old} \leftarrow A$
  $N \leftarrow Succ(A_{old}, < X, D, C, f >)$
  $A \leftarrow Choice(N, f)$
until ***stable_solution*** $\vee$ ***timeout***;
return A

2.2 Initial solution

2.3 Generating neighbors

3 Results

3.1 Experiment 1: Model parameters

3.1.1 Setting

3.1.2 Observations

3.2 Experiment 2: Different configurations

3.2.1 Setting

3.2.2 Observations