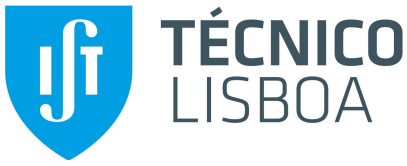

Search and Planning

Alameda Campus

Project (2018/2019)

Group number: 5

Name: Ana Sofia Aparicio Number: 81105

Name: Rodrigo Lousada Number: 81115

Grading:

Sum of the hours spent exclusively for developing this work: ~35h

Introduction

The problem present is known as Dial-a-Ride Problem. On our case, the problem consisted on obtaining a complete affection of the shifts of driving transport vehicles, to be done by the workers on a day of work. This way, plan the shifts of the workers, under a set of constraints.

The solution to this type of problems was stablished to be NP-hard. For this type of problems systematic search strategies will not achieved good results.

This study will be focused on systematic search applied with a simple heuristic. Testing if how it will perform on a small problem and the solution achieved, if any will be provided. We will also implement an iterative sampling strategy as well as an improved limited discrepancy search.

Problem modeling

The problem was modeled considering a structure shift that has two lists of lists guarding trips.

The restrictions to this problem were:

- Every shift needs to end and start in location L1
- The max duration of a shift is 8h (480min), if a shift is less than 6h (360) it counts as having 6h nonetheless
- Each shift has one meal break of 40min
- The worker ca not take a meal while being transported
- The maximum time a worker can work before a meal is 4h (240)

Data Structures

The problem was modeled considering the following structures:

- **Structure Trip**

The structure to represent a trip is described below.

(defstruct trip

local_departure

local_arrival

instant_departure

instant_arrival)

Regarding a trip is necessary to store the information related to the local of departure (*local_departure*), local of arrival (*local_arrival*), and two integers: the instant of departure (*instant_departure*) and the instant of arrival (*instant_arrival*).

- **Structure Shifts**

To represent a set of shifts, we created the following structure.

(defstruct shifts

trips_list

non_allocated_trips

cost)

trips_list is a list of lists. These lists represent a shift, and inside each shift is a list of trips (several structures *trip*).

non_allocated_trips is a list of *trips.* They are the trips "waiting" to be allocated on a determined shift.

cost is sum of all the cost of the shifts stored in that particular structure.

Cutting Strategies

As it is possible to notice by the complexity of this problem, it is impossible to search the complete space in the particular timeframe. To solve this problem, we implemented a strategy to limit exceedingly the number of successors generated.

The Cut Strategy used was to sort the input list with all the trips by the instant of departure. By doings this we only have to add a new trip at the end of a shift, preventing the generation of successor were a new trip is inserted in the middle of beginning.

Strategies Tests

To test used was the test of the project guide (figure 2):

(faz-afectacao '((L2 L1 1 25) (L1 L2 34 60) (L5 L1 408 447) (L1 L1 448 551) (L1 L1 474 565)) algorithm)

Results of the test

Algorithm	Time-elapsed	Expanded Nodes	Generated Nodes
A*.melhor.heu	0,009309 s	11	**16**
A*.melhor.heu.alterna	0,007597 s	10	**15**
sondagem.iterativa	0,004597 s	12	**16**
ILDS	0,006597 s	12	16

The approach that presented better results was the A* algorithm with an improved heuristic. ILDS and Iterative Sampling a quicker than the other two, but the end generating and expanding more nodes.

A*

The heuristic chosen to implement this algorithm was a heuristic that would sum the time (in minutes) spent on all the shift present in that state.

A* Alternative Heuristic

For the alternative heuristic we chose to consider the following costs:

- The time wasted on a shift
- The quantity of shifts

Then we summed these two values, giving more priority to the quantity of shifts.

Critical Analysis and Remarks

First, we concluded that some algorithms are no the best to apply to this algorithm, such has the ILDS and the Iterative Sampling. They end being faster, but no always presenting the best results.

Second, our heuristics might not be the best, however the lack of information regarding what could be the best heuristics in a Dial-a-Ride Problem and considering the difficulty of the problem were factors that conducted to the choice of our heuristics.

There was a possible improvement regarding the generation of successors. To make the algorithms quicker we could consider making a binary tree with only the two nodes that had the best cost cost function in each level.