Università
della
Svizzera
italiana

Facoltà
di scienze
informatiche

Travel Salesman Problem

Ant Colony System

Artificial Intelligence Cup 2016

Author:
Francesco Saverio
Zuppichini

Professor:
Luca Maria
Gambardella

November 13, 2016

1 Problem

The Problem know as the Traveling Salesman Problem, or TSP, consists in, given a finite set of cities, find the best tour such as each city is visited only one time. Each problem instance consist in a set of cities with the relative position in a Cartesian place.

2 Implementation

2.1 Choice

Antz *Antz*[1] was one of my favourite movies when i was i child, i always like how the films based on animals, in their simplicity, can reflect our complex society problems. In *Antz* the nest is separated into two categories, the *Workers* and the *Soldiers*, in which, the main protagonist, a low level worker, manage to break this strong division.

[1] http://www.imdb.com/title/tt0120587/

Nature Inspiration Personally, i really like our science take advantages and inspiration from nature. Let's just think about our the birds, for example, help the man "learn" how to fly.

Performance I, also, select this algorithm since it is one of the better algorithm to compute our task.

2.2 Algorithm

A deep explanation of the algorithm can be found here[2], basically n ants, placed in random cities, deposit a certain amount of pheromone and select the new route according to a given probability formula.

2.3 Optimizations

Math.pow: The first thing that comes at my mind when i was starting coding was to find a way to make java Math.pow faster. After some "googling" i found out a formula[3] that approximate the real power, and, quoting the author, it is *"23 times as fast as Math.pow()"*.

Local Search: In order to compute better results i'm running a 2-opt after each ant has constructed his local solution.

Probability Matrix: Most of the time spent by the ants is on computing the given formula in order to select the next city, even if an other ant has already computed the same formula before. For this reason i've implemented a matrix that is shared among all the ants instances and cached the previous computation. I've run some quick benchmark on the *ch130* with 2000 iterations and no local search managing to save the 98% of the computation.

Inverse Distance Matrix: The distance matrix is heavily used by all ants in order to get the inverse of an edge cost, so i decided to create another probability that just stored the inverse of each entries in the distance matrix in order to save the inverse computation.

[2]http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.9245rep=rep1type=pdf
[3]http://martin.ankerl.com/?s=pow

Memory: The idea is let a certain amount of ants to have a memory of the last best solution. Precisely, with some lucky, an ant can inherit a random subset from the best ant's tour so far.

Candidate List: The ants usually follow the *exploitation* phase, thus i decided to added a candidate list of size $i = N, j = 4 + N/4,$, where N is the problem size, made of the cheapest n nodes in the i row. If a solution is not found in this list, then the normal procedure will be triggered.

3 System Specification

3.1 Machine

All the problem instances were run with the MacBookPro given by the university with the following hardware and software specification.

System Version:	OS X 10.11.6 (15G1004)
Kernel Version:	Darwin 15.6.0
Boot Volume:	Macintosh HD
Boot Mode:	Normal
Computer Name:	fugace frammento informatico
User Name:	Francesco (VaeVictis)
Secure Virtual Memory:	Enabled
System Integrity Protection:	Enabled
Time since boot:	1 day 8:11

3.2 Compiler

javac 1.8.0_77

4 Run the Program

4.1 Run the code

In order to run the program you need to use the following command line:

```
$ java -jar  AI_Cup_JAVA.jar

filename=<filename> seed=<seed>
```

For your convenience i provide the following running example:

```
$ java -jar AI_Cup_JAVA.jar filename="./problems/rat783.tsp" see
```

4.2 Ant Colony System parameters

I do not allow any ACS parameter to be set, these are the default and optimal parameters:

$$\alpha = 1,\ \beta = 5,\ ants = 10, \rho = 0.1$$

5 Results

5.1 Final Report Table:

You can find table with the results and the seeds in:

./AI_cup_2016_ZuppichiniFrancescoSaverio_results

5.2 Reports:

You can find all the reports into the folder *./reports*, be aware that we start at index 0, therefore city 1 is showed as city 0. I'm sorry about that, since i'm running out of time i've fixed the output in the console, but not the one in the reports.

6 Conclusions

This project was quite challenging and funny at the same, in the end i manage to get a relative global error of only 0.24% that is a really good result. The most interesting part was trying to optimizing. I started with 83 iterations per second in the *ch130.tsp* with the given configuration and i manage to archive a maximum number of 460 iteration per second.

References

[1] Marco Dorigo, Luca Maria Gambardella *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*,Université Libre de Bruxelles

[2] Marco Dorigo, Luca Maria Gambardella *Coupling Ant Colony System with Local Search*,Université Libre de Bruxelles,Ann ´ee acad ´emique 2014-2015