Università degli studi di Milano-Bicocca

Decision Models

Final Project

Q-ant & Q-gen

Authors:
Dario Bertazioli-847761-d.bertazioli@campus.unimib.it
Fabrizio D'Intinosante-838866-f.dintinosante@campus.unimib.it
Massimiliano Perletti-847548-m.perletti2@campus.unimib.it

June 16, 2019

Abstract

1 Introduction

The problem: the travelling salesman problem (TSP) is an algorithmic problem tasked with finding the shortest route between a set of points and locations that must be visited. In the problem statement, the points are the cities a salesperson might visit. The salesman's goal is to keep the distance travelled as low as possible. TSP has been studied for decades and several solutions have been theorized. The simplest solution is to try all possibilities, but this is also the most time consuming and expensive method. Many solutions use heuristics, which provides probability outcomes. It must be considered that the results are approximate and not always optimal.

Our approach: in this project we tried to apply two meta-heuristics named *Ant Colony Optimization* and *Genetic Algorithm*, implementing their "classical" version and a custom one integrating *Reinforcement Learning Algorithm*, namely *Q-learning*.

2 Datasets

The datasets used in this work are taken from `https://wwwproxy.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/`, a large source of TSP datasets largely cited in literature. There are datasets of variable dimension and for everyone is also available the optimal solution so that is possible for us to compare our results with the optimal one. Every solution is available at `https://wwwproxy.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/STSP.html`. Every dataset is composed by a list of *"cities"* with two coordinates points; the only preprocessing we applied was to compute a matrix containing the distance between every point and the other ones.

3 The Methodological Approach

3.1 Theoretical context

Genetic Algorithms: In this work we focus on two different approaches to the TSP. The former is the **Genetic Algorithm**, which we initially implement in its classical version. This algorithm is based on a biological metaphor: the resolution of a problem is seen as a competition among a population whose evolving individuals become better and better candidates solutions over time. A "fitness" function is used to evaluate each individual to decide whether it will contribute to the next generation. Then, in analogy with the biological metaphor (the gene transfer in sexual reproduction), a crossover operator is applied in order to generate the next generation of the population. This process, according to the evolutionary theory (Darwinism), should lead after a certain number of iterations to a much more fit ensemble of individuals representing "good" candidate solutions to the considered problem.

The pseudo code of the standard genetic algorithm is summarized in the Fig. 2, where Tc is the crossover rate or parameter that determines the rate at which the crossover operator is applied, Tm is the equivalent for the mutation rate, Tp is the population size (number of chromosomes) and MaxG the number of generations used in the experiment.

With the aim to explore a new variation of the standard algorithm, we try to integrate a **Q-Learning** Algorithm in the genetic procedure in order to provide a better guideline for the initialization of the population and the crossover operation.

Ant Colony Optimization: The latter kind of algorithm we implement is the **Ant Colony Optimization** algorithm

4 Results and Evaluation

The Results section is dedicated to presenting the actual results (i.e. measured and calculated quantities), not to discussing their meaning or interpretation. The results should be summarized using appropriate Tables and Figures (graphs or schematics). Every Figure and Table should have a legend that describes concisely what is contained or shown. Figure legends go below

Algorithm 1 Genetic Algorithm

procedure Genetic(Tc,Tm,Tp,MaxIt)
  $Pop \leftarrow GeneratePopulation(Tp)$
  $Pop \leftarrow Evaluation(Pop)$
  for $i = 1 \ldots MaxIt$ **do**
    $Pop \leftarrow Selection(Pop)$
    With probability Tc do:
    $Pop \leftarrow Crossover(Pop)$
    $Pop \leftarrow Selection(Pop)$
    With probability Tm do:
     $Pop \leftarrow Mutation(Pop)$
  end for
  return the best solution in Pop
end procedure

Table 1: Genetic Algorithm pseudocode

the figure, table legends above the table. Throughout the report, but especially in this section, pay attention to reporting numbers with an appropriate number of significant figures.

5 Discussion

The discussion section aims at interpreting the results in light of the project's objectives. The most important goal of this section is to interpret the results so that the reader is informed of the insight or answers that the results provide. This section should also present an evaluation of the particular approach taken by the group. For example: Based on the results, how could the experimental procedure be improved? What additional, future work may be warranted? What recommendations can be drawn?

6 Conclusions

Conclusions should summarize the central points made in the Discussion section, reinforcing for the reader the value and implications of the work. If the results were not definitive, specific future work that may be needed

Algorithm 2 Ant Colony Optimization

Main Algorithm

```
for generation in generations:
    create n_ants artificial ants
    for one_ant in ants:
        make_path
        compute_path_length
        update best_dist and best_path
    update pheromon matrix (local for child process)
return best_dist, best_sol
```

Make path

```
start from a vertex
add start vertex to visited nodes
for each remaining vertex:
    list the neighbors
    list the not yet visited neighb
    calculate the probability of choosing a vertex
    choice the vertex according to probability
    add the vertex to the passed list
    return the chosen vertex id
```

Update pheromon matrix

```
evaporate pheromon
for ant in ant_colony :
    for each vertex of one_ant_path :
        pheromon_matrix.increase(c_v, n_v,+1)
```

Update pheromon matrix

```
gather from MPI env all the pm matrix
if process is the parent process (rank==0):
  for each element average over the n_cores matrices.
broadcast obtained pm matrix to the child process
```

Table 2: Ant Colony pseudocode

can be (briefly) described. The conclusions should never contain “surprises”. Therefore, any conclusions should be based on observations and data already discussed. It is considered extremely bad form to introduce new data in the conclusions.

References

The references section should contain complete citations following standard form. The references should be numbered and listed in the order they were cited in the body of the report. In the text of the report, a particular reference can be cited by using a numerical number in brackets as [**?**] that corresponds to its number in the reference list. LaTeXprovides several styles to format the references