Traveling salesman problem

Bert Mortier
Toon Nolten

The problem

- There is an enormous amount of variability:
  - Several operators
  - A whole range of parameters for each of the operators
  - We need to make this a lot smaller
- Which combination of operators might perform well is not at all apparent
  - So we do not want to make any assumptions

First idea: brutest-bruter-…

- First use a coarse mesh to cover the parameter space:
  - For example crossover probabilities of 0, 0.2, 0.4, 0.6, 0.8
- Because of the large amount of parameters and slow simulation the total time rapidly becomes prohibitive
- Even if we succeed in solving this, then this would only be a first iteration
- We would have then have to interpret these results and set up a next experiment

First idea: brutest-bruter-…

- The reason we thought this is not optimal is because it would never allow us to search the entire parameter space efficiently
- To be feasible we would require a very crude mesh over the parameter space
- This means we would have to do many iterations, which means that we have to do many manual selections
- It is exactly those manual selections we wish to avoid, since it would mean we incorporate our prejudices, which might be blind for emergent phenomena between several operators

Meta GA

- The problem of a large parameter space with difficult relationships between different parameters seems very well suited for a GA approach
- So we decided to use a simple GA program to find good parameters for the TSP GA
- We now let the program decide which operators function well and which don't
- Since we are no longer needed as an interpretation step, we can use more complex situations, with for instance multiple allowing multiple crossover and mutation operators at the same time, without the problem of guessing when there are emergent features at work and when not

Meta GA: representation

- As the representation we use a string with all the different parameters we wish to tune
- These include:
  - The number of individuals
  - The chance for each crossover operator and mutation operator to be used
  - The selection procedure that is used
- This means our string contains integers together with real values and even function handles

Meta GA: algorithm

- But it is evident that this will have a tendency for a number of individuals that is as large as possible
- This gives the obvious disadvantage of a larger computational time
- Hence we need a ‘regularization’ term in the cost function

Meta GA: Regularization

- We should incorporate a cost for the run time of the simulation otherwise the algorithm would have a tendency towards higher
- Now we work with an explicit time cost (tic toc)
  - Time measurement is highly variable
- Better is to have something deterministic
  - E.g. punish large values for the number of individuals or large values for heavy operators
  - This gives us more parameter, exactly the issue we tried to solve by using a meta GA
  - But we can find these parameters exactly by measuring a mean time each of the steps cost

Future goals

- Island model
- Adaptive parameters over time