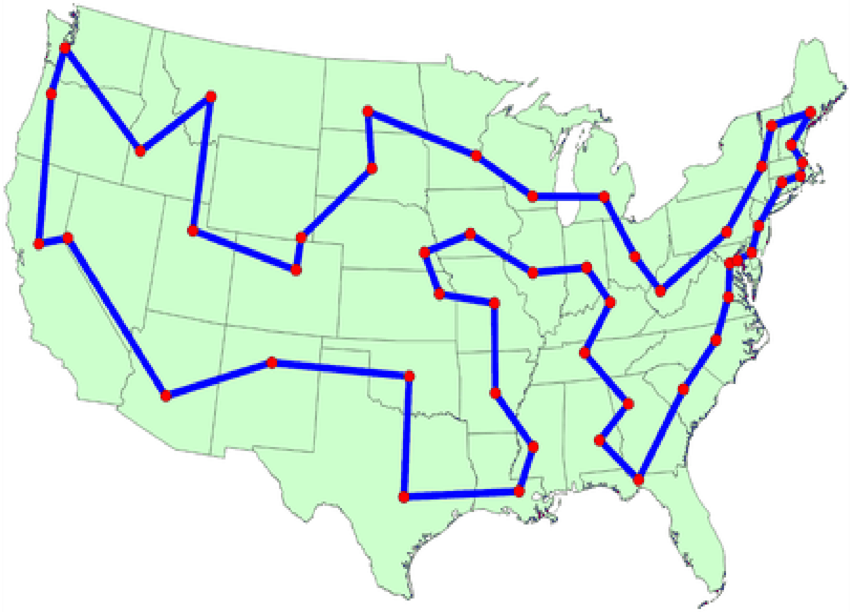
Solving mTSP using evolutionary algorithms

Semestral work presentation
EOA – Teymur Azayev

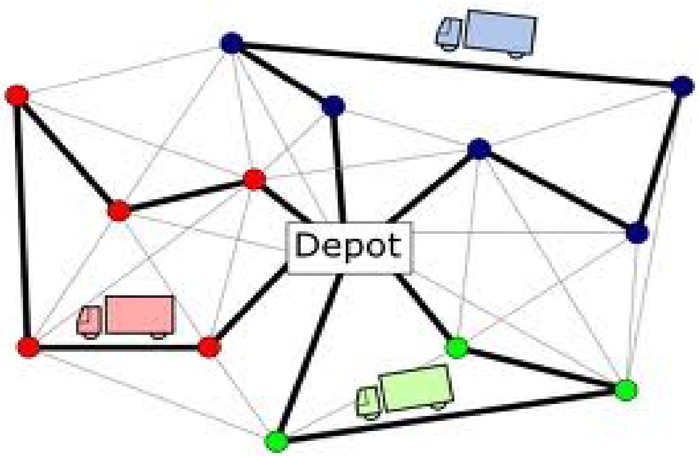
The TSP problem



Source: support.sas.com

Given n cities, find shortest tour.

mTSP problem



Source: rcasts.com

M – agents, 1 depot

Solving the TSP

- General TSP problem is NP-Hard
- Exact solution is not necessary, good solution is enough
- Use of evolutionary algorithms to solve difficult optimization problems

Problem representation

- Solution is represented as a sequence – bounds pair:

  seq : [2,5,1,6,2,8,4,3,10,12,7,11,9]

  bounds: [4,8]

  For n = 12 cities and m = 3 agents.

- The above means that agent 1 starts at the depot, goes through cities [2,5,1,6] and goes back to depot. Agent 2 starts at the depot,

  goes through cities [2,8,4,3] and then goes back to the depot, etc..

- Fitness: Length of longest tour from all the agents.

Local search

- Simple swapping of two cities n times. After n iterations the best swap is kept.
- Easy implementation, stuck at local minimum.

- Improvement: Perform n swaps every iteration, allow for worsening, keep best swap.

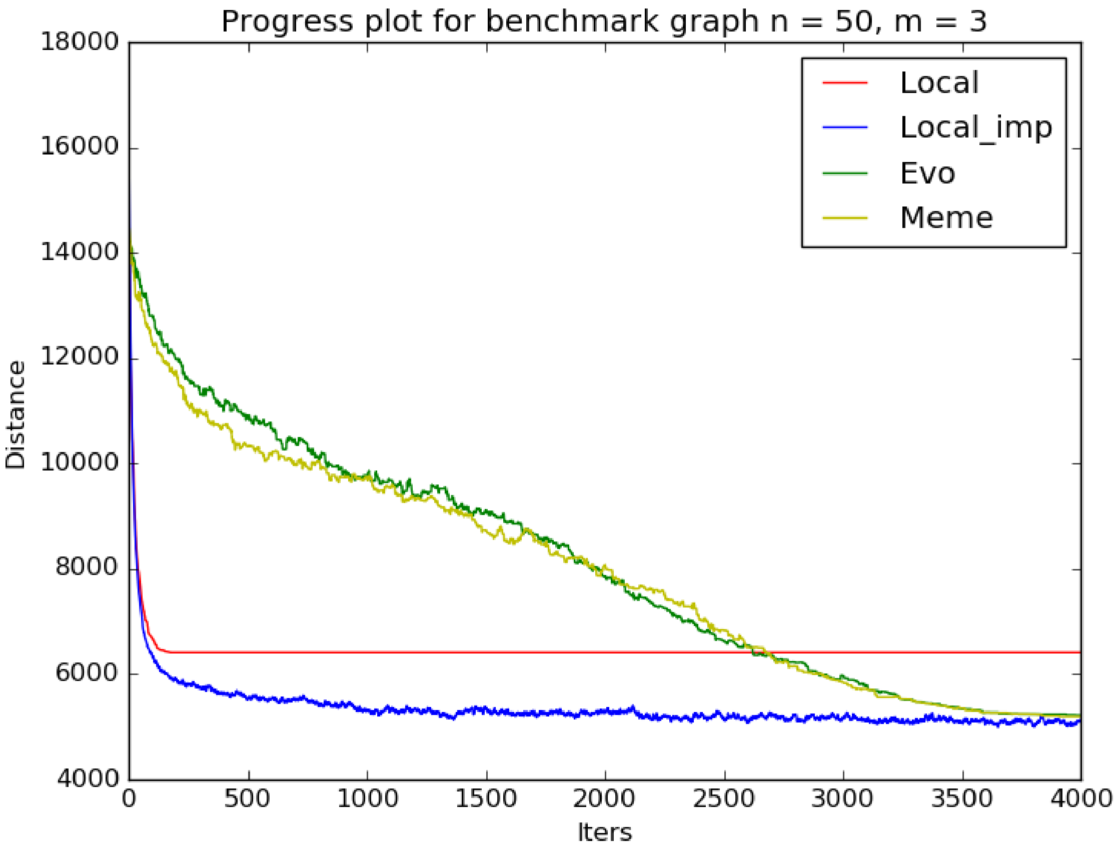
Evolutionary algorithm

- One point crossover (with slight modification to get valid sequence).
- Boundary mutation
- Roulette selection of p parents, new offspring always replaces parents if better, replace with a small chance if worse.

Memetic search

- Same crossover as in the evolutionary algorithm
- Addition of 2-opt after creation of new individuals.
- Gives chance to weak individuals to locally improve themselves and have higher chance to be selected in the next round.

Progress plot



Average results

	Local	Local imp	Evo	Meme
Longest tour	6411	5100	5161	5194

Conclusion

- Improved Local search (simulated annealing) outperformed EA.
- Works out of the box on small problems.
- EA very slow, inefficient for large problems, requires hyperparameter tuning and population analysis to prevent premature convergence and other bad effects.