FFI Forsvarets forskningsinstitutt

Solving impossible problems

With evolutionary computation

Sondre A. Engebråten – PhD Candidate, University of Oslo.
Researcher at the Norwegian Defence Research Establishment

What are evolutionary methods?

How do infants solve problems?

No background knowledge

No way of asking for help

New and unknown problem

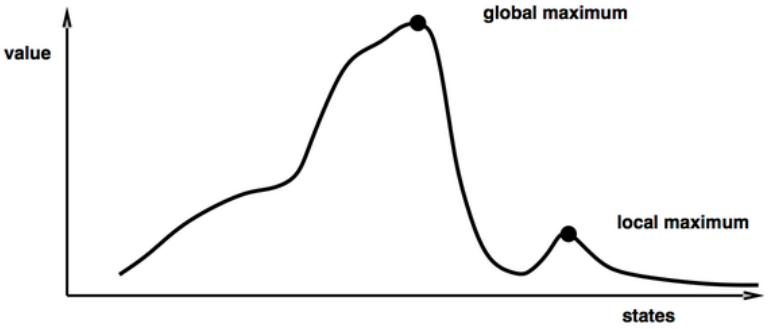
Trial and error problem solving

What is trial and error?

- Trying many solutions
- Keeping good solutions
- Discarding bad solution
- Keep tweaking good solutions

What is evolutionary computation?

- A way to solve complex problems
- A function optimizer
- A heuristic search method
- A hill climber optimization method

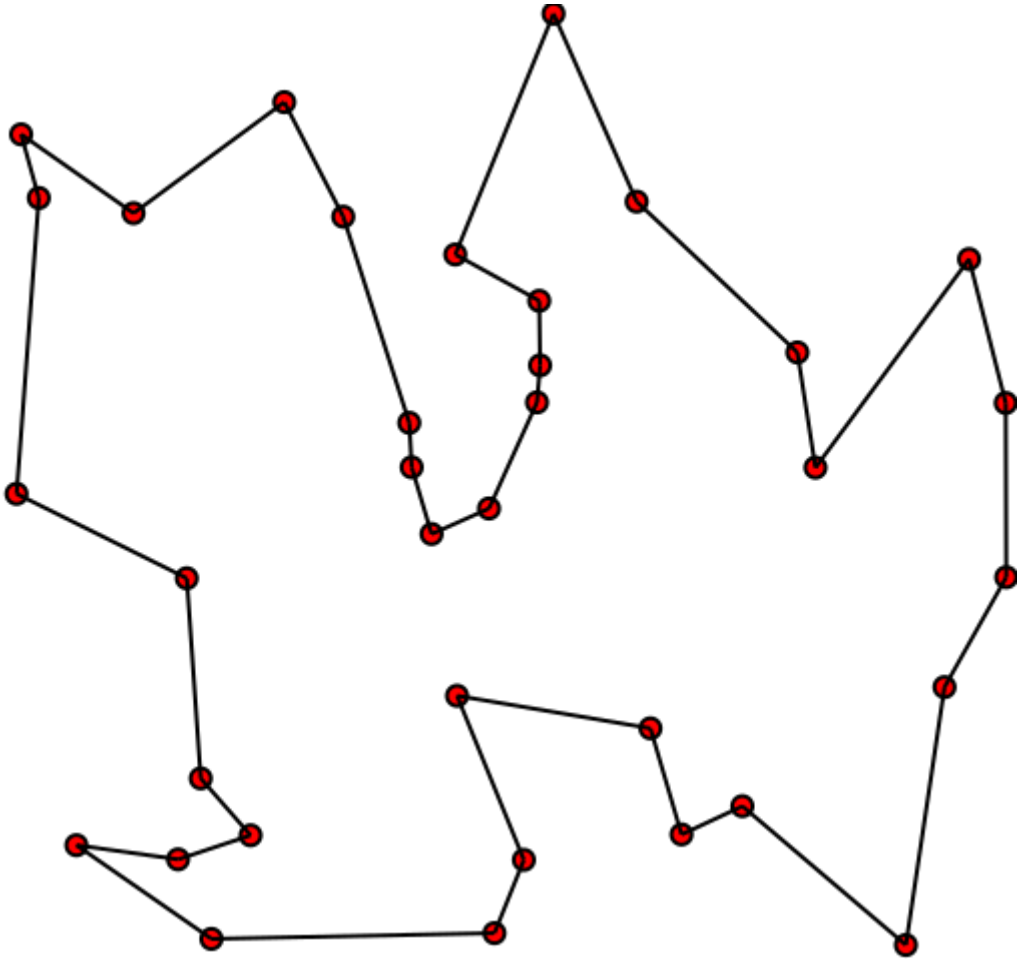


What is not evolutionary computation?

- A no prior information, ultimate solver for all problems
- A free lunch
- A way to evolve little creatures inside a computer

What has evolutionary computation been used for?

Travelling salesman problem TSP (NP-hard)



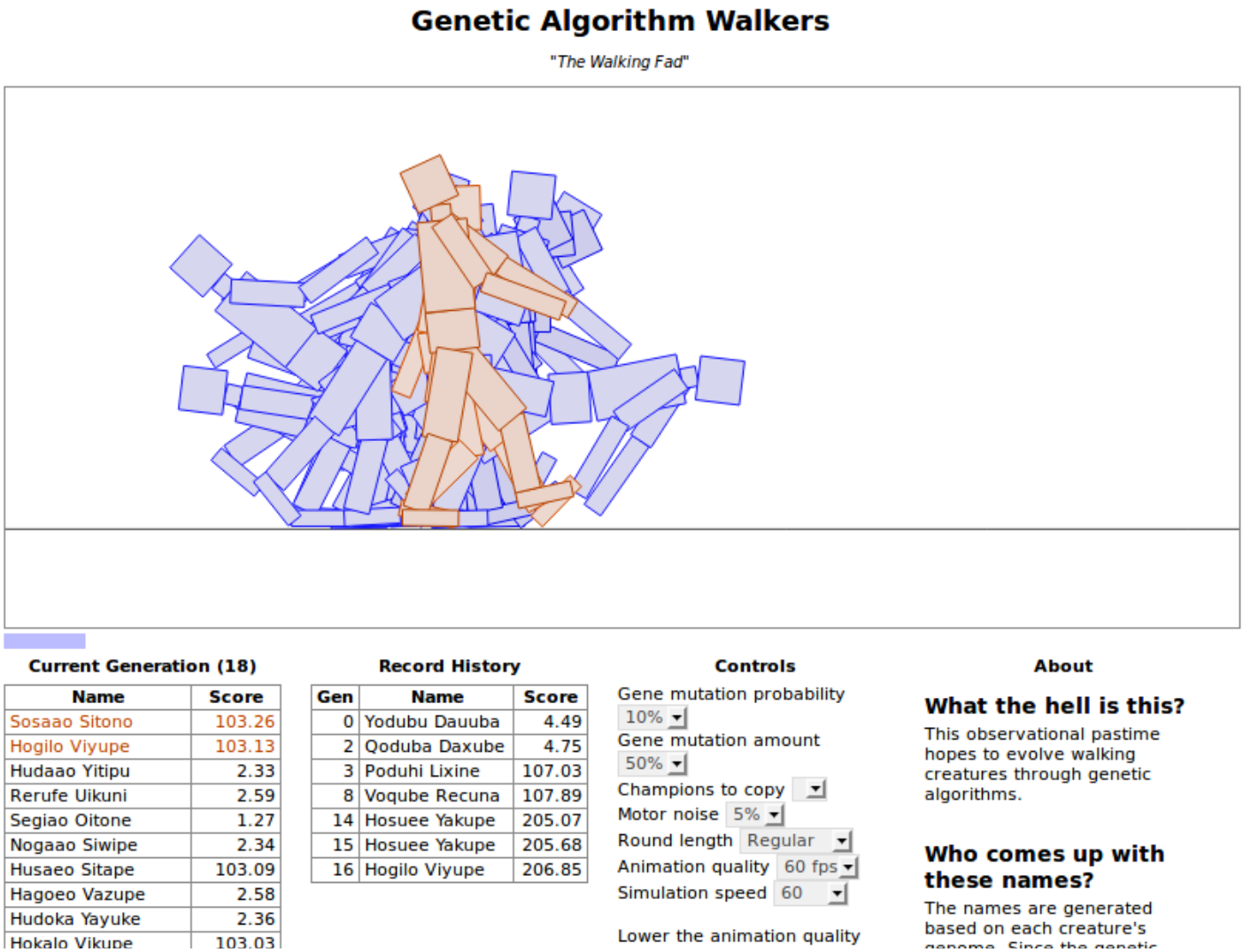
N = 35 N! = 10333 147966386 144929666 651337523 200000000

NASA evolved antenna

- Antenna design is hard
- Simulating antenna is also hard
- Describing how, or more importantly why antennas work is hard

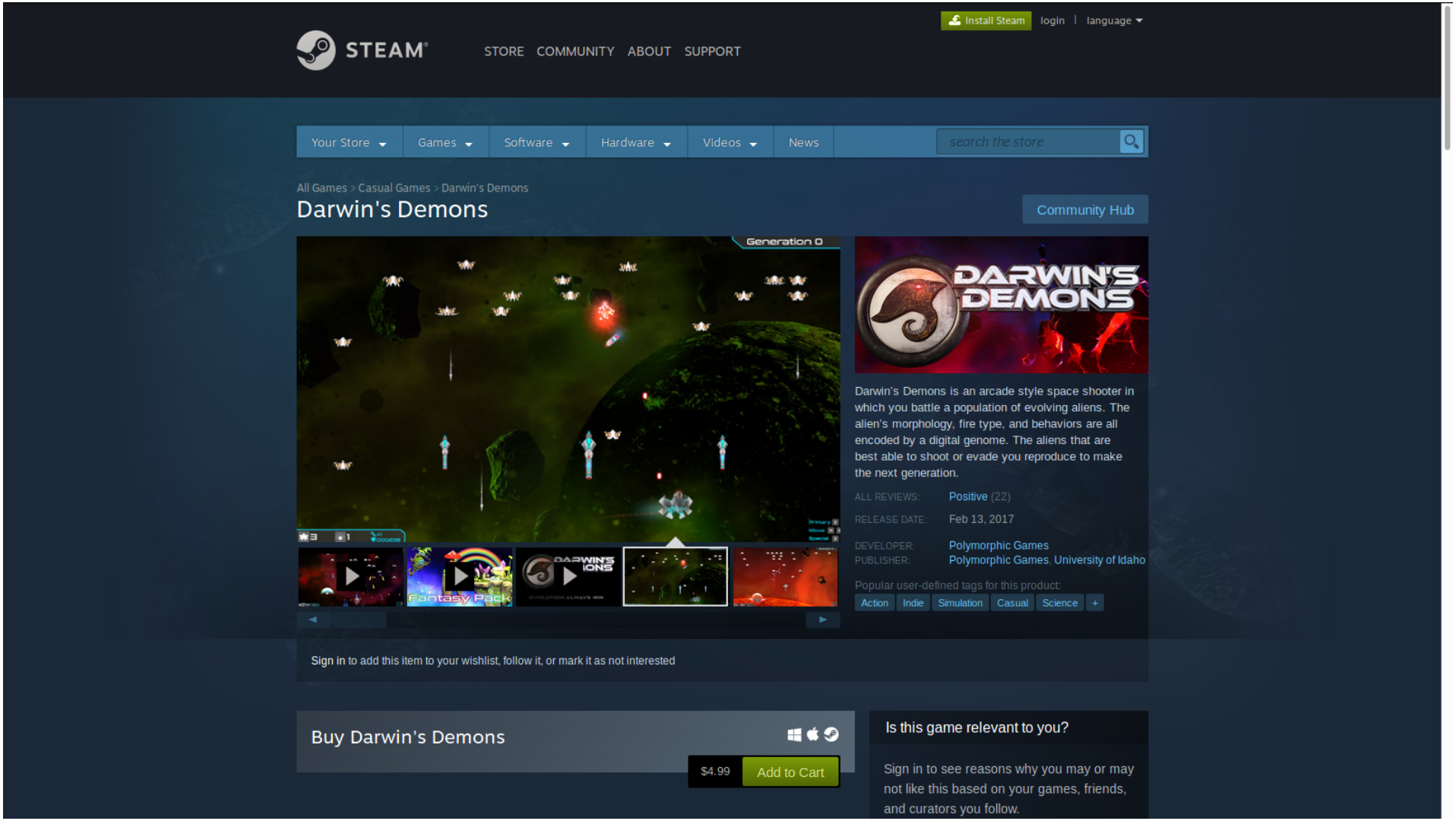
http://alglobus.net/NASAwork/papers/Space2006Antenna.pdf

Evolving locomotion patterns



Walkers: http://rednuht.org/genetic_walkers/ Cars: http://rednuht.org/genetic_cars_2/

Evolving game playing opponents

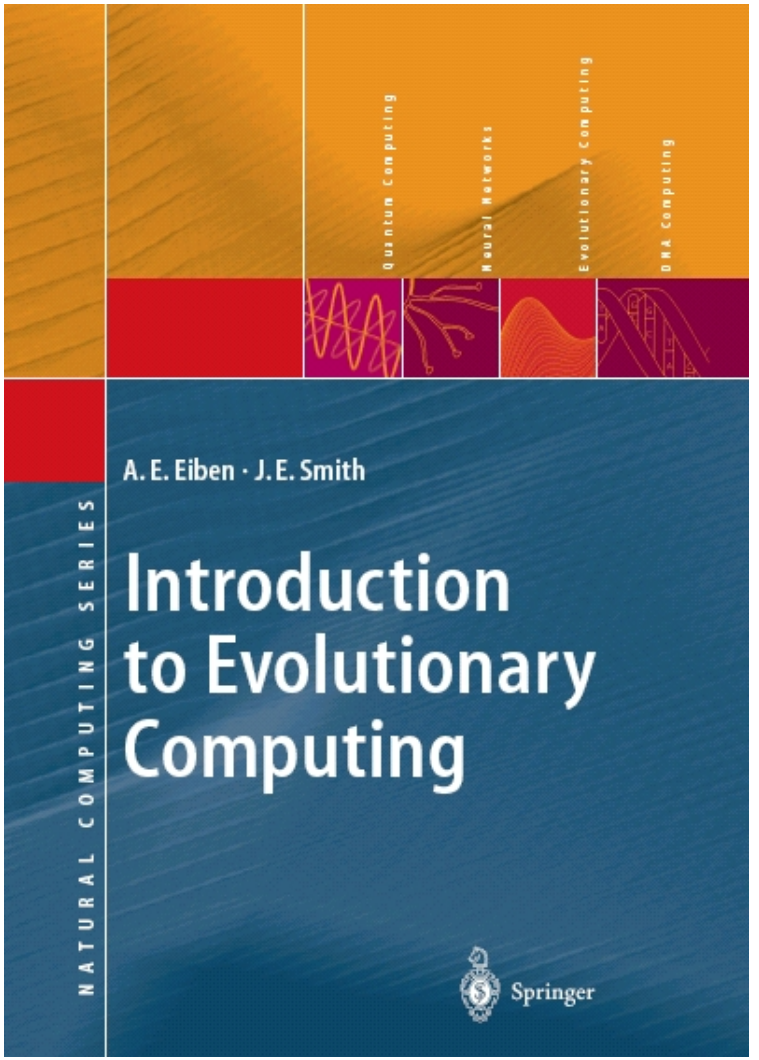


Paper: Darwin's Demons: Does Evolution Improve the Game?

Variations of evolutionary methods

- Classical taxonomy:
  - Genetic Algorithm (GA)
  - Evolutionary Strategies (ES)
  - Evolutionary Programming (EP)
  - Genetic Programming (GP)

- More recent work:
  - Divergent search
  - Novelty search



Genetic algorithm

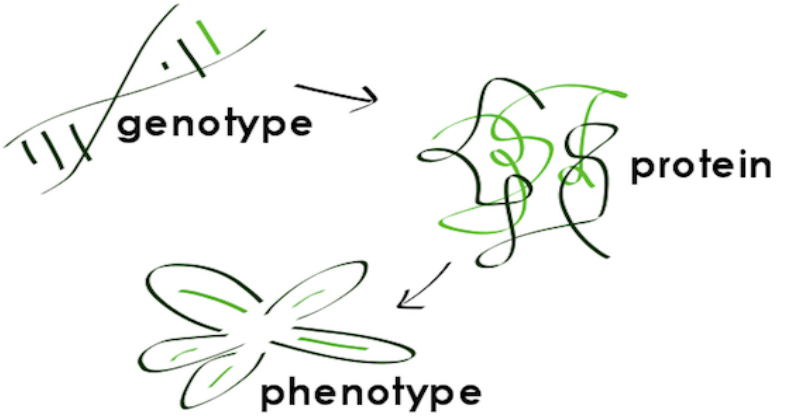
- Simulates evolution
- A population
- Full solutions
- Solution space
- Survival of the fittest
- Not guaranteed optimal
- General algorithm, not problem specific

Basic outline of a gentetic algorithm

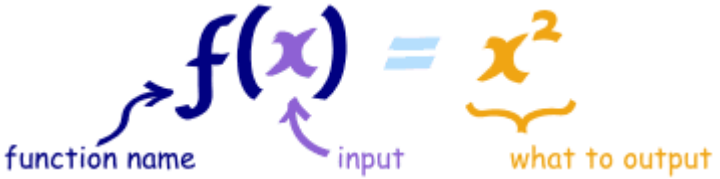
1) Create a random population of individuals
2) Calculate the fitness of the individuals in the population
3) Select some individuals in the population for mating; these are considered the parent.
4) Create a new set of individuals based on the ones previously selected (the children)
5) Mutate the children based on some (low) probability
6) Calculate the fitness of the new individuals and combine the children with the parent generation
7) If the desired performance of the population as a whole has been reached stop executing, else go back to step 3.

Adapting the genetic algorithm to a problem

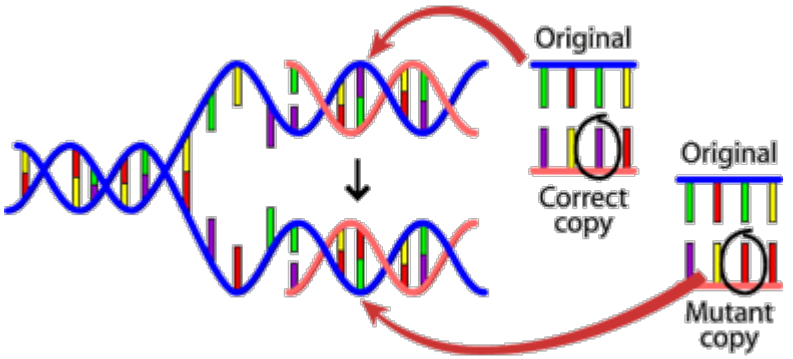
- Genotype and phenotype



- Fitness-function



- Crossover and mutation-operators
- Problem specifics



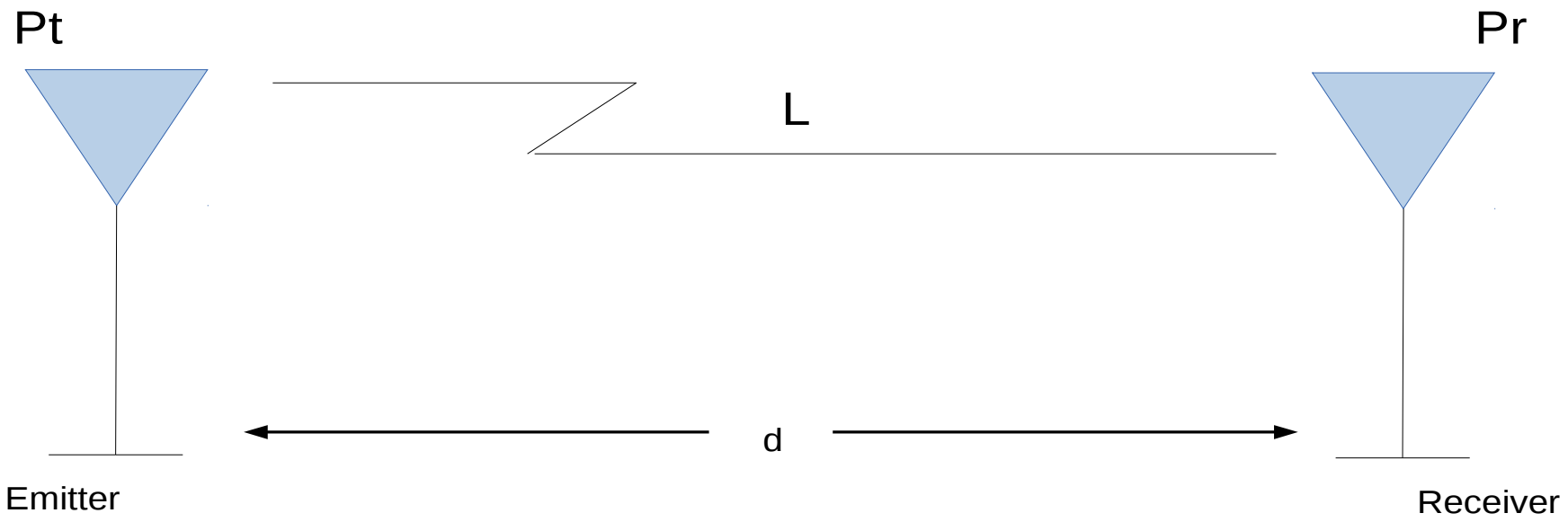
What have I used evolution for?

1) Path planning for UAVs

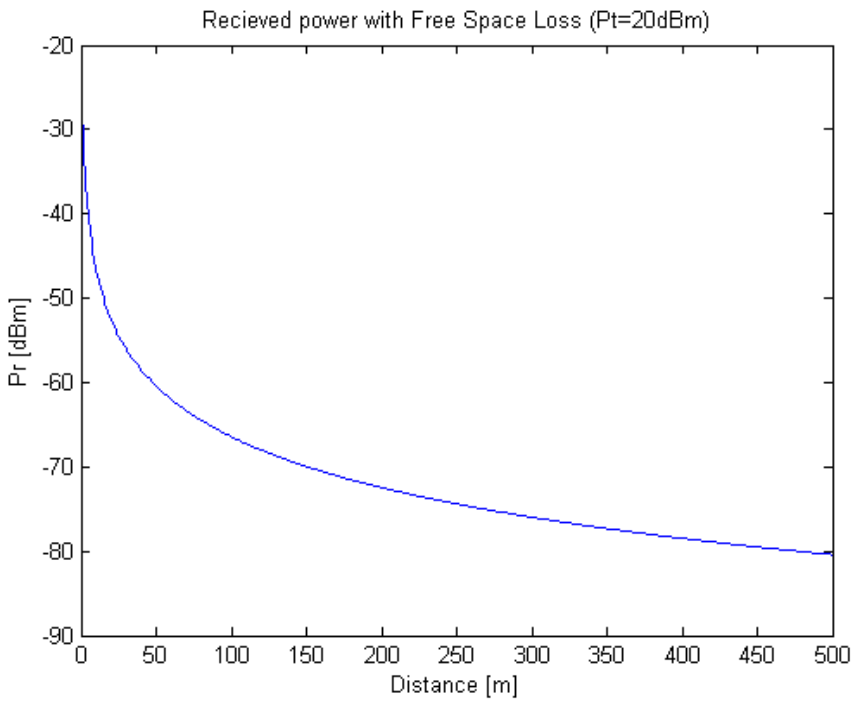
2) Multilateration for RF emitters

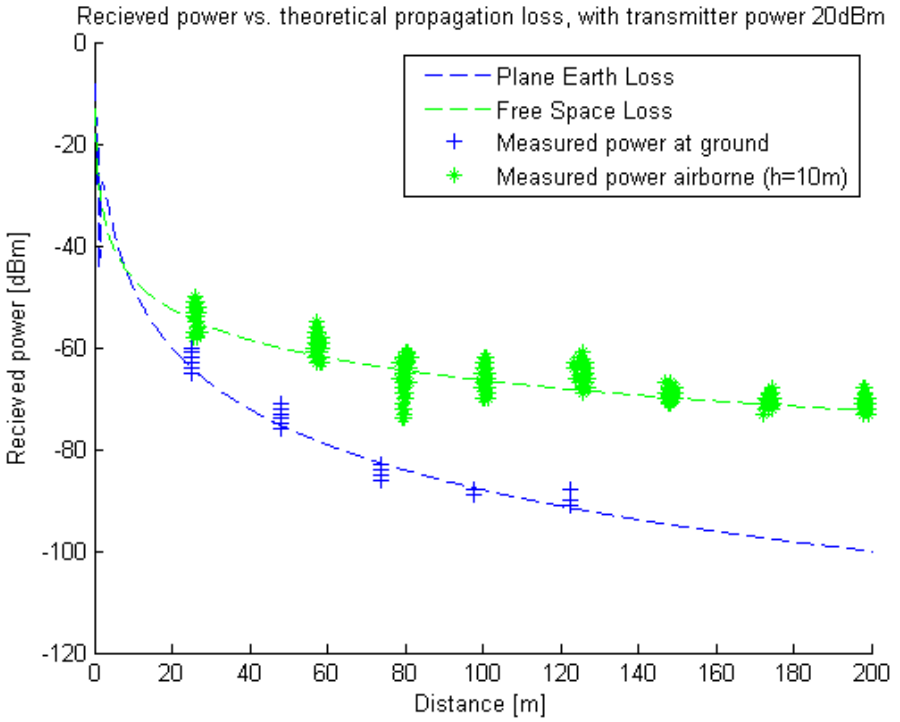
3) (Now) Evolving swarm behaviors

Path loss between emitter (sender) and receiver

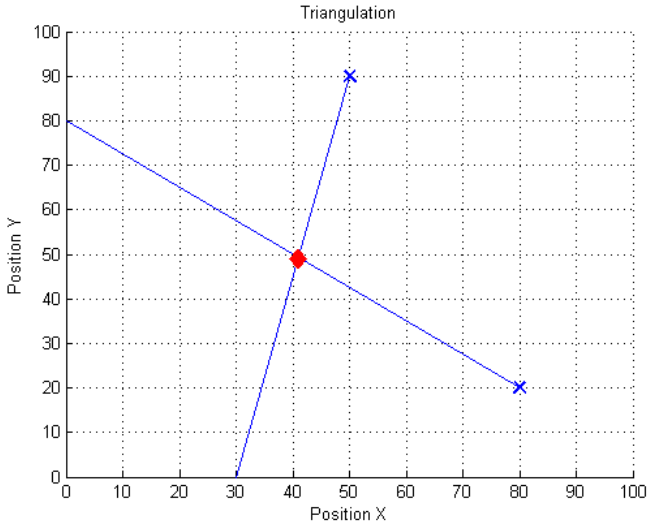


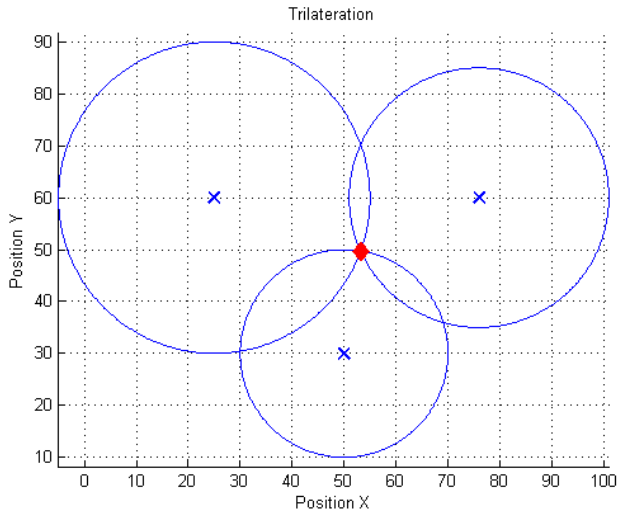
Path loss between emitter (sender) and receiver

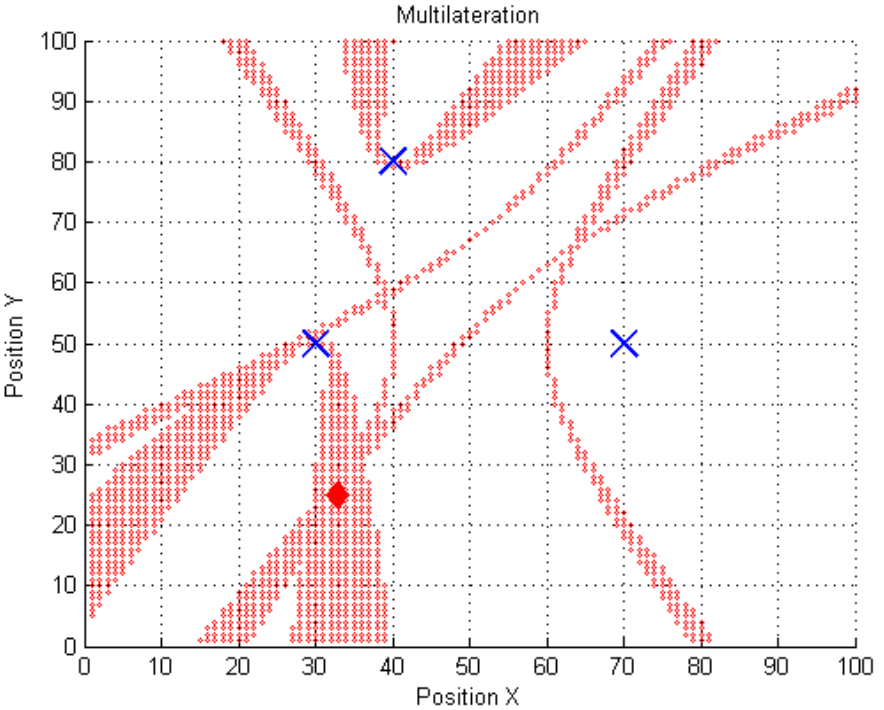




Methods of locating a radio frequency emitter







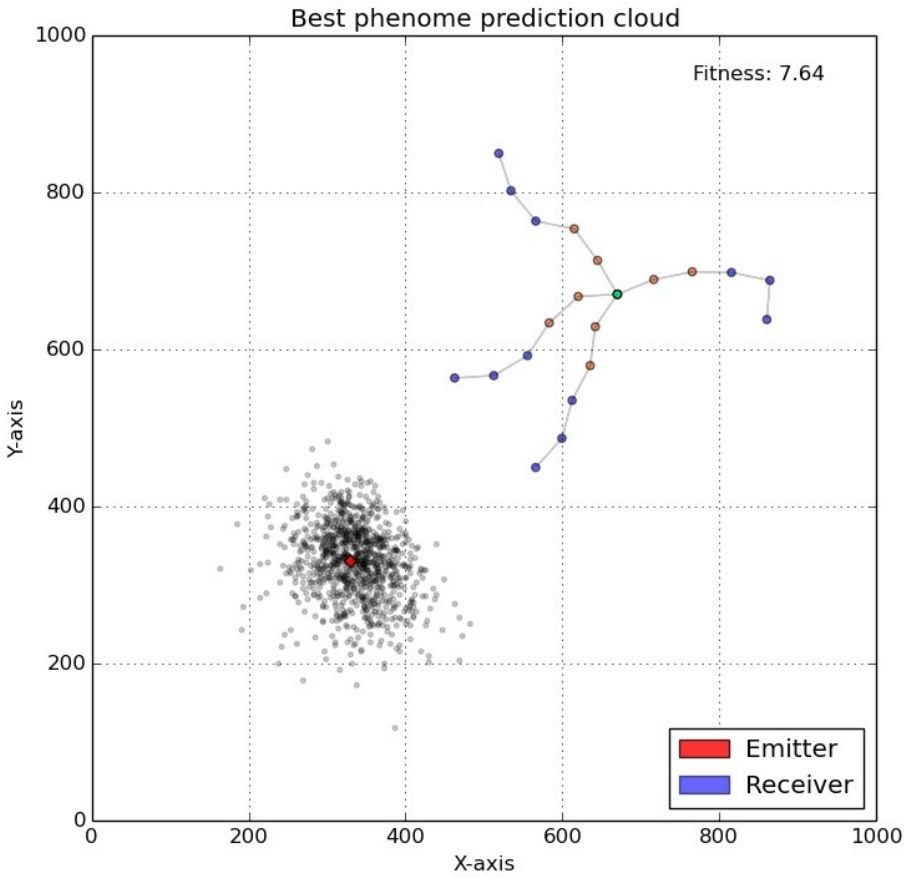
Need to solve a non-linear error function

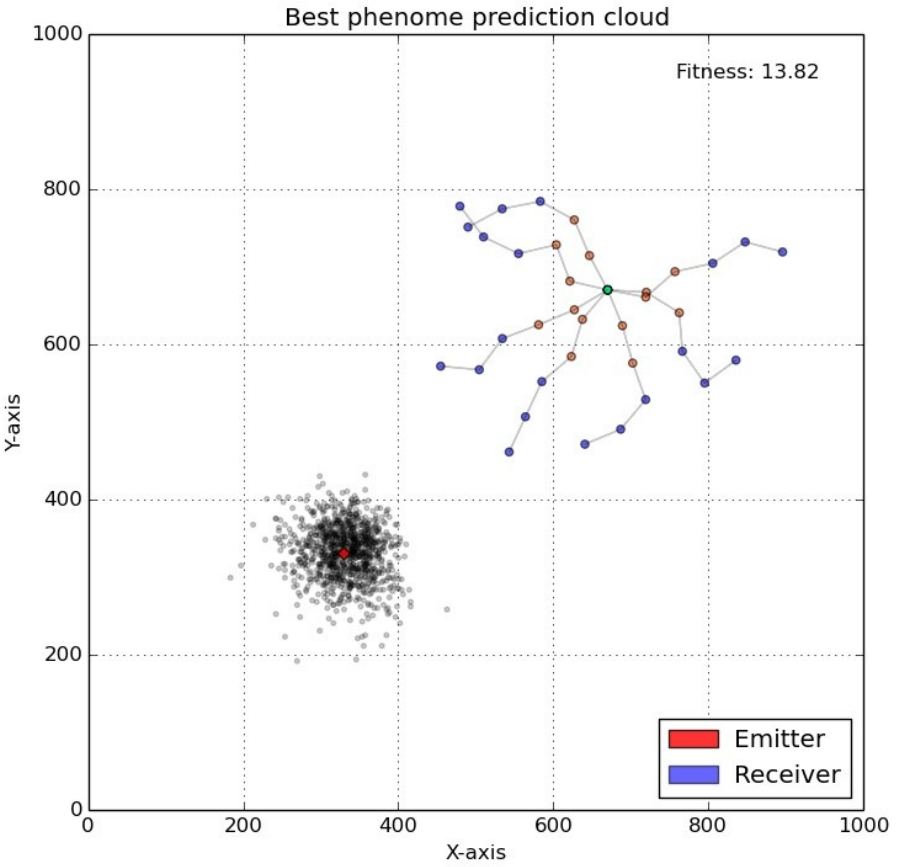
$$P_{kl} = P_k - P_l \tag{2.8}$$

$$Q(x,y) = \sum_{k<l} [P_{kl} - 5\alpha \log(\frac{(x-x_l)^2 + (y-y_l)^2}{(x-x_k)^2 + (y-y_k)^2})]^2 \tag{2.9}$$

$$\min_{0 \leq x \leq m} (\min_{0 \leq y \leq n} Q(x,y)) \tag{2.10}$$

Results of evolving path plans for UAVs





Meta-heuristics for improved RF emitter localization

S. Engebråten, J. Moen and K. Glette (sondre.engebraten@ffi.no)

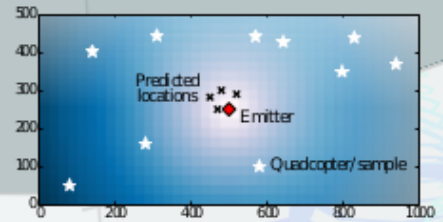
Making PDOA viable for small UAV platforms through meta-heuristics

Background

RF emitter localization

One method of locating an RF emitter is Power Difference of Arrival (PDOA). PDOA relies on measuring the differences in received signal strength at multiple points in space. Given a guess for the emitter location, it is possible to calculate an error measurement using the free space path loss model.

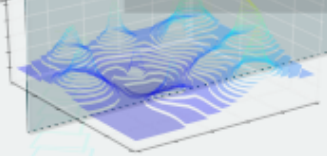
An estimate of the emitter location be can found by solving the non-linear error function for the emitter position. Non-linearity makes this a time consuming optimization. In this paper, meta-heuristics are applied to solve this optimization problem faster, and more precisely, than conventional methods



Non-linear optimization problem

$$P_{kl} = P_k - P_l$$

$$Q(x,y) = \sum_{k<l} [P_{kl} - 5\alpha \log(\frac{(x-x_l)^2+(y-y)^2}{(x-x_k)^2+(y-y_k)^2})]^2$$



Challenges

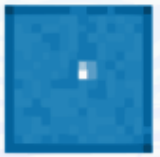
Multi-modality/deceptiveness makes this problem challenging. Instances of the optimization problem often has multiple local optima. These are caused by noisy sensor measurements and the local optima may often be far away from the true emitter location

Strict time constraints are enforced as the goal of this paper is to enable the use of PDOA on resource restricted platforms. The search space is smooth, which may allow for use of local information.

Lack of optimality guarantees combined with strict time constraints make this a challenging problem for meta-heuristic search.

Methods

Preselect NM

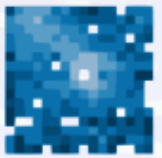


The figures are histograms of how each algorithm explores a single problem instance, accumulated over several runs

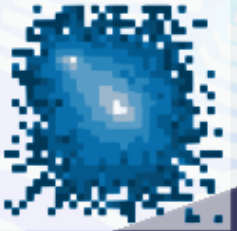
PSO

NM

Diff. Evo.



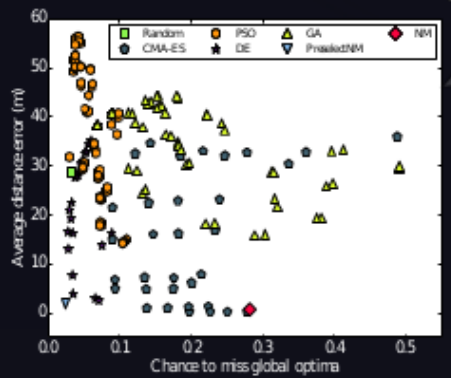
CMA-ES



GA

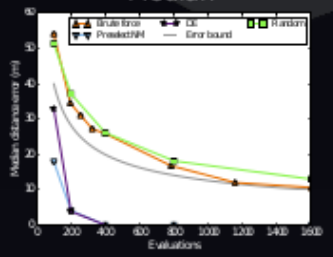
Results and conclusion

Multi-objective performance

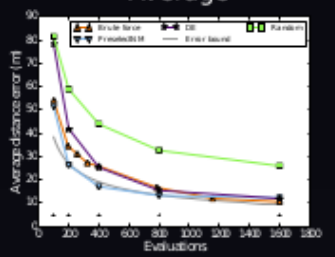


Unreliability affects average performance

Median



Average



Conclusion

A no-cost speedup up 2x can be achieved using Preselect NM

Using median metric (instead of average), 400 evaluations with a meta heuristic is comparable to 40 000 evaluations (100x) using "brute force"

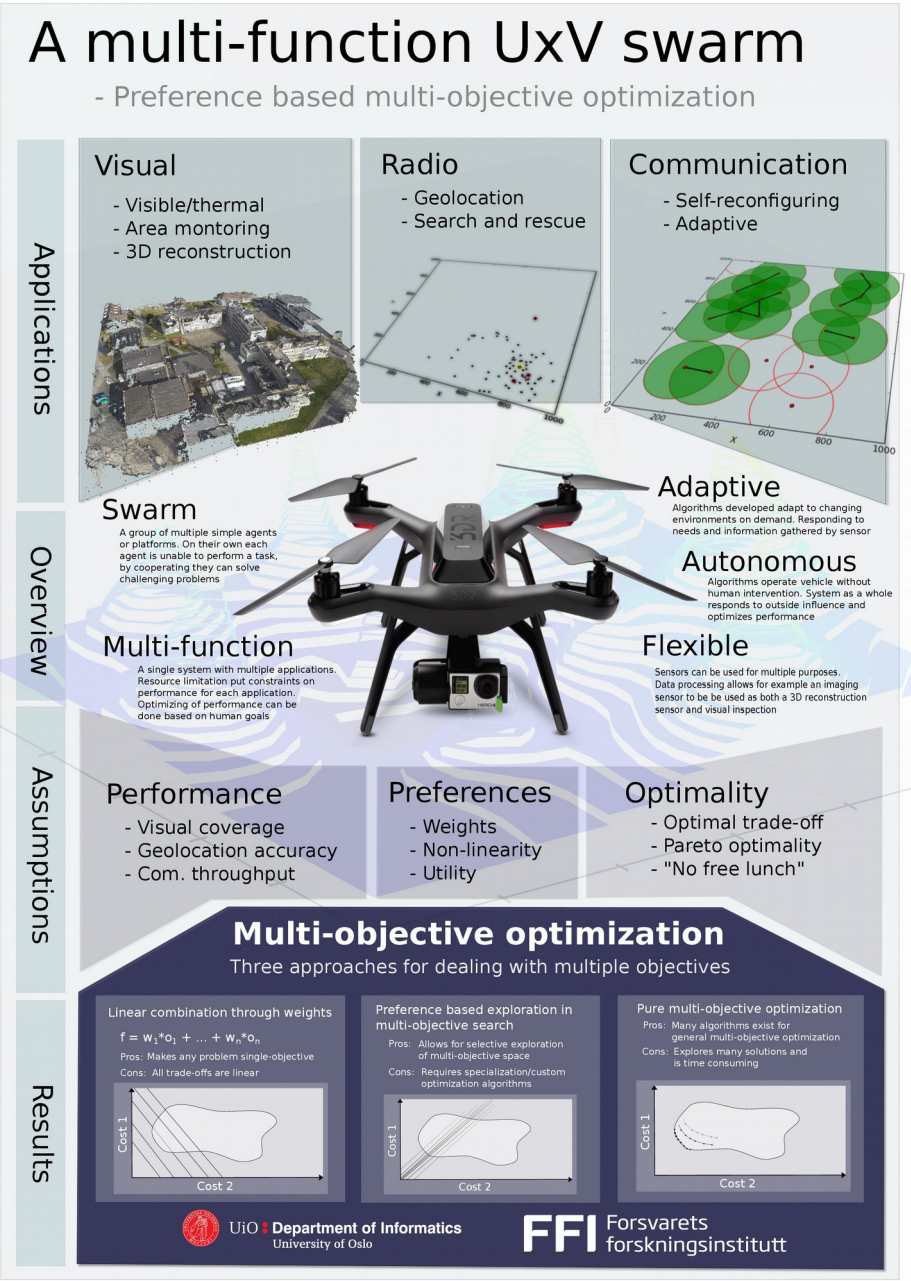
For quick convergence and hill climbing, an adaptive mutation/permutation step is essential.

FFI Forsvarets forskningsinstitutt

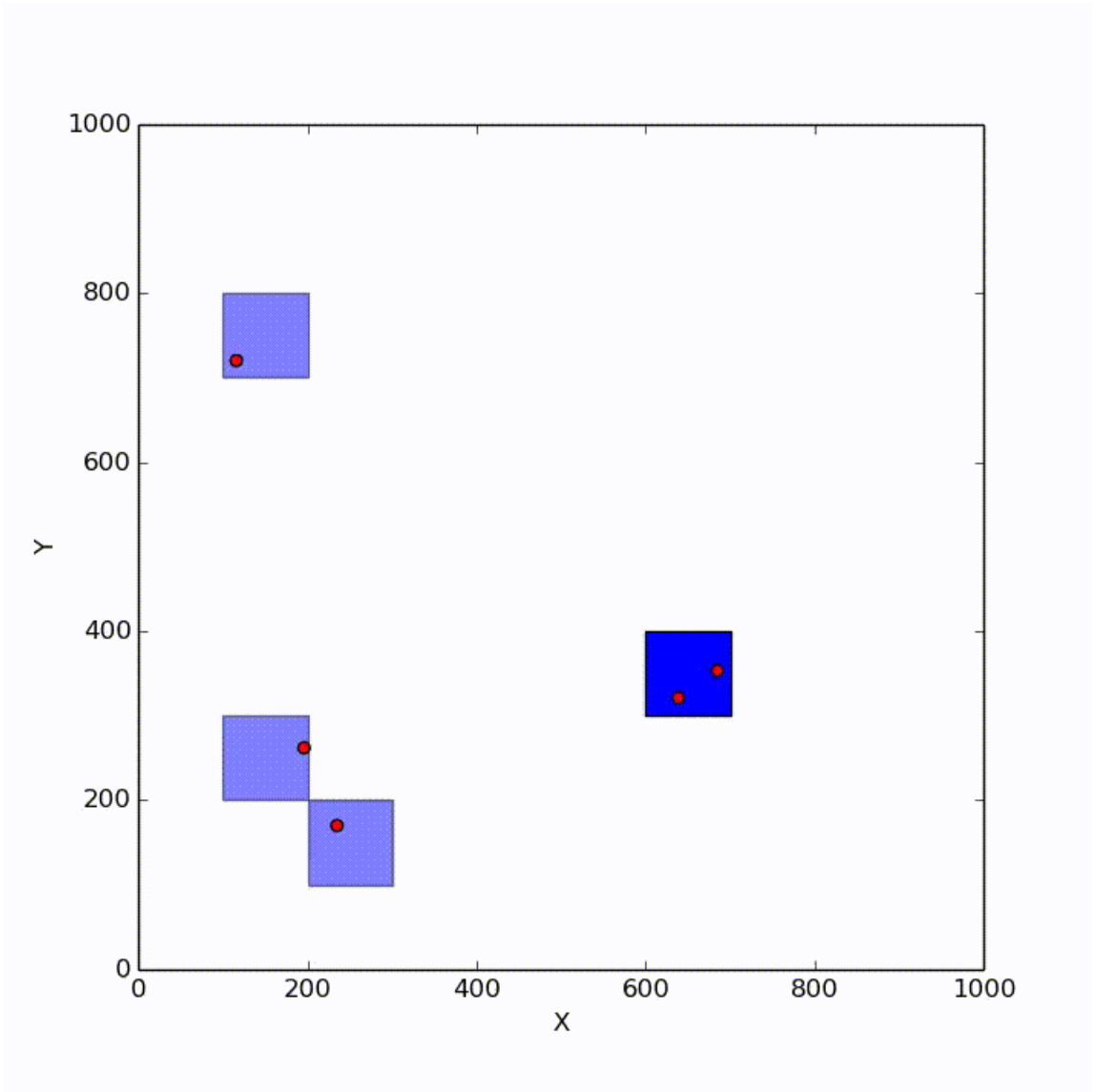
UiO : Department of Informatics
University of Oslo

Multi-function swarm

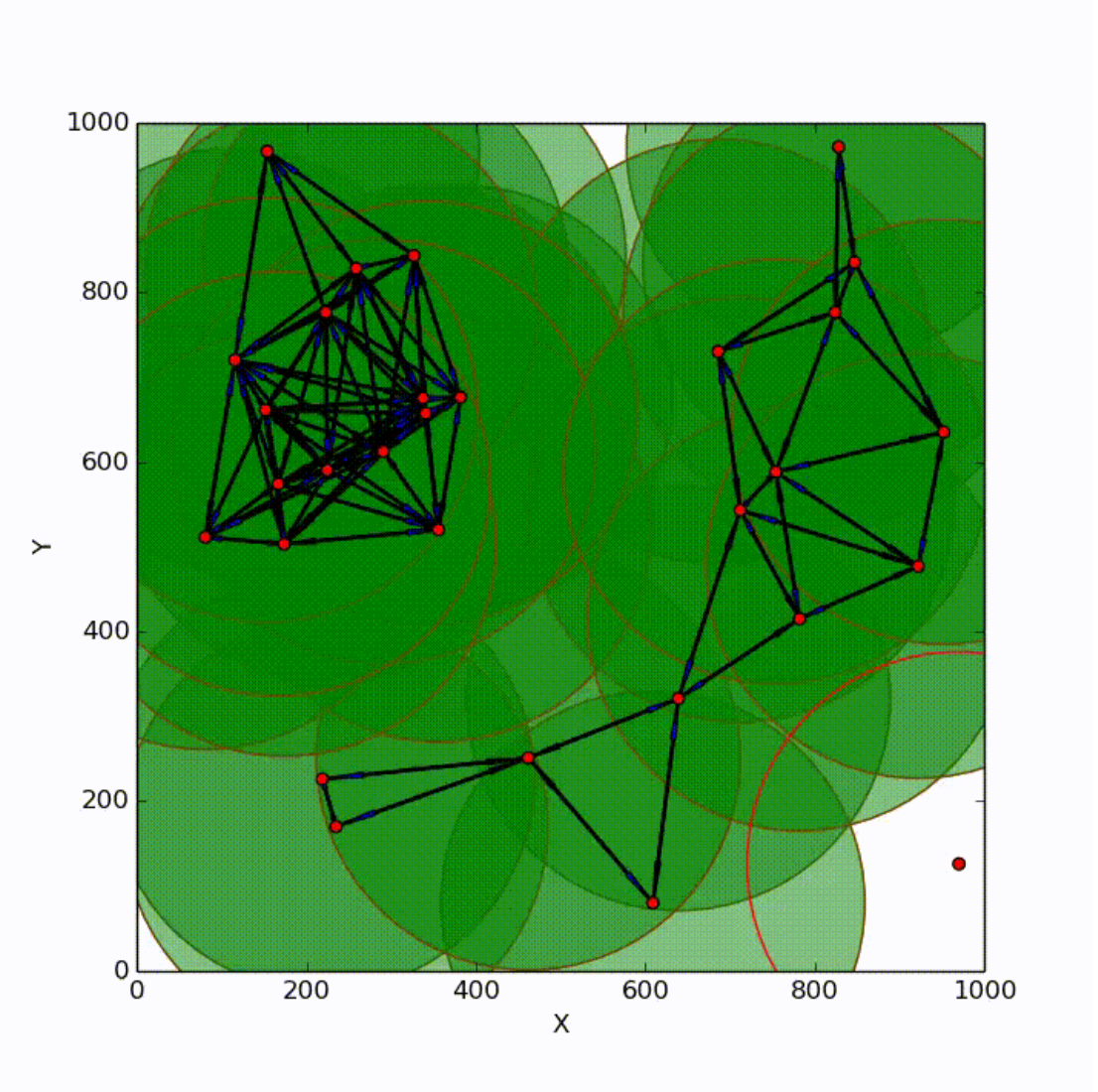
- ***What*** is a multi-function swarm?
  - One system that solves many tasks
  - Many agents/units that cooperate
  - A terminology to optimize resource use
- ***Why*** work on a multi-function swarm?
  - Capable system
  - Connects activites with a common goal
- ***How*** work towards a multi-function swarm?
  - Identify applications
  - Handle/maximize user utility



Algorithms for perimeter surveillance



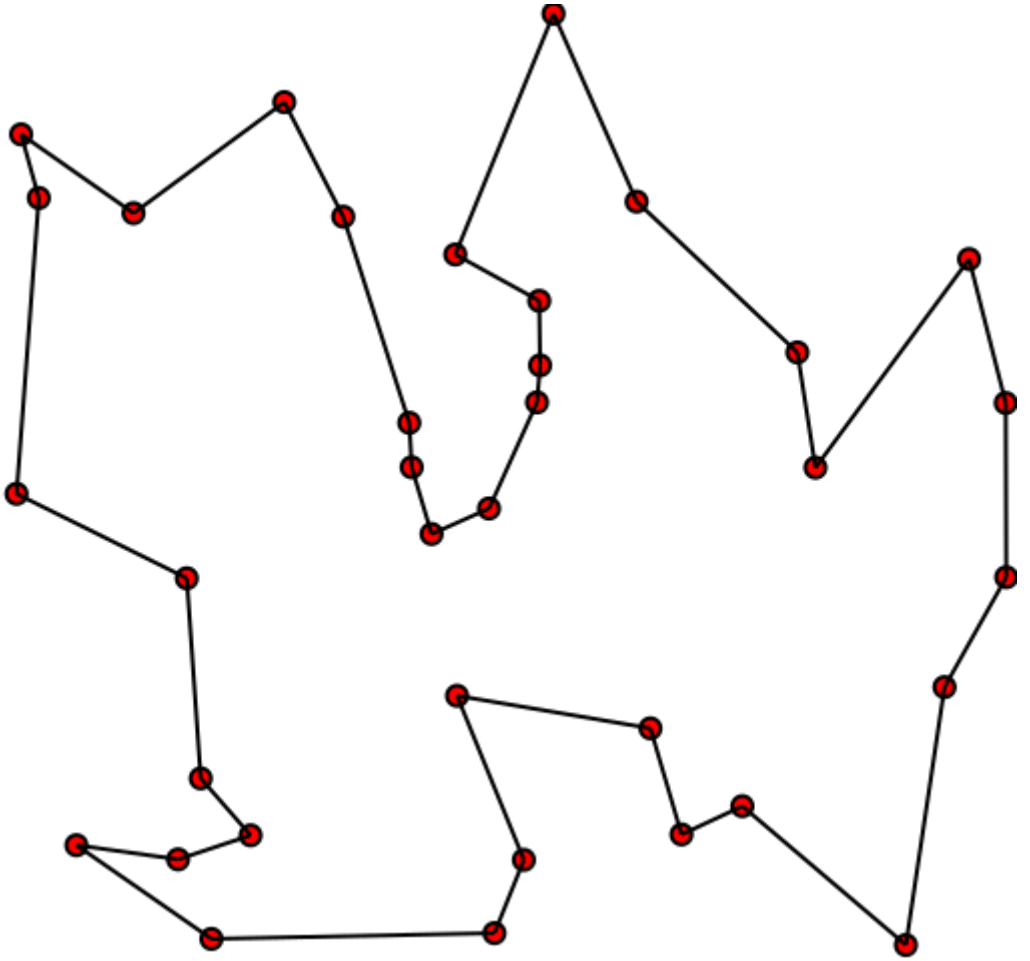
Algorithm to construct communication network



Impossible problems, why are they impossible?

Best algorithms to solve them take a really long time for large problems

Travelling salesman problem TSP (NP-hard)



N = 35 N! = 10333147966386144929666651337523200000000

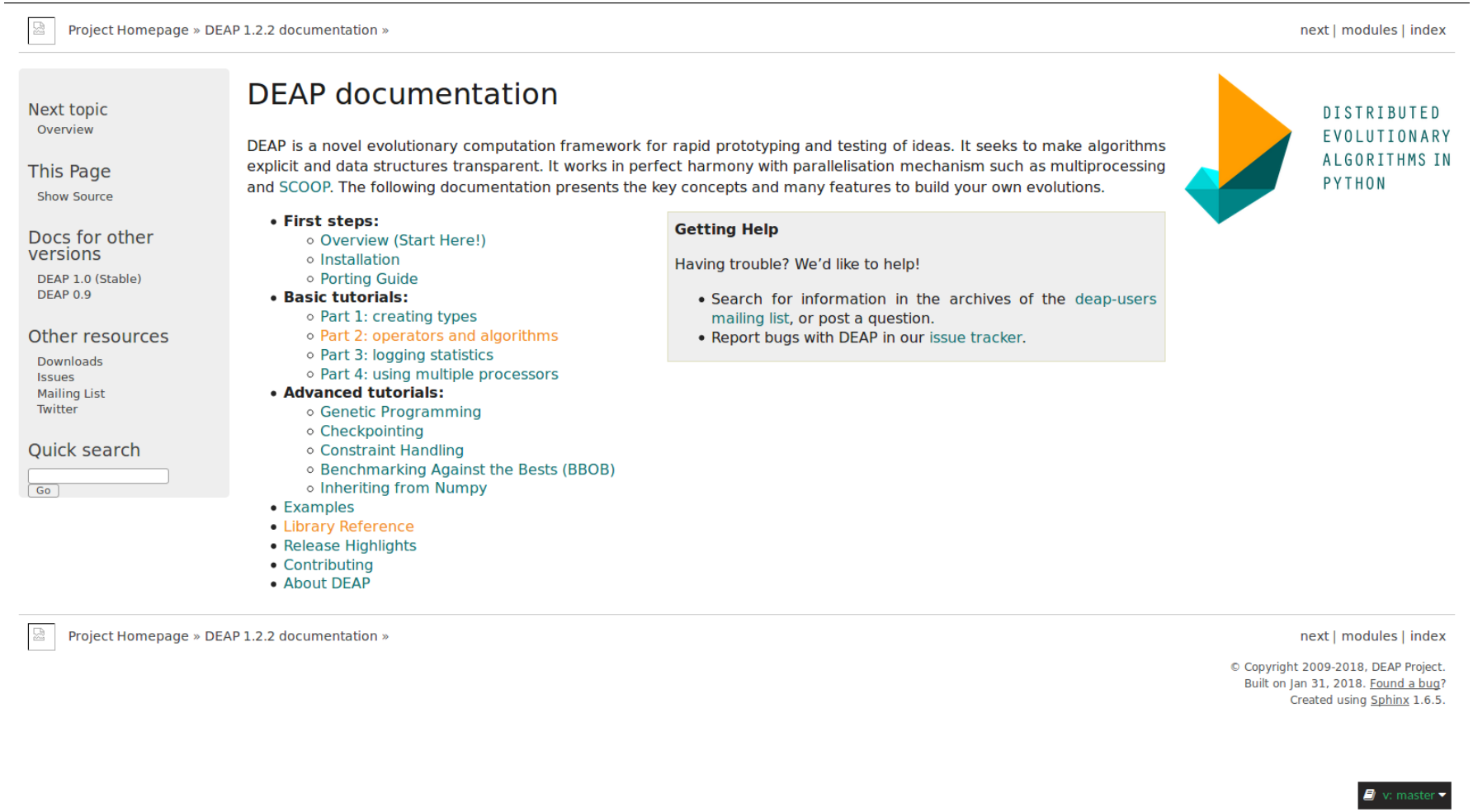
Travelling salesman problem - scaling

1 1
2 2
3 6
4 24
5 120
6 720
7 5040
8 40320
9 362880
10 3628800
11 39916800
12 479001600
13 6227020800
14 87178291200
15 1307674368000
16 20922789888000
17 355687428096000
18 6402373705728000
19 121645100408832000
20 2432902008176640000

Travelling salesman problem - scaling

21 51090942171709440000
22 112400072777607680000
23 2585201673884976640000
24 62044840173323943936000 0
25 1551121004333098598400000 0
26 40329146112660563558400000 0
27 1088886945041835216076800000 0
28 30488834461171386050150400000 0
29 884176199373970195454361600000 0
30 26525285981219105863630848000000 0
31 822283865417792281772556288000000 0
32 26313083693369353016721801216000000 0
33 868331761881188649551819440128000000 0
34 29523279903960414084761860964352000000 0
35 1033314796638614492966665133752320000000 0
36 37199332678990121746799944815083520000000 0
37 1376375309122634504631597958158090240000000 0
38 52302261746660111176000722410007429120000000 0
39 2039788208119744335864028173990289735680000000 0
40 81591528324789773434561126959611589427200000000 0

DEAP – Distributed Evolutionary Algorithms in Python



Examples: https://goo.gl/NLr9gY

Hands on example: Solving TSP!

Link: https://github.com/sondreengebraten/RobodojoImpossible2018