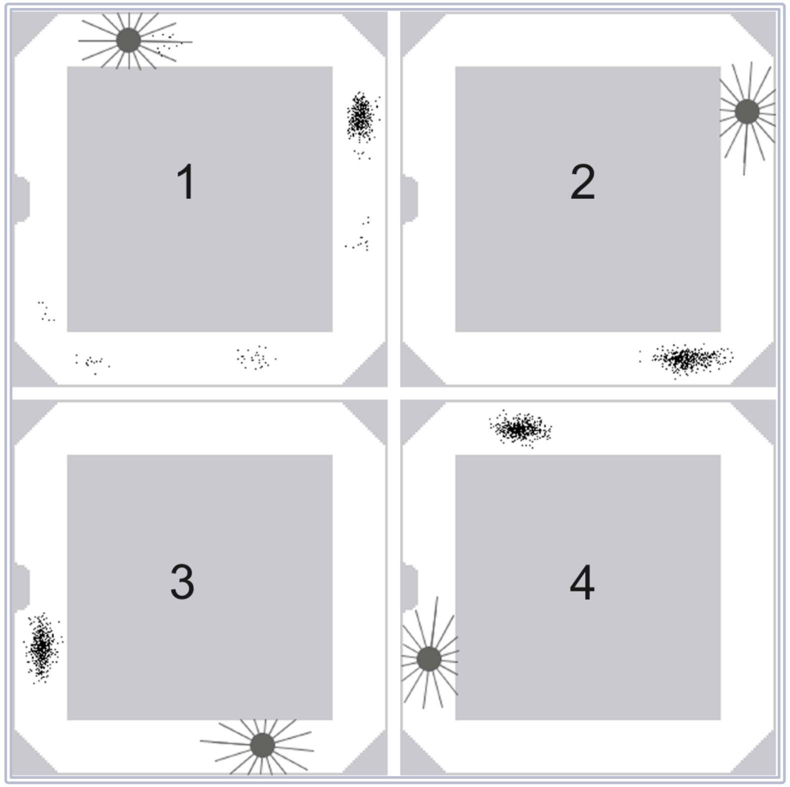
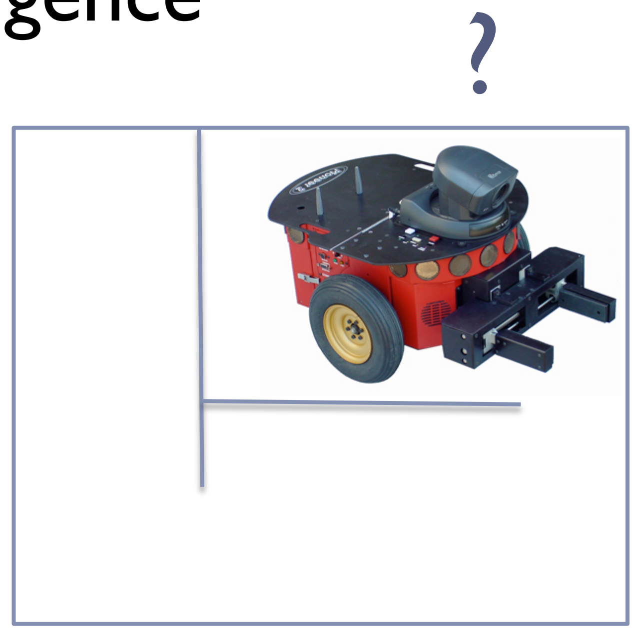


Diversity maintenance in particle filters

Gert Kootstra

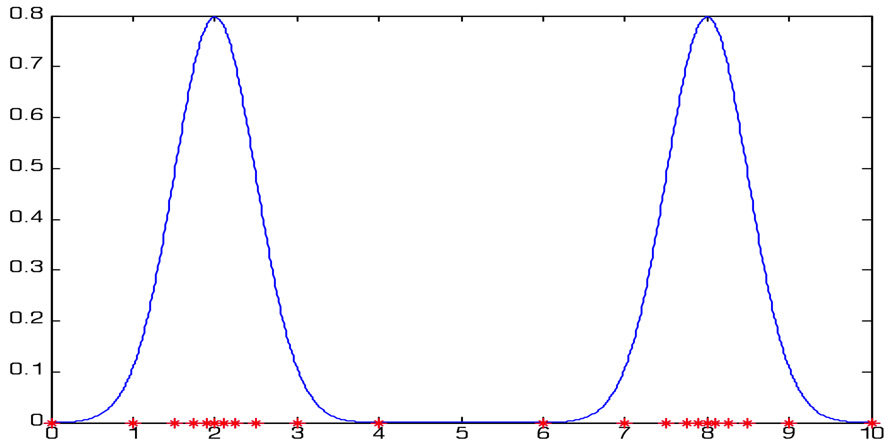
Overview

- Particle filters for localization
- Similarities between particle filters and genetic algorithms
- Problem of premature convergence
- Niching methods
- Experiments
- Results
- Conclusion



Particle filters

- A particle filter represents the probability distribution with a set of particles



- Distribution can take any form
- Density of particle should approximate the true distribution. True for $N \uparrow \infty$

Monte-Carlo localization

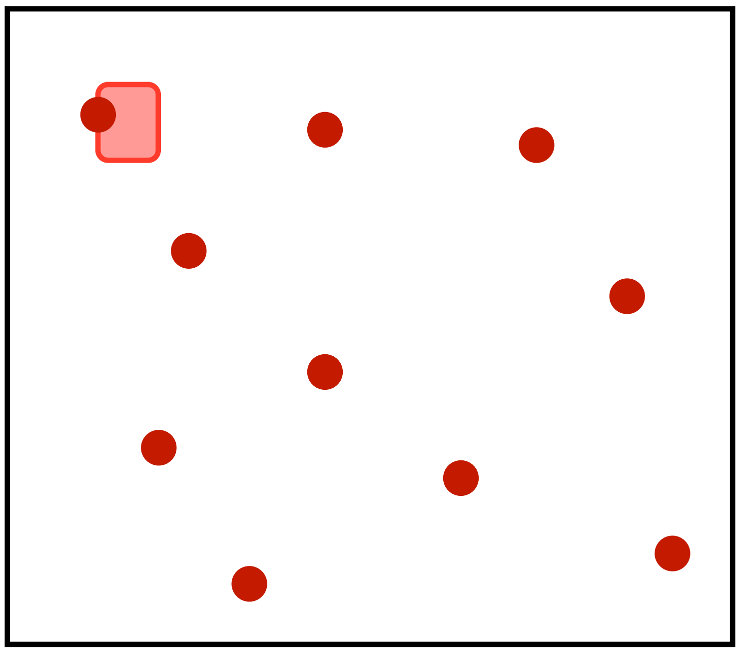
- In MCL, a particle filter is used to estimate the position of the robot.
- Information used
  - Map of the environment
  - Sensory reading of the robot
  - Action (motion) performed by the robot

Monte-Carlo localization

- Set of particles $S_t = \{(x_t^i, w_t^i) \mid i = 1, \ldots, N\}$
  - Each particle is a hypothesis about the location x_t^i
  - w_t^i is the weight (probability) of that particle
- Start with random distribution of particles
- Iterative optimization process
  1. Apply motion model $x_{t+1}^i = f(x_t^i, u_t^i)$
  2. Apply sensor model $w_{t+1}^i = g(x_{t+1}^i, z_t^i)$
  3. Resample the particles

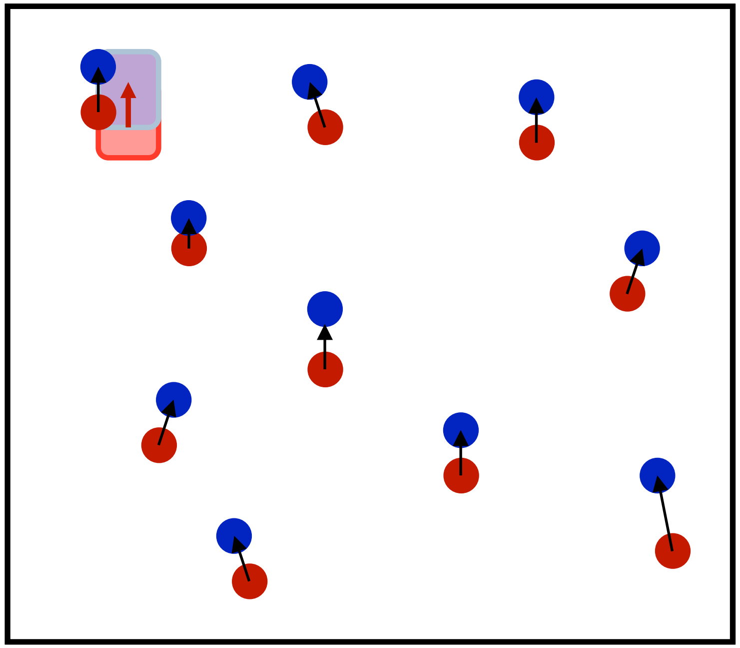
Monte-Carlo localization

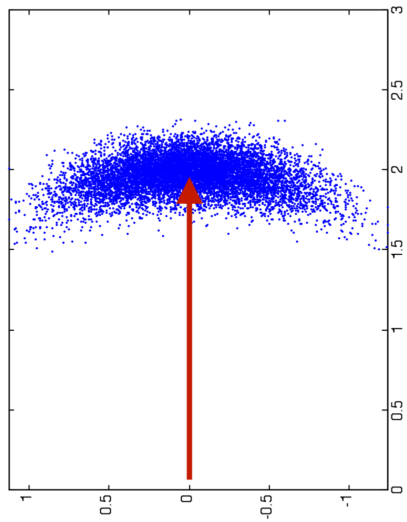
- Initial distribution: random



Monte-Carlo localization

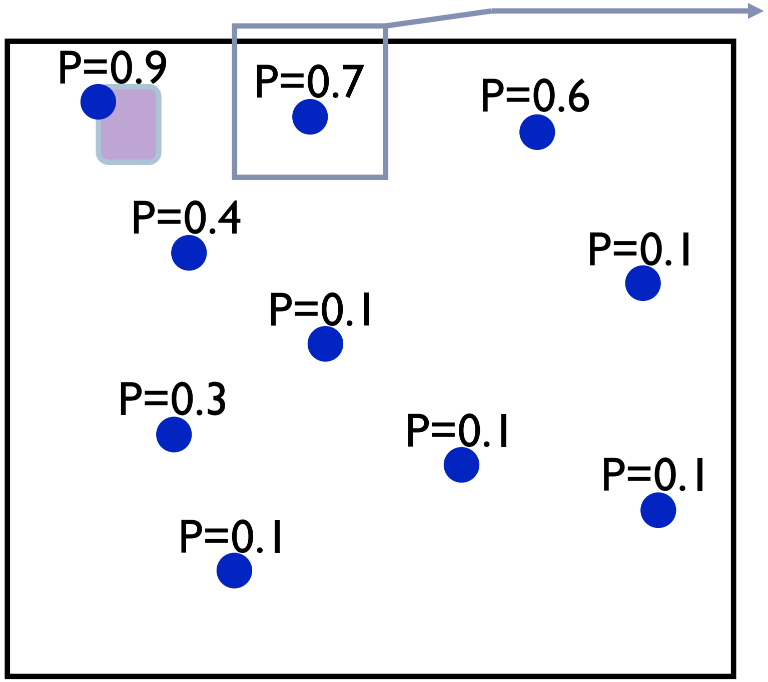
- Applying motion model
  - Next position of p's based on motion
  - Including noise to represent uncertainty

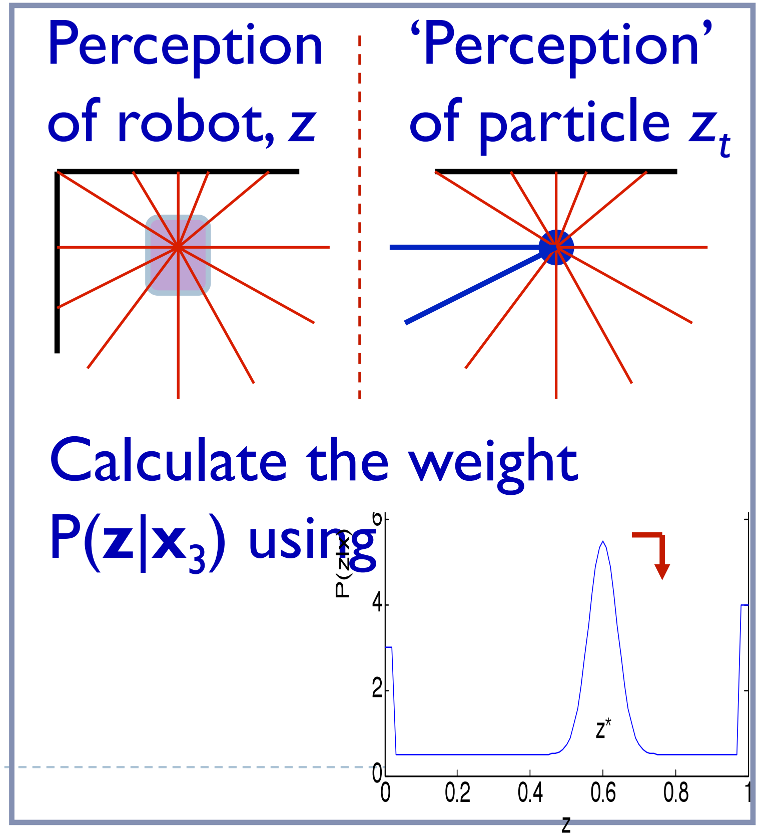




Monte-Carlo localization

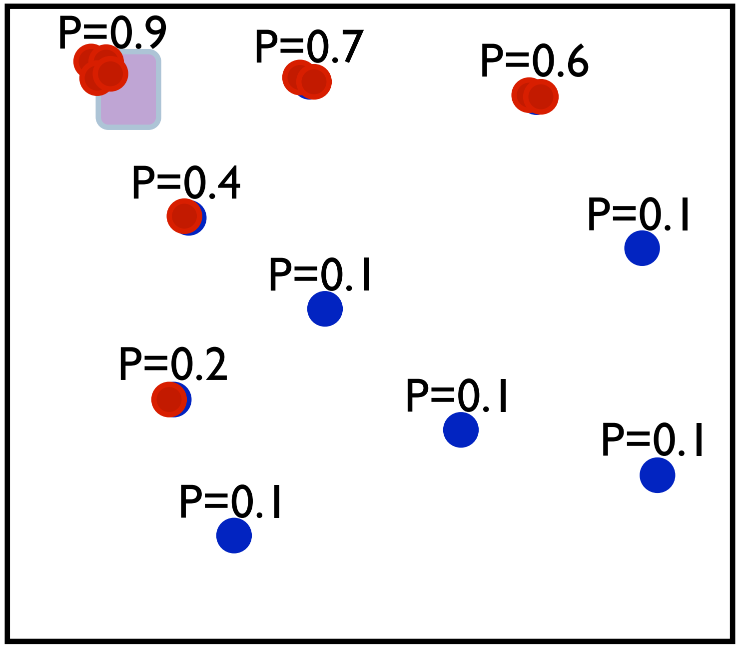
- Applying sensor model
  - Calculating the particle weights





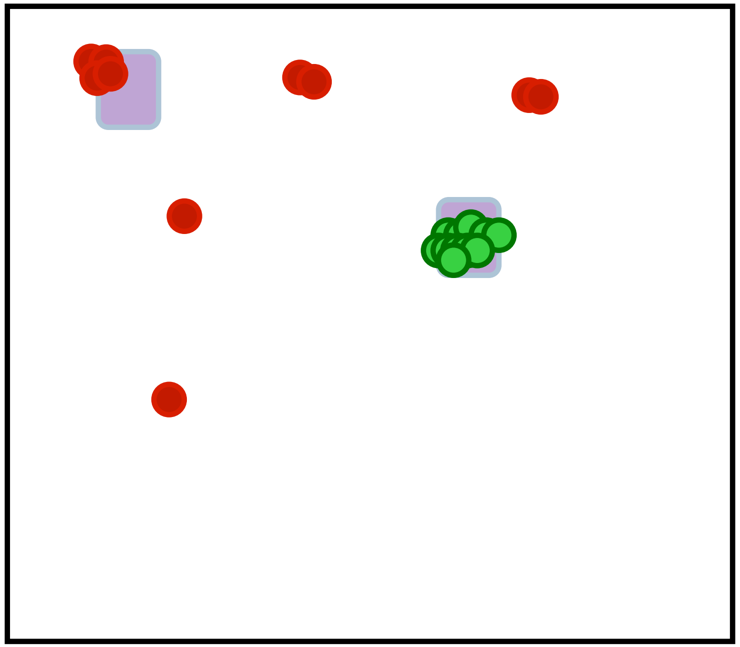
Monte-Carlo localization

- Resampling the particle population
  - Weight-proportional sampling

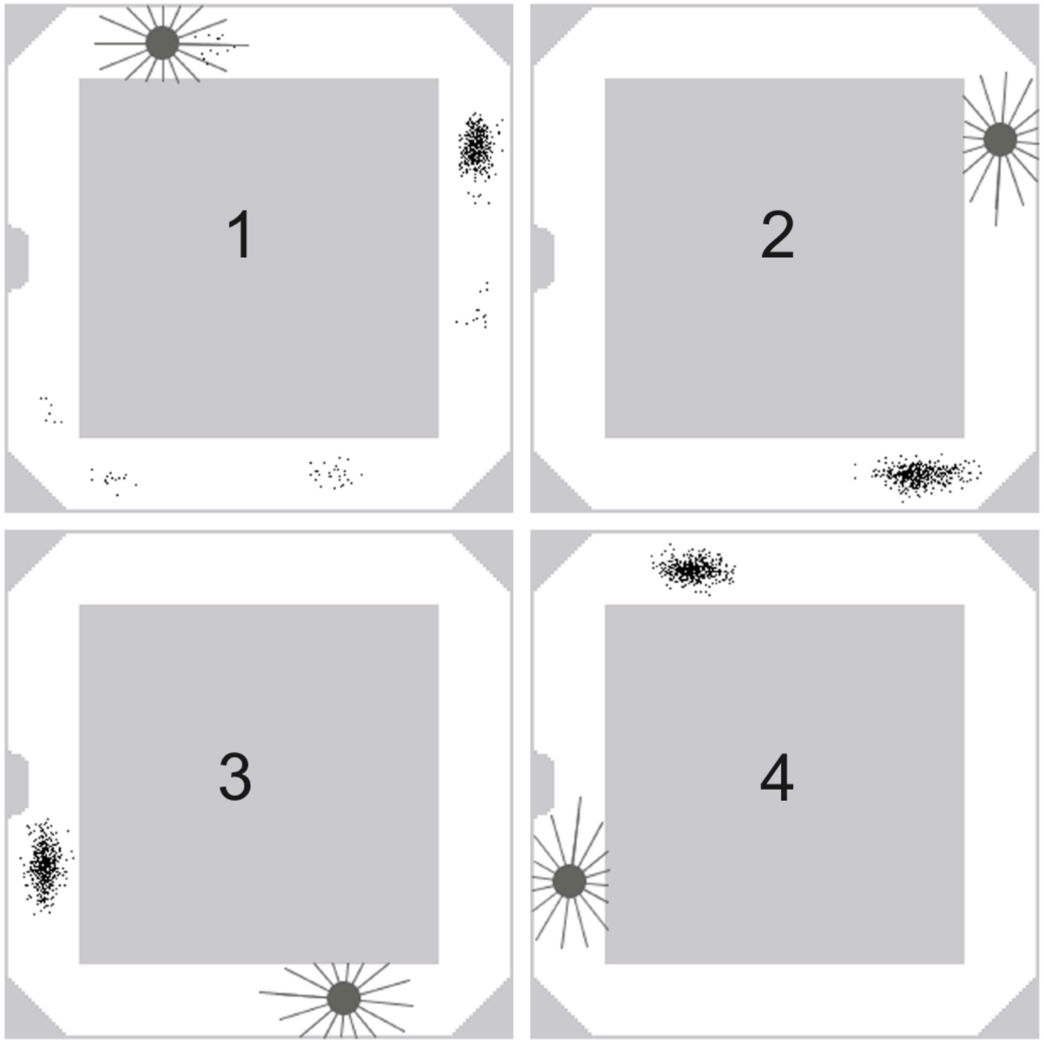


Monte-Carlo localization

- Final distribution



Problem: premature convergence

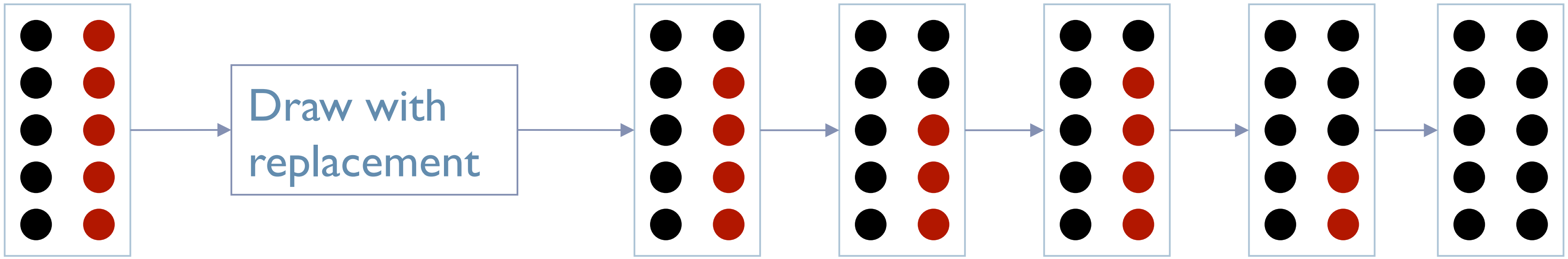


Premature convergence

- Loss of diversity
- Makes the filter end up in a sub-optimal solution
- Especially when observations lead to ambiguous situations
  - In symmetrical environments (many office buildings)
  - With multiple solutions
  - With noisy sensors

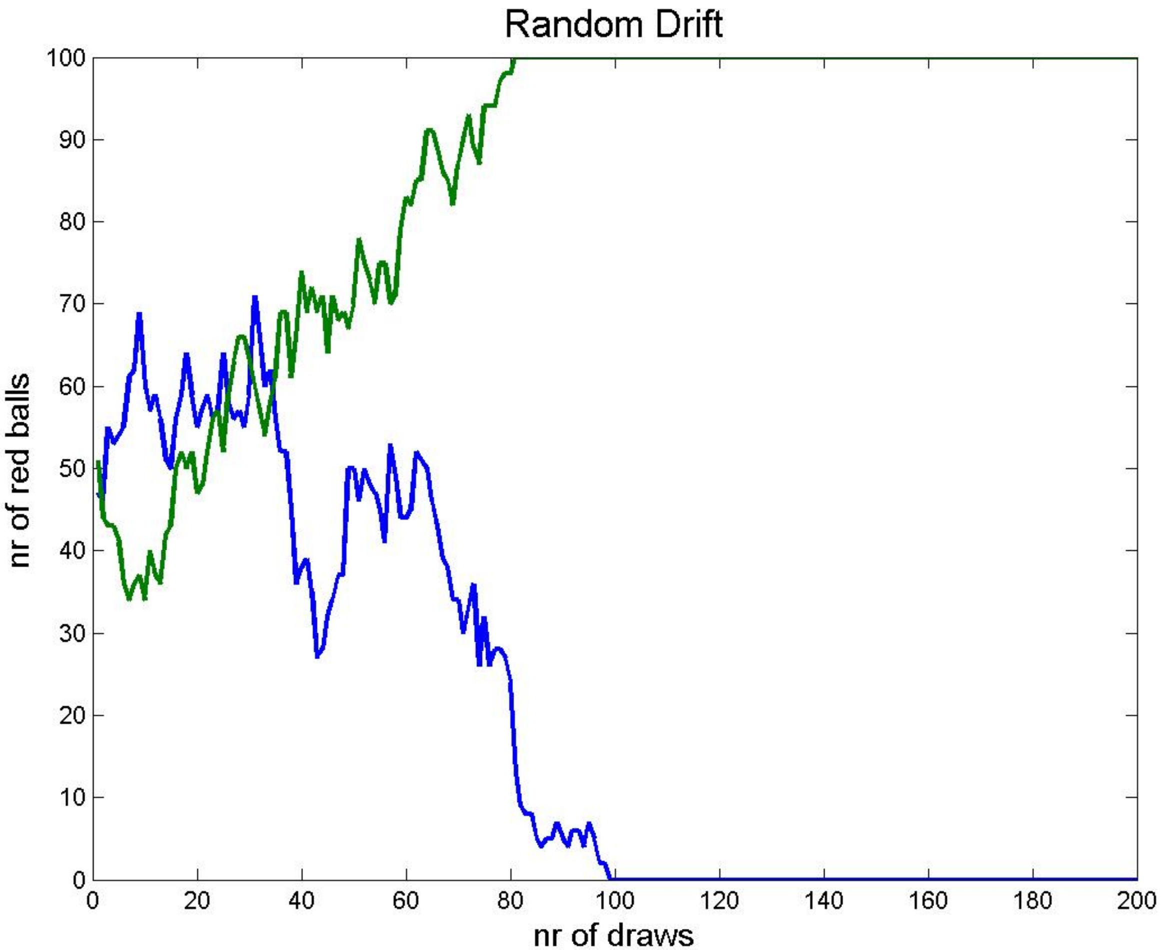
Random drift

- Reason: random (genetic) drift
  - Consider 5 particles for solution A en 5 for B
  - All the same weight
  - This is what happens in the resampling process



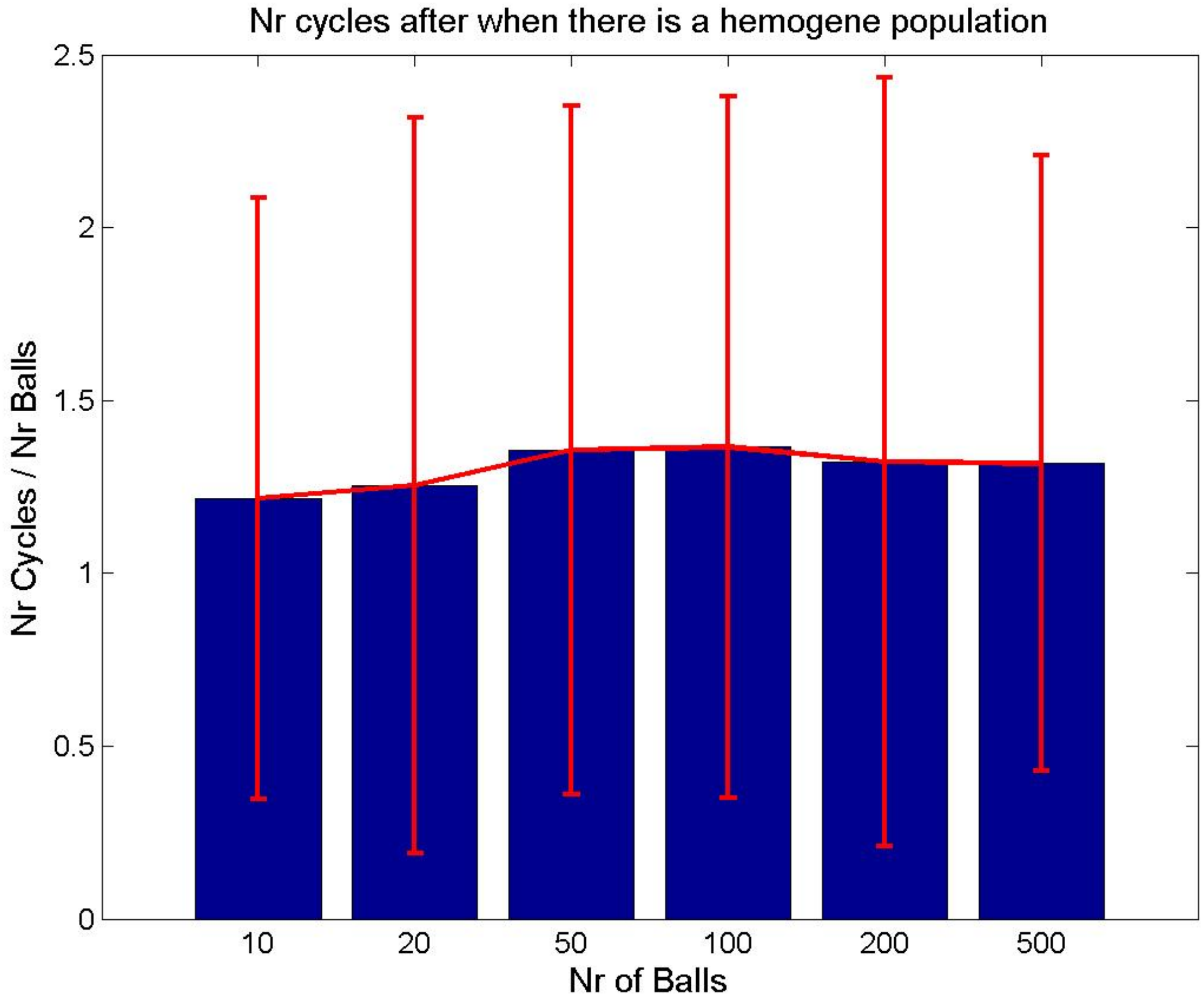
Random drift

- Two examples with 100 particles starting 50-50



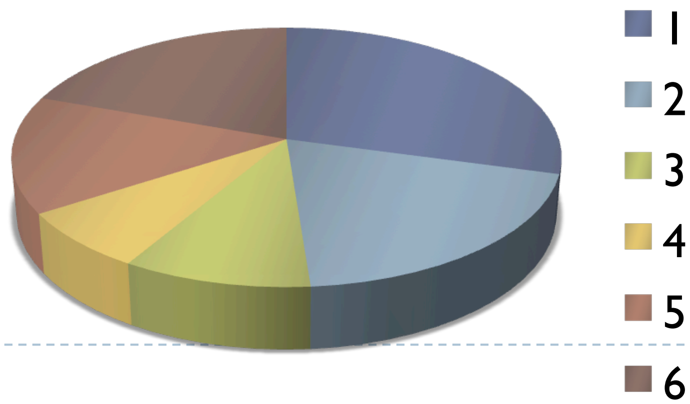
Random drift

- Many examples: time to convergence



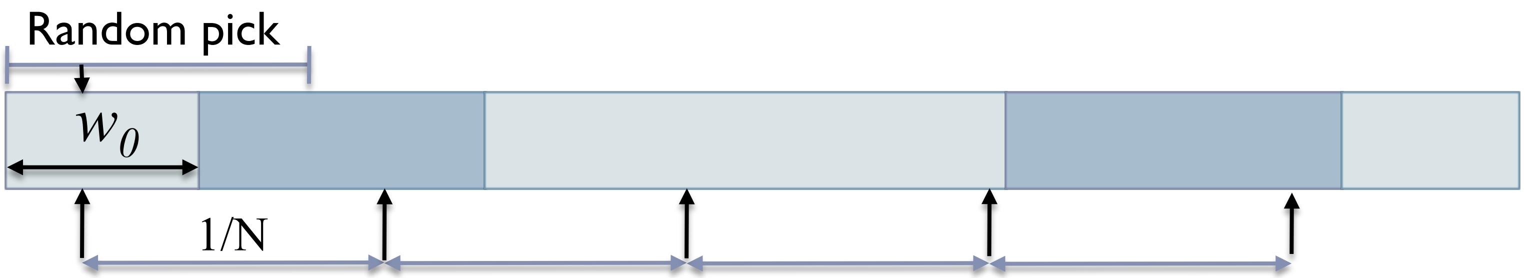
Premature convergence

- Reason for this drift
  - The variance in the sampling method
  - Population after sampling might not resemble the weight distribution
  - Particles in different regions compete for limited resources (N particles in next generation)
- Variance of roulette-wheel sampling is particular high



Stochastic universal sampling

- Stochastic universal sampling: lower variance



  - Only one random number generator
  - Less variance in sampling: ±1 particle
  - Also faster
- However, premature convergence remains
  - Demo

Particle filters vs genetic algorithms

- Same mechanism: iterative optimization

Particle Filter	Genetic Algorithm
Particles	Individuals
Random initial distribution	Random initial distribution
Motion model + noise	Mutation (noise)
Transition model	Fitness function
Resampling	Reproduction (weight based)
Random drift	Genetic drift

- Same problems and same solutions

Diversity in natural systems

- What is the reason that in nature there are many different species and not one due to genetic drift?
- Two of the answers:
    1. No competition between different niches
    2. Fitness advantage for species with less members (frequency-dependent selection)

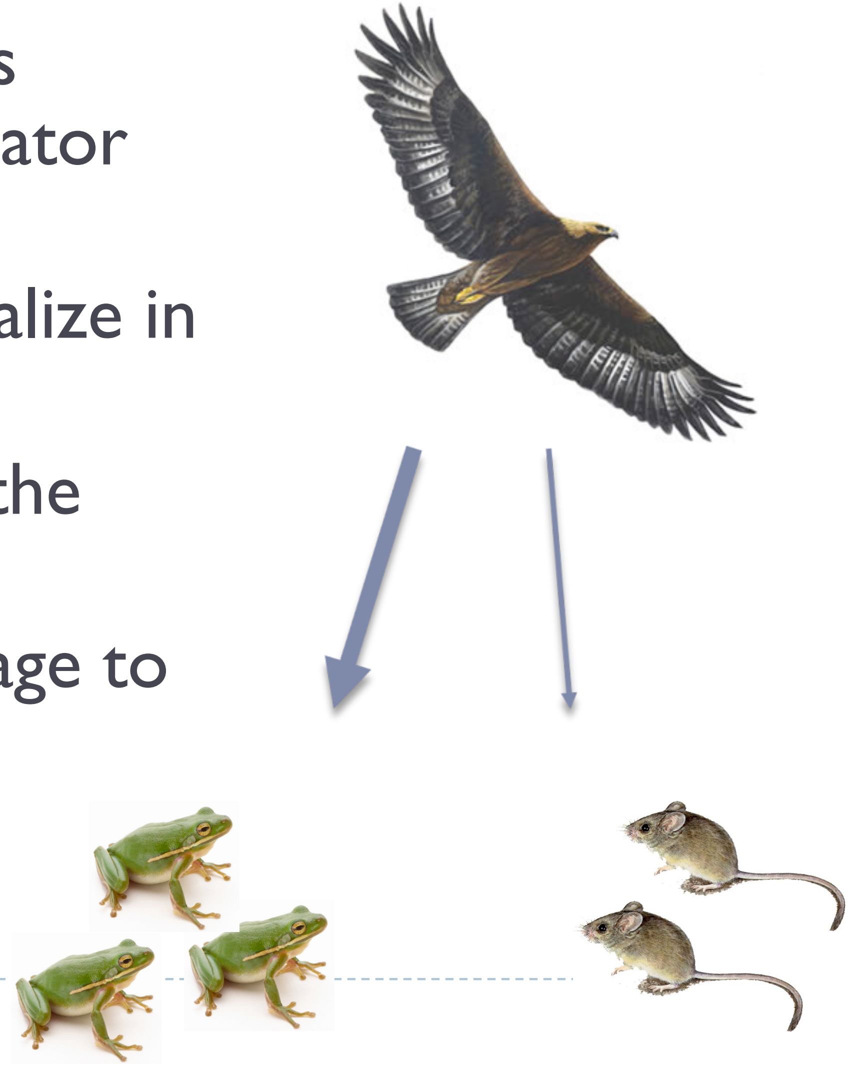
- Niche
    - Environment for particular species (food, temp,…)

1. No competition between niches

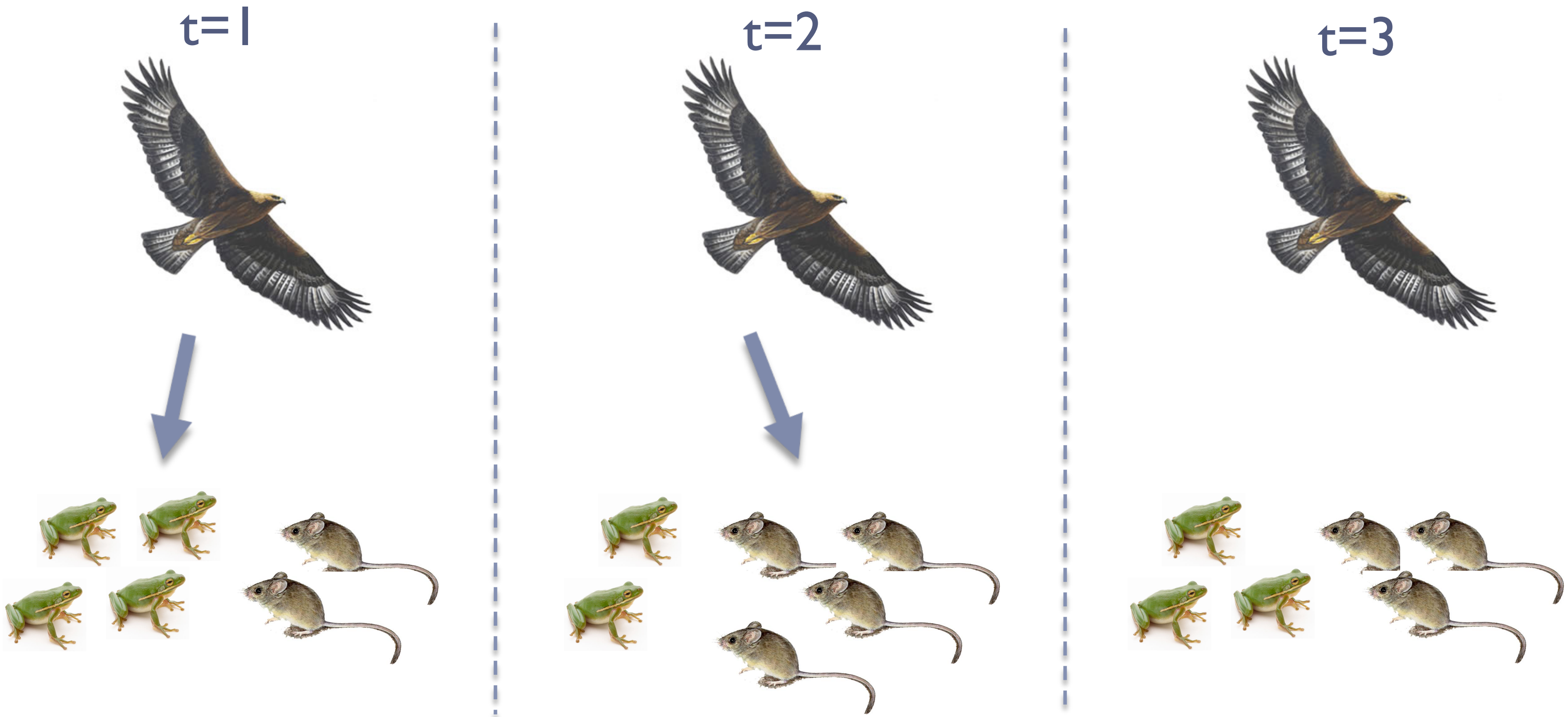
- Source of genetic drift
  - competition between different niches for limited resources
- But many species do not compete because of different niches
  - No competition for space, food, etc.

2. Frequency-dependent selection

- Predator-prey systems
  - Consider two prey species (mice, frogs) and one predator (eagle)
  - The predator has to specialize in one of the two
  - It obviously specializes in the largest group
  - This gives a fitness advantage to the smaller group and disadvantage to the bigger



2. Frequency-dependent selection



- Smaller groups have advantage
- Results in balance between group size

Niching methods in GA

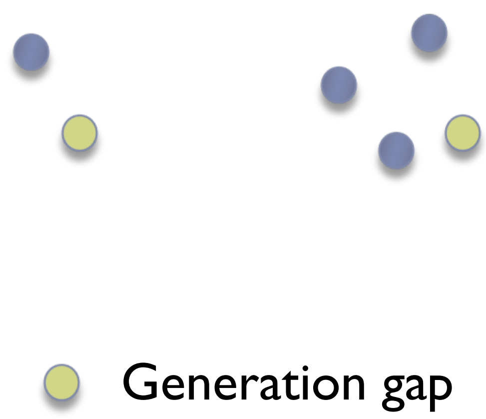
- Terminology
  - Niche → Solution
  - Limit resource → Limit nr of individuals

Niche	Solution
Limit resource	Limit nr of individuals

- Solutions in GA field: Niching methods
  - Crowding
  - Sharing / Frequency dependent selection
  - Local selection

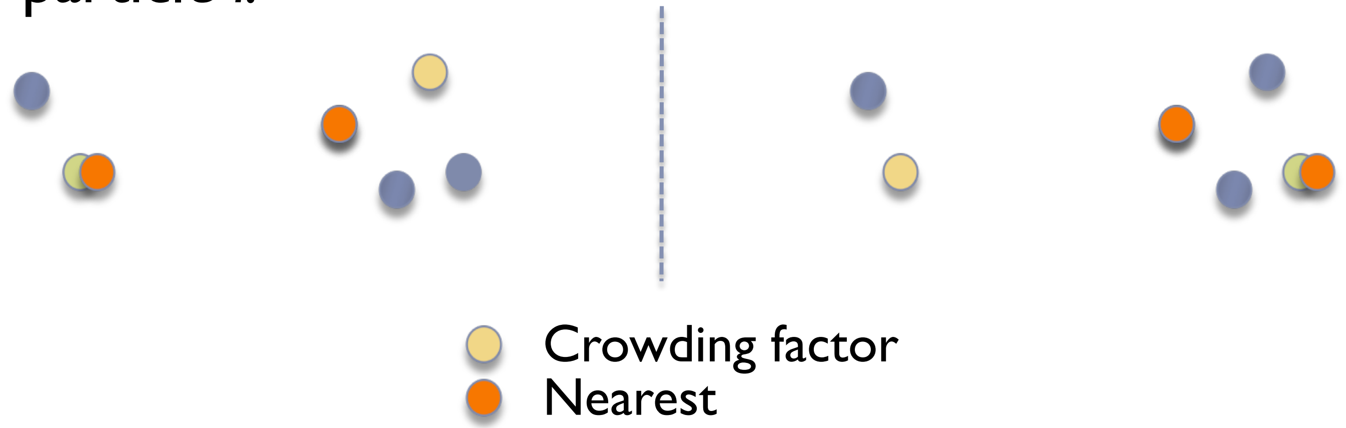
Crowding / Closest of the Worst

- Apply motion and sensor model to all particles
- Crowding instead of the standard resampling:
  - Select part (20%) of the population, the generation gap, to reproduce using weight-proportional selection, so selecting the more probably particles



Crowding / Closest of the Worst

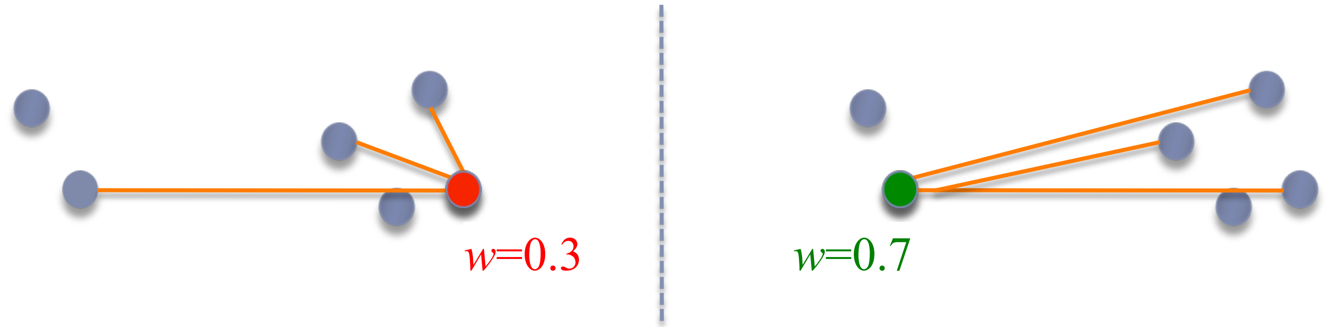
- For every particle *i* in the generation gap
  - A proportion (1%), the crowding factor, is randomly sampled from the worst particles
  - The nearest particle in the crowing factor is replaced by particle *i*.



  - Large groups: competition within the niche
  - Small groups: change to get a particle from another group

Frequency dependent selection

- Apply motion and sensor model to all particles
- Adjust the weights
  - $$\hat{w}_{t+1}^{i} = w_{t+1}^{i} \cdot \sum_{j=1}^{2 \cdot N} \mathrm{dist}\left(x_{t+1}^{i}, \mathrm{rand\_particple}\right)$$
  - Fitness advantage for small groups



- Resample normally

Local selection

- Different method
- The size of the particle population adapts to the carrying capacity of the environment.
  - No competition for limited resources
  - A niche will maintain as many particles as suited for the niche's 'fitness'.

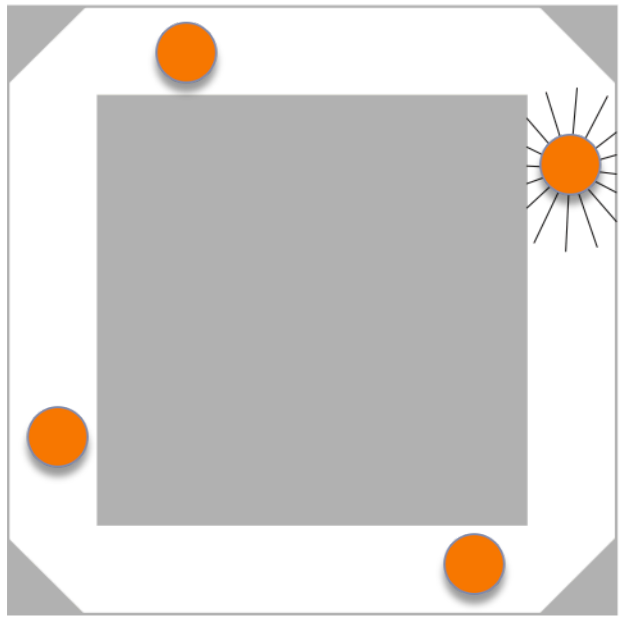
- Every particle now has an amount of energy

Local selection

- Apply motion and sensor model to all particles
- Divide world in bins and count particles per bin
- For all particles
  - Update energy of the particle: $E_{in}^{i} = w_{t+1}^{i} / \mathrm{wbin}[x_{t+1}^{i}]$
  - *Reproduction* or *death* based on the amount of energy $E_{t+1}^{i} = E_{t}^{i} + \left(E_{in} - E_{out}\right)$
    - $E_{t+1}^{i} > \theta$ Make copy of particle and share energy
    - $E_{t+1}^{i} < 0$ Take particle out of the population
- Stable niche size when $E_{\mathrm{in}} = E_{\mathrm{out}}$

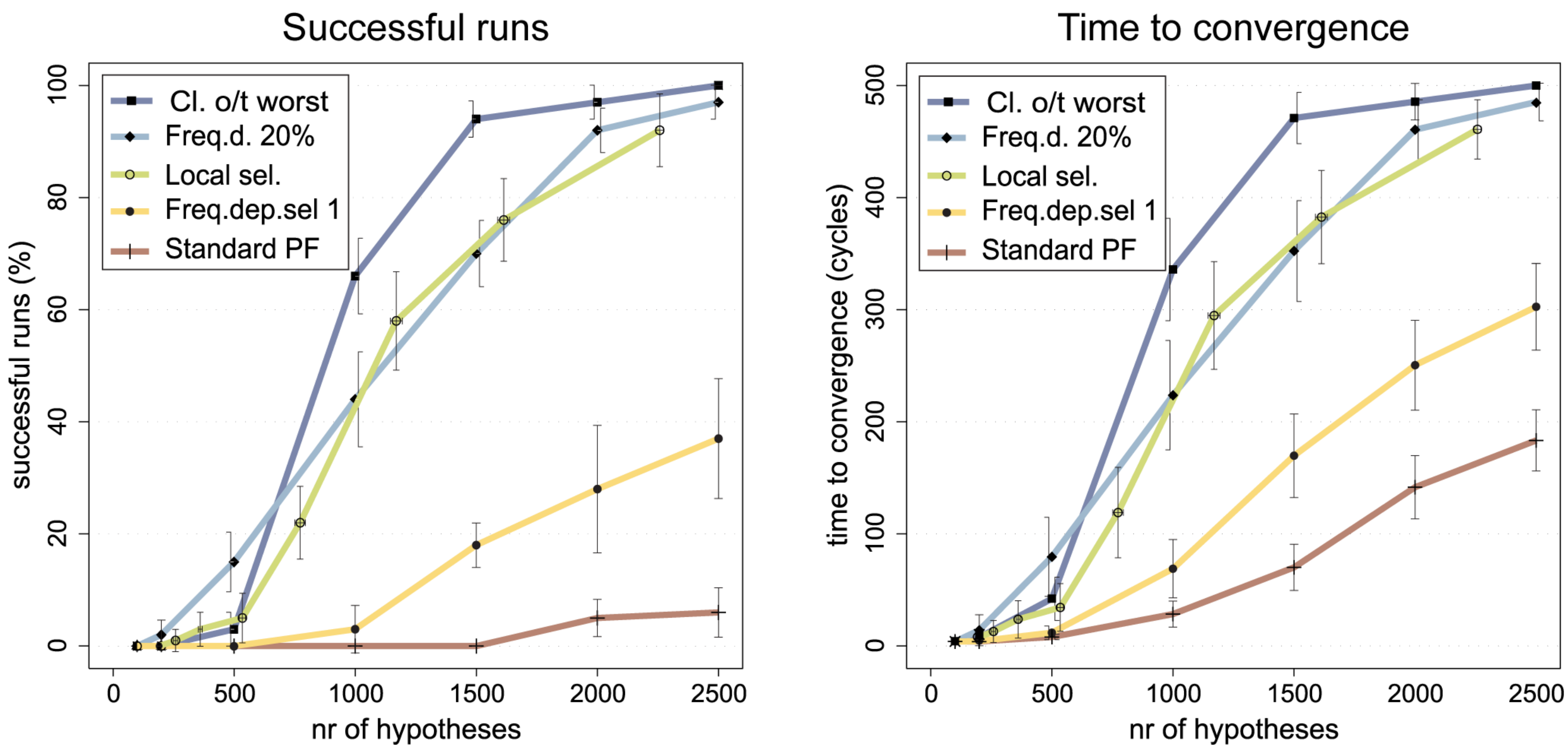
Experiments

- Test the algorithms in a highly symmetrical environment
  - Particle filter needs to maintain all four possible solutions
  - Subpopulations should be compact
  - Estimation error should be small



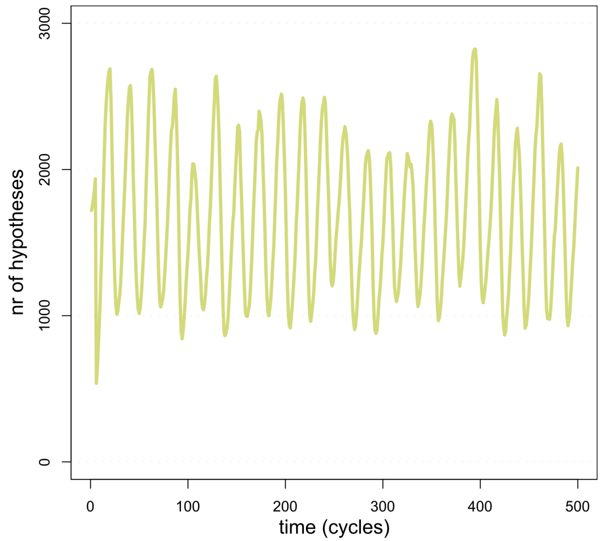
- Demonstration

Results: Diversity maintenance

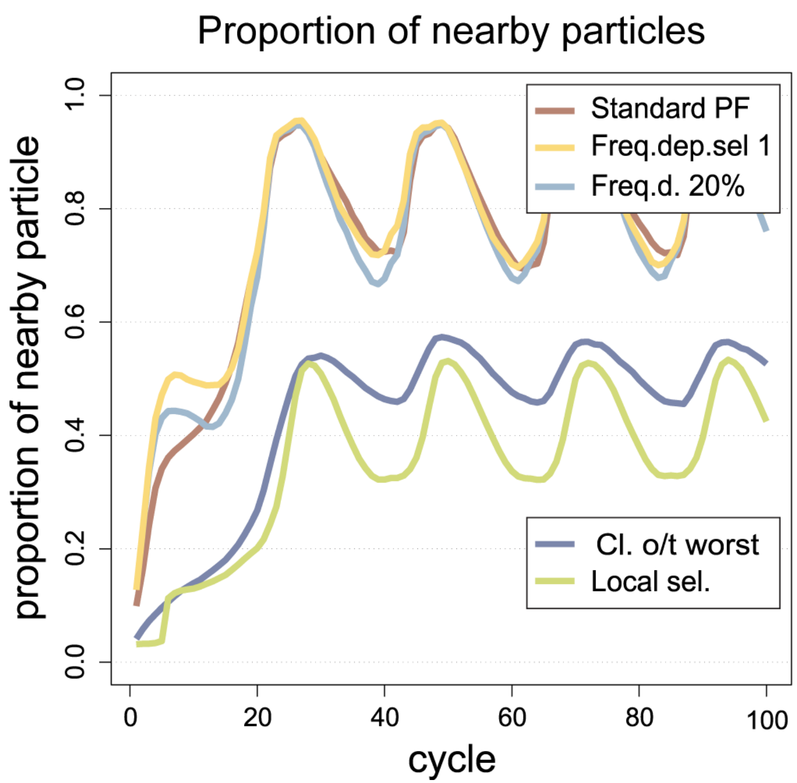


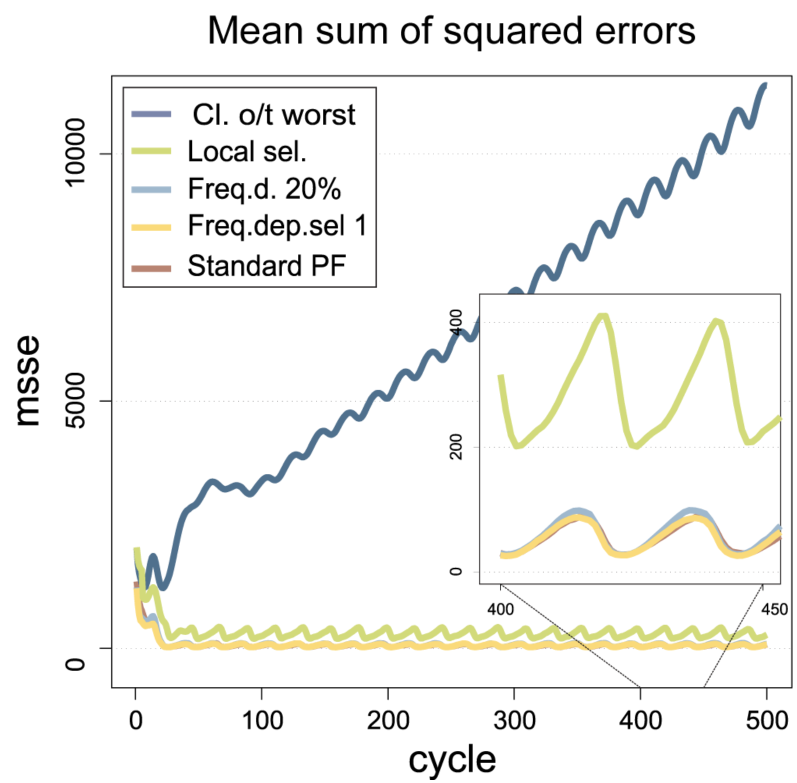
Results: Diversity maintenance

- Standard particle filter
  - Poor diversity maintenance performance
- Crowding (closest of the worst)
  - Best performance
  - O(gg.cf.N^2), in our example faster than the standard
- Frequency dependent selection
  - Good performance
  - O(χ N^2), little overhead
- Local selection
  - Good performance
  - O(N), but N varies somewhat



Results: Compactness

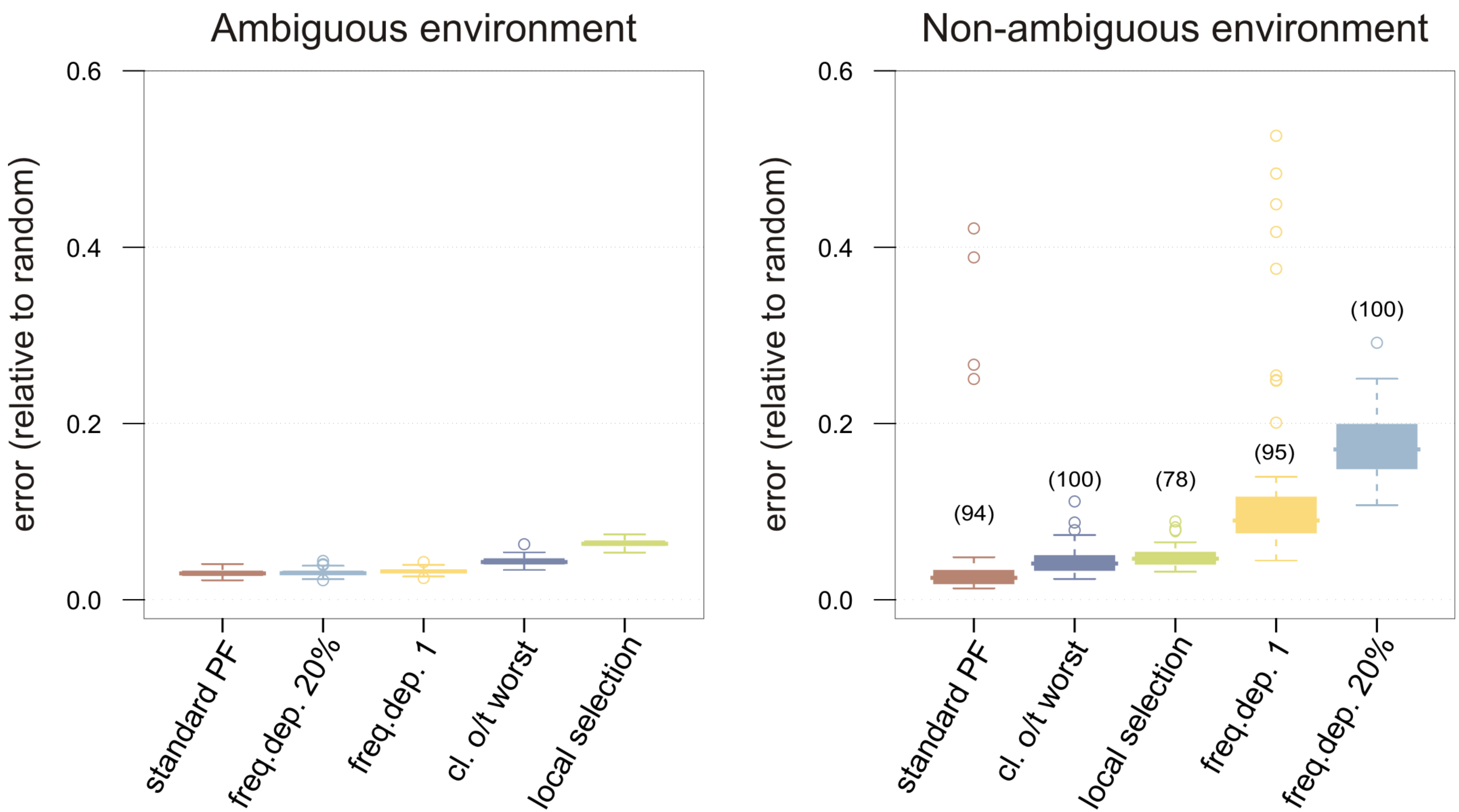




Results: Compactness

- Standard particle filter
  - Most compact
- Crowding
  - Not very compact and problem with loose particles
- Frequency dependent selection
  - Compactness similar to particle filter
- Local selection
  - Less compact

Results: Estimation error



Results: Estimation error

- Standard particle filter
  - Best estimation (NB not taking premature convergence into account)
- Crowding / Closest o/t Worst
  - Very good estimation in both environments
- Frequency dependent selection
  - Very good in ambiguous, but suffers from ghost clusters in non-ambiguous environments
- Local selection
  - Good estimation in both environments

Conclusions and discussion

- Premature convergence is a problem in particle filters
  - For localization, as demonstrated
  - But also for particle filters used in SLAM (FastSLAM)
- Particle filters and genetic algorithms are very similar
- Niching methods can successfully be used in PFs
- Problems of loose particles and ghost cluster can be overcome

Questions?

- Kootstra, G. & de Boer, B. (2009) Tackling the Premature Convergence Problem in Monte-Carlo Localization. *Robotics and Autonomous Systems* 57(11): 1107-1118.

- kootstra@kth.se

Crowding

- for all particles *i*
  - Apply motion and sensor model
- end
- $G_t \leftarrow \mathrm{SAMPLE}(S_t, gg \cdot N)$
- for all particles i in G_t
  - $\hat{S}_t \leftarrow \mathrm{worst\_particles}(S_t, N/3)$
  - $C \leftarrow \mathrm{uniform\_sample}\left(\hat{S}_t, cf \cdot N\right)$
  - $j \leftarrow \arg\min_{k \in C}\left(dist(i,k)\right)$
  - $p_t^j \leftarrow p_t^i$
- end