DAIMLERCHRYSLER

Competition and Cooperation in Extended Evolutionary Algorithms

Hartmut Pohlheim

DaimlerChrysler AG, Research & Technology
Methods and Tools, FT3/SM

Alt-Moabit 96a, 10559 Berlin, hartmut.pohlheim@daimlerchrysler.com

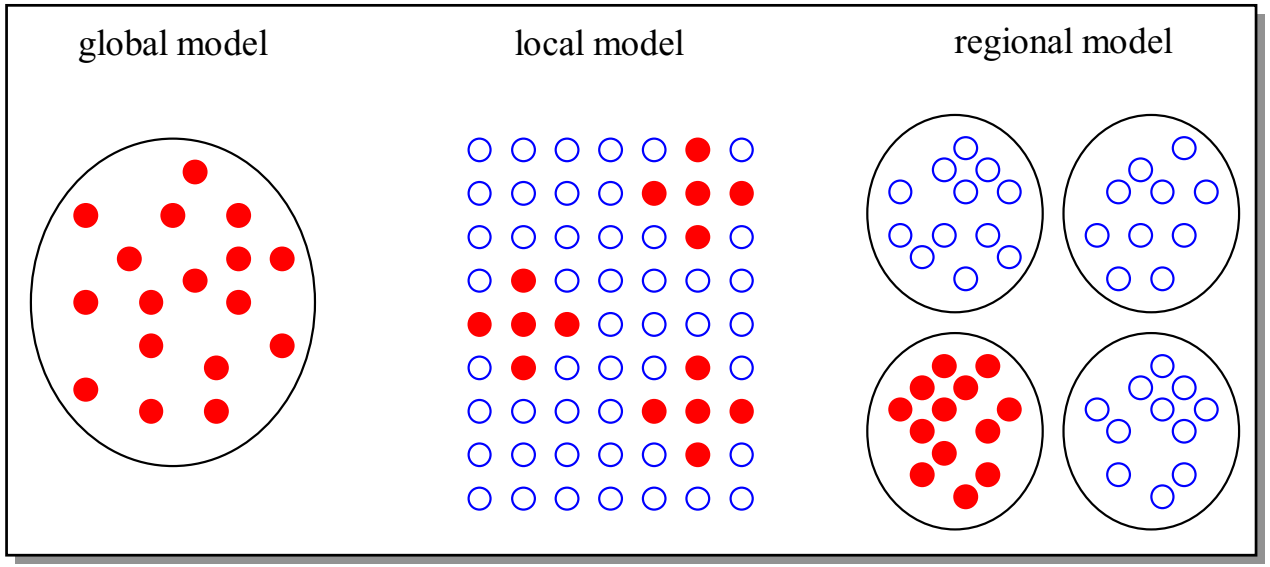
Overview

1. Motivation
   - → Classification of population models
   - → Definition and structure of Evolutionary Algorithms
2. Extensions of Evolutionary Algorithms
   - → Application of different strategies
   - → Competition between subpopulations
   - → Application of competing subpopulations
   - → Advantages of extended Evolutionary Algorithms
3. Summary and real-world applications

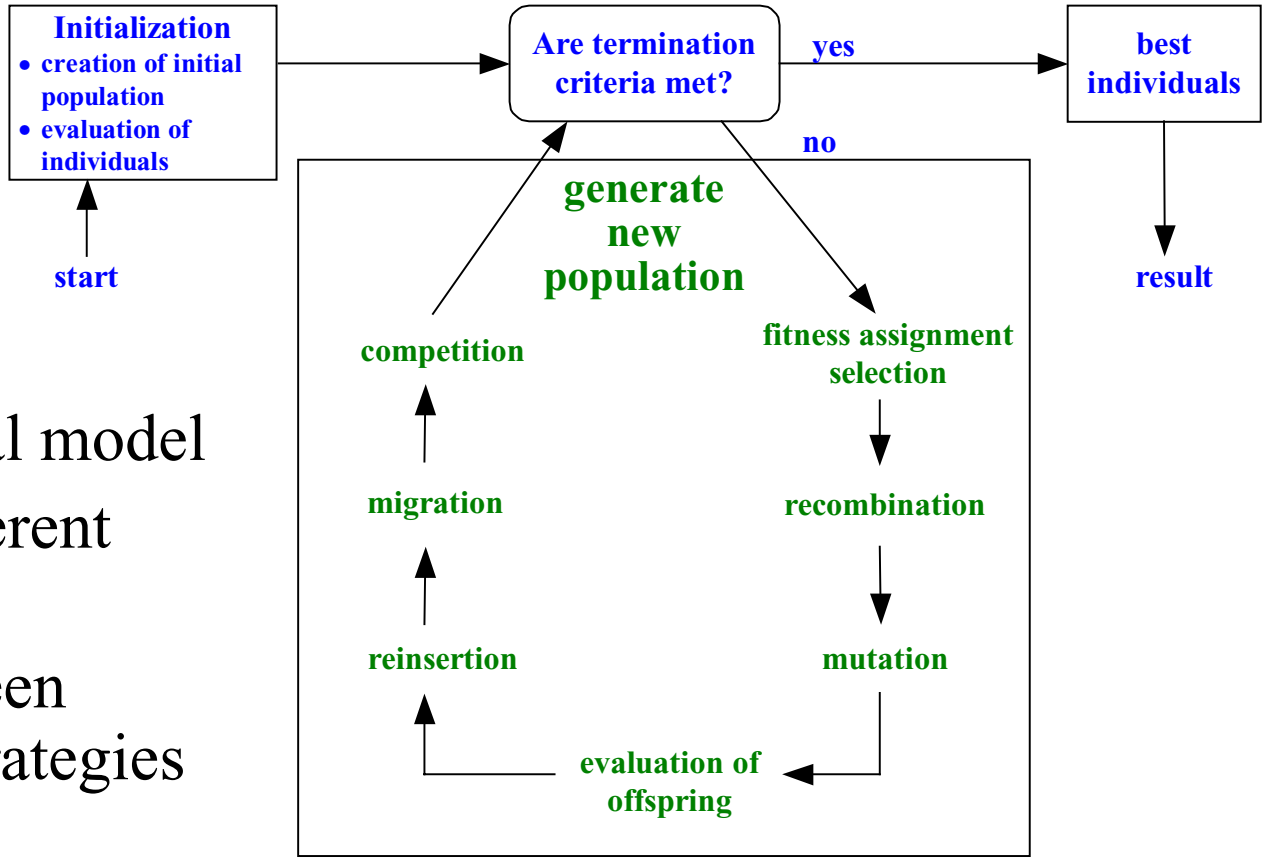
1 Population models

Division of population by range of selection

- global model (master/slave model)
- local model (diffusion, neighbourhood, or fine grained model)
- regional model (migration, island, or coarse grained model)



2 Extended Evolutionary Algorithms



Extensions of regional model

- Application of different strategies
- Competition between subpopulations / strategies

2.1 Extension: Application of different strategies

Standard:

all subpopulations in the regional model use the same parameters

Extension: each subpopulation uses its own strategy

→ different parameters for each subpopulation
(for instance, different recombination and/or mutation operators)

→ thus: exploration of the search space with different strategies
(rough/fine, globally/locally oriented, ...)

Calculation of order of subpopulations (ranking)

→ based on fitness values of individuals (fitness assignment)

→ combine fitness of individuals to an evaluation of the subpopulation
(e.g. calculate average fitness value of the individuals in a subpopulation)

2.2 Extension: Competition between subpopulations

Extension:

Competition for resources between subpopulations

→ good/successful subpopulations receive more resources

→ less good/unsuccessful subpopulations receive less resources

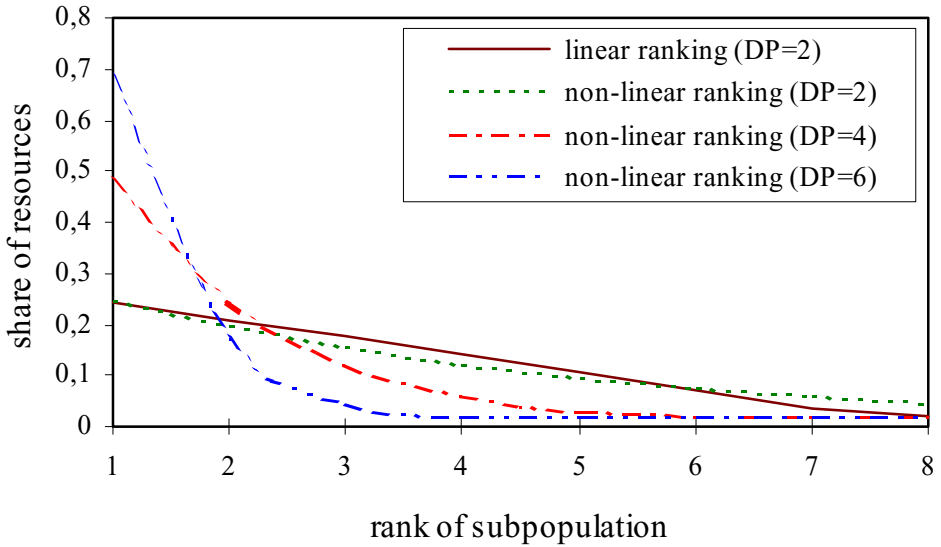
⇒ good subpopulations receive individuals from unsuccessful subpopulations

- Parameter
  - → **Division of resources** (depending on the ranking of the subpopulations)
  - → **Competition interval** (frequency of a competition for resources)
  - → **Competition rate** (maximum share of resources which has to be transferred)
  - → **Subpopulation minimum** (minimum size of an unsuccessful subpopulation)
  - → Robust settings of these parameters are known

2.3 Division of resources

- Similarities with fitness assignment
  - → individuals: objective value → rank → fitness value
  - → subpopulationens: rank → part of the available resources

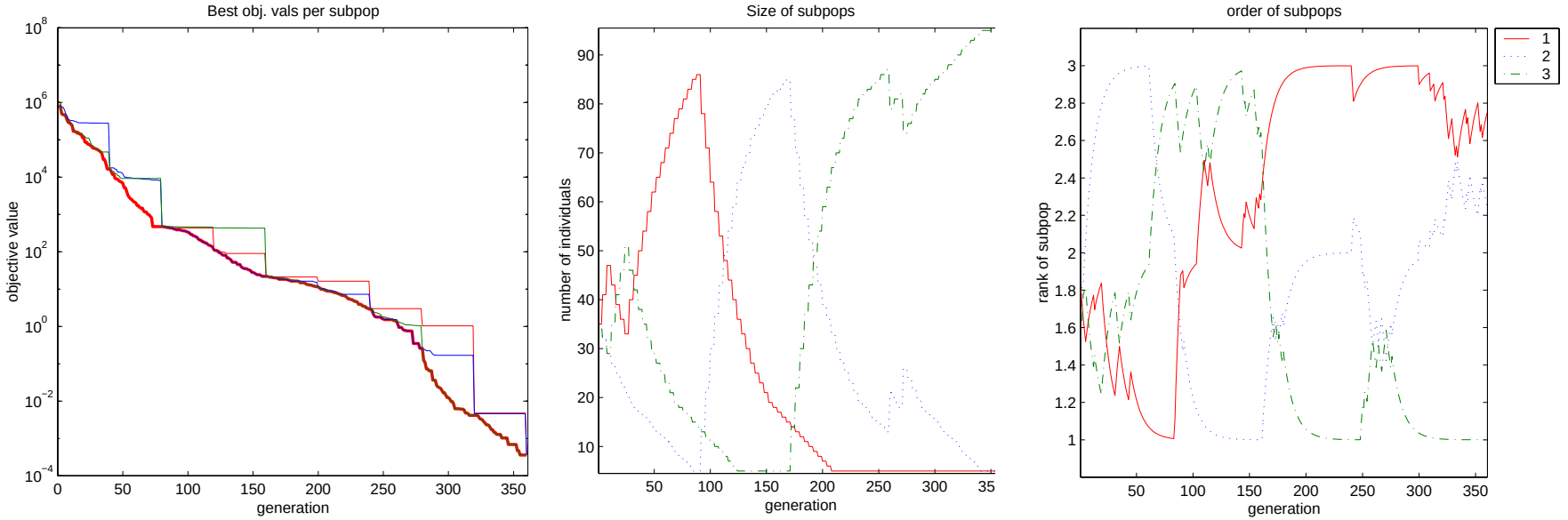
- Method:
  - → Apply fitness assignment methods directly (check subpopulation minimum)
  - → Instead of selection pressure: **Division pressure DP**



2.4 Example of competing subpopulations

- 3 subpopulations using different strategies
  - → subpopulation 1: rough search (big mutation steps)
  - → subpopulation 3: fine search (small mutation steps)

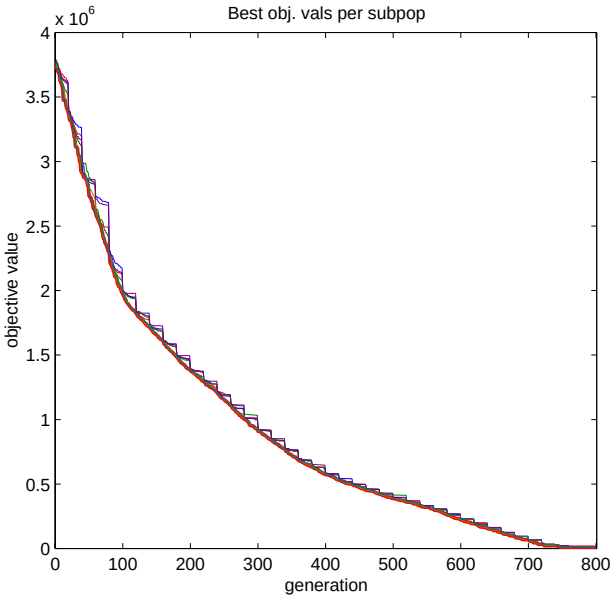
(RASTRIGIN's function 6, $n = 20$, $-500 < x < 500$)

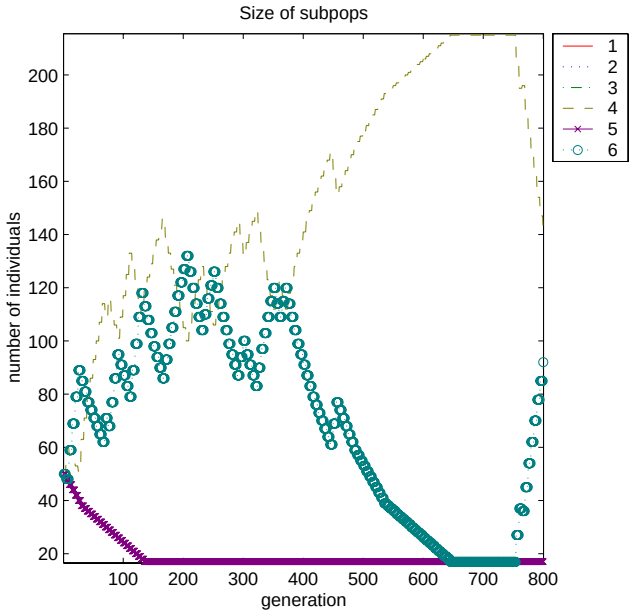


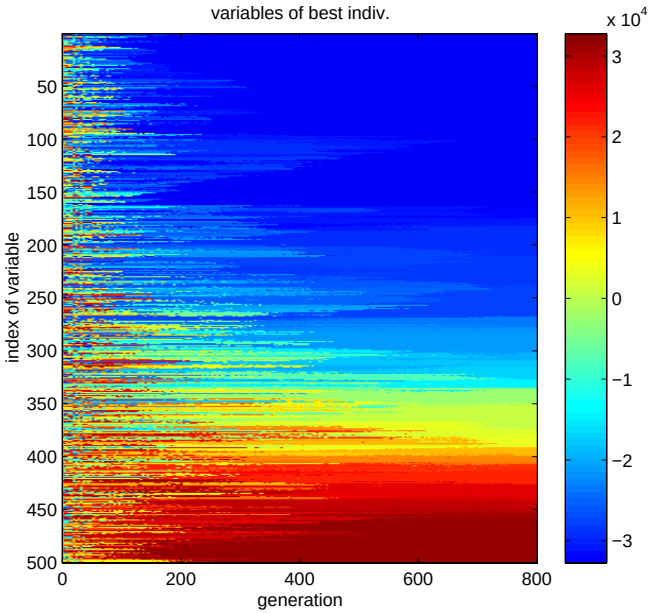
2.5 Application of competing subpopulations

- 6 subpopulations with different strategies
  - → 2 mutation operators, each employing rough, middle and fine search

(temporal testing of software module Bubblesort, $n = 500$)







2.6 Advantages of Extended Evolutionary Algorithms

Application of different strategies:

- → simultaneous testing of a number of parameter settings
- → identify which strategy is at which point of time successful
- → cooperation between strategies (better results than separated strategies)

Competing subpopulations:

- → automatic resource distribution between subpopulations / strategies
- → indirect adaptation of the strategy parameters during an optimization run
- → efficient use of resources

3 Summary

- Extensions for population models
  - → Application of different strategies
  - → Competing subpopulations

  - ⇒ **step towards the development of powerful Evolutionary Algorithms**
  - ⇒ **especially for the solution of large and complex problems**

- Implementation
  - → GEATbx: Genetic and Evolutionary Algorithm Toolbox for Matlab, v. 3.x
    http://www.geatbx.com/

3 Application of Extended Evolutionary Algorithms

DaimlerChrysler AG:

- Search for maximum and minimum run time of software moduls
- Automatic test data generation for structural software testing
- Testing of safety critical software systems

Dissertation:

- Optimal climate control in greenhouses