PA184 - Heuristic Search Methods

Lecture 5 – Evolutionary Algorithms (EAs)

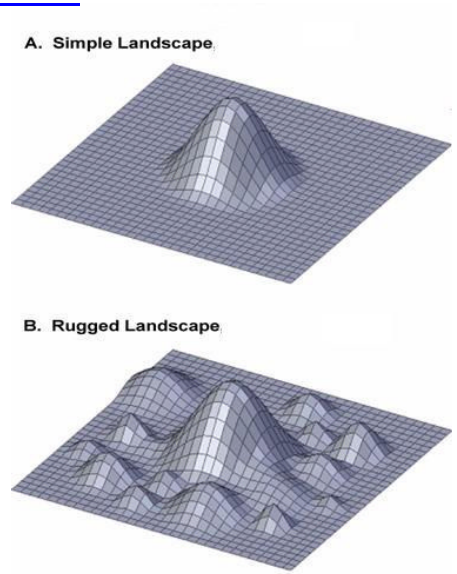
- Basics of Evolutionary Algorithms
- Design of Evolutionary Algorithms
- Examples of Practical Evolutionary Algorithms

Learning outcomes:

- Understand key components and working principle of EAs
- Describe the role of each component in an EA
- Appreciate the different ways to implement EAs
- Understand the basics of some representative EAs

Basics of Evolutionary Algorithms

The rationale for evolutionary algorithms (EAs) is to maintain a population of solutions during the search. The solution (individuals) compete between them and are subject to selection, and reproduction operators during a number of generations.



Exploitation vs. Exploration

- Having many solutions instead of only one.
- Survival of the fittest principle.
- Pass on good solution components through recombination.
- Explore and discover new components through self-adaptation.
- Solutions are modified from generation to generation by means of reproduction operators (recombination and self-adaptation).

Darwinian Evolutionary Algorithm

- Competing population
- Dynamic population
- Fitness
- Dynamic genetic inheritance

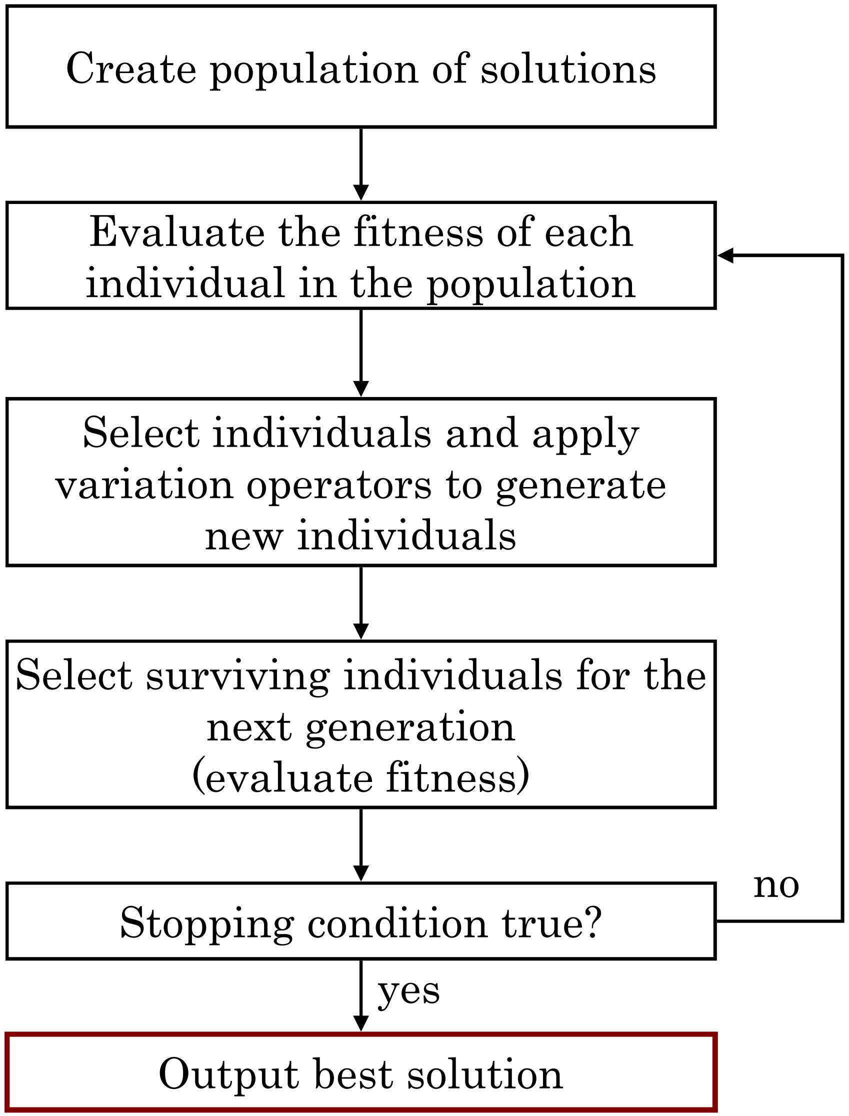
Lamarckian Evolutionary Algorithm

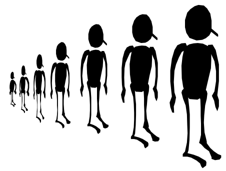
- Competing population
- Dynamic population
- Fitness
- Dynamic cultural inheritance

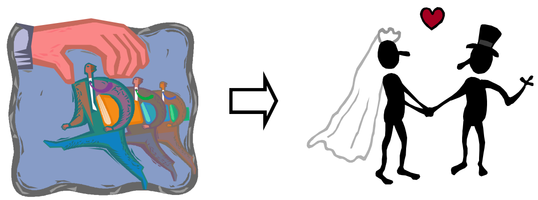
Baldwinian EAs use a kind of intermediate between genetics and culture suggesting the existence of a strong interaction between learning and evolution.

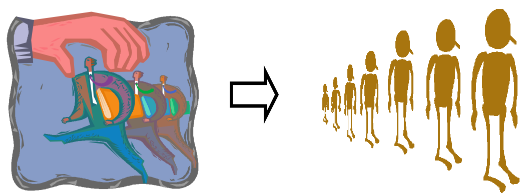
Dawkins proposed the concept of memetics to refer to cultural heritage.

Typical Evolutionary Algorithm Cycle









Design of Evolutionary Algorithms

Typical Components of an EA

- Solution representation
- Population size
- Initialisation strategy
- Evaluation function
- Reproduction operators
- Selection mechanism
- Stopping condition to detect convergence

A diverse range of evolutionary procedures can be designed for the subject problem by tuning the above components.

It is crucial to design components and tune parameters while considering the choices made on the various parts of the algorithm.

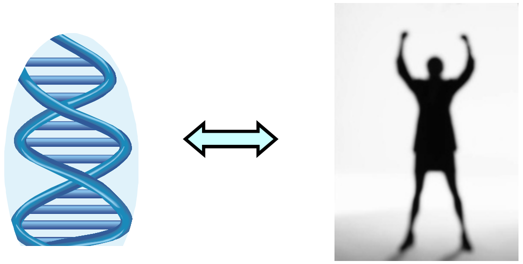
Solution Representation (encoding)

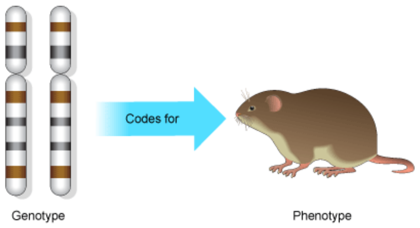
A good representation ensures that variation operators maintain a functional link between current and new solutions.

Some common representations:

- Binary vectors
- Permutations
- Symbolic (e.g. parse trees)

The representation can be genotypic of phenotypic and then the Hamming or Euclidean distance can be used to measure the distance between solutions.





Population Size and Initialisation Strategy

Commonly, μ denotes number of parents and λ denotes number of offspring, the new population is obtained from the $\mu + \lambda$ individuals.

Common initialisation strategies:

- Uniformly at random
- Incorporating problem domain knowledge
- Incorporating some elite solutions

The number of parents λ can be seen as the degree of parallelism of the evolutionary search.

The number of offspring μ can be seen as the degree at which the parents as used as the basis to generate new individuals.

The difficulty of the fitness landscape must be taken into account when setting the population size parameters μ and λ.

Evaluation Function

A good evaluation function assesses the quality of evolved solutions and helps to assess the effectiveness of the reproduction operators.

Usually, evaluating fitness is a time-consuming process in EAs so delta and/or approximate evaluation might be useful.

Moreover, it might be enough to have a way to discriminate between individuals without conducting an exact fitness evaluation.

Reproduction Operators

The design of reproduction operators must be done according to the solution representation and the fitness landscape.

Some common operators (recombination and mutation):

- Changing solution component values
- Perturbing solution component values
- Combining complete solutions
- Blending solution component values

When designing reproduction operators ⇒ allow infeasibility?

Mutation: some variation of the parent after cloning.

Recombination: components of parents are cloned then combined.

Note: reproduction operators within an EA can be applied to produce 1 or more offspring at the same time.

Selection Mechanism

Selection is applied in two contexts: for deciding which individuals will reproduce to generate new solutions (selection for reproduction) and for deciding which individuals will survive to the next generation (selection for survival).

Some issues to consider in the selection mechanism:

- Sampling with or without replacement?
- Generational or steady-state?
- Elitism strategy?
- Deterministic of stochastic?
- In stochastic selection: proportional, tournament or ranking?

Common notation: μ denotes number of parents and λ denotes number of offspring. Some selection strategies: $(\mu+\lambda)$, (μ,λ)

In terms of selection pressure, deterministic selection is the most stringent while proportional selection is the least.

Examples of Evolutionary Algorithms
(and other population-based algorithms)

- Genetic Algorithms (GA)
- Evolutionary Strategies (ES)
- Genetic Programming (GP)
- Ant Colony Optimisation (ACO)
- Memetic Algorithms (MA)
- Particle Swarm Optimisation (PSO)
- Differential Evolution (DE)
- Estimation of Distribution Algorithm (EDA)
- Cultural Algorithms (CA)

Since this a very active research area, new variants of EAs and other population-based meta-heuristics are often proposed in the specialised literature.

Evolutionary Multi-objective Simulated Annealing

Multiobjective Optimisation Problem

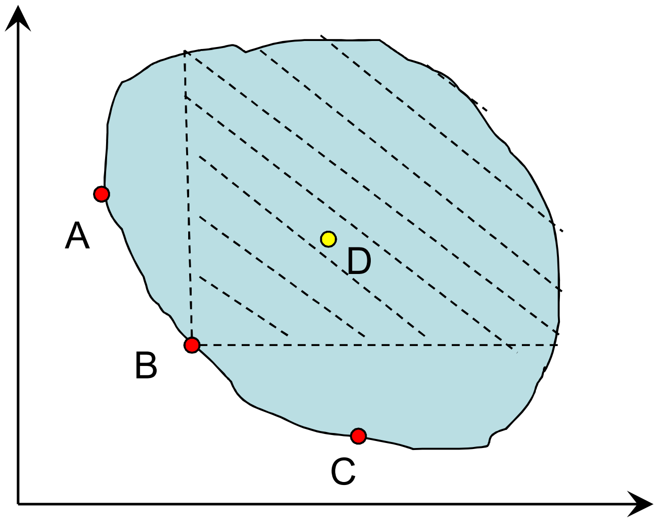
Minimize $\{f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\}$
s.t. $\mathbf{x}$ in $\boldsymbol{\Omega}$

Dominance

Solution **x** is said to dominate solution **y**, if and only if the function values of **x** is not worse than those of **y** in all objectives, and there exists one objective, in which **x** is better than **y**.

Non-dominance

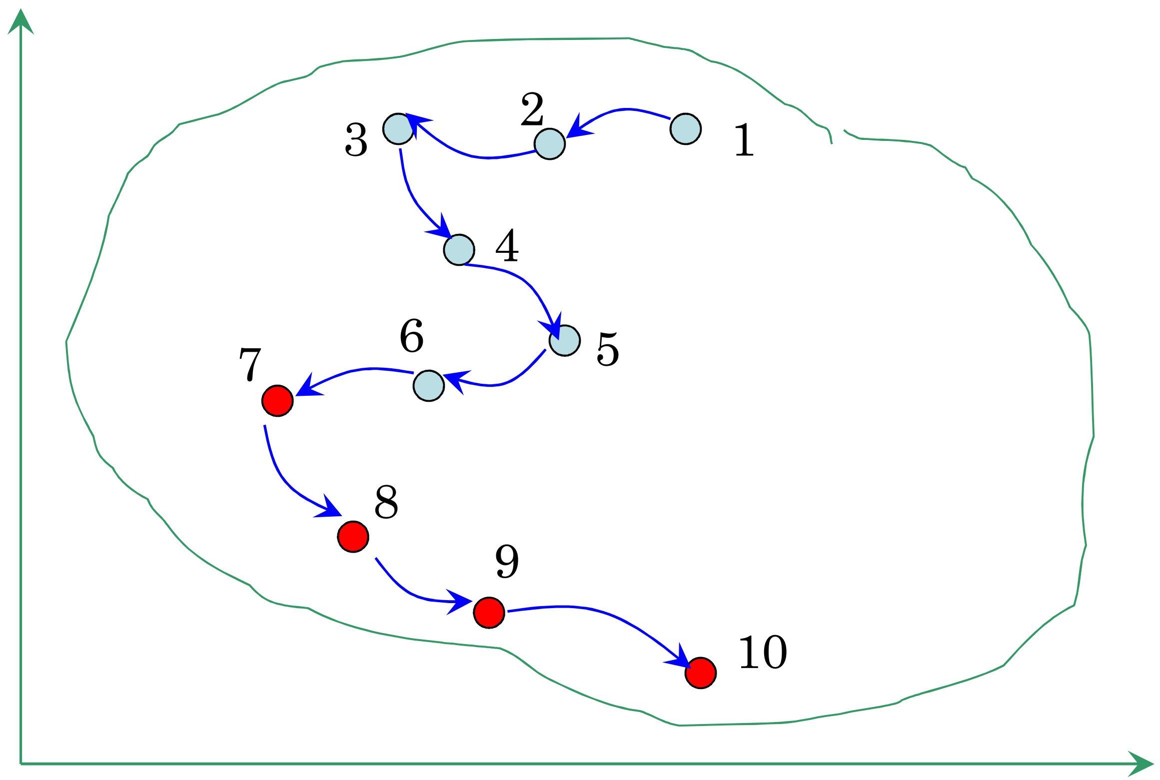
Solution **x** is said to be nondominated to solution **y** if **y** doesn't dominate **x**.



Pareto Optimality

The Pareto-optimal set of multi-objective optimisation problem is the set all solutions which are nondominated to any other solution in the search space.

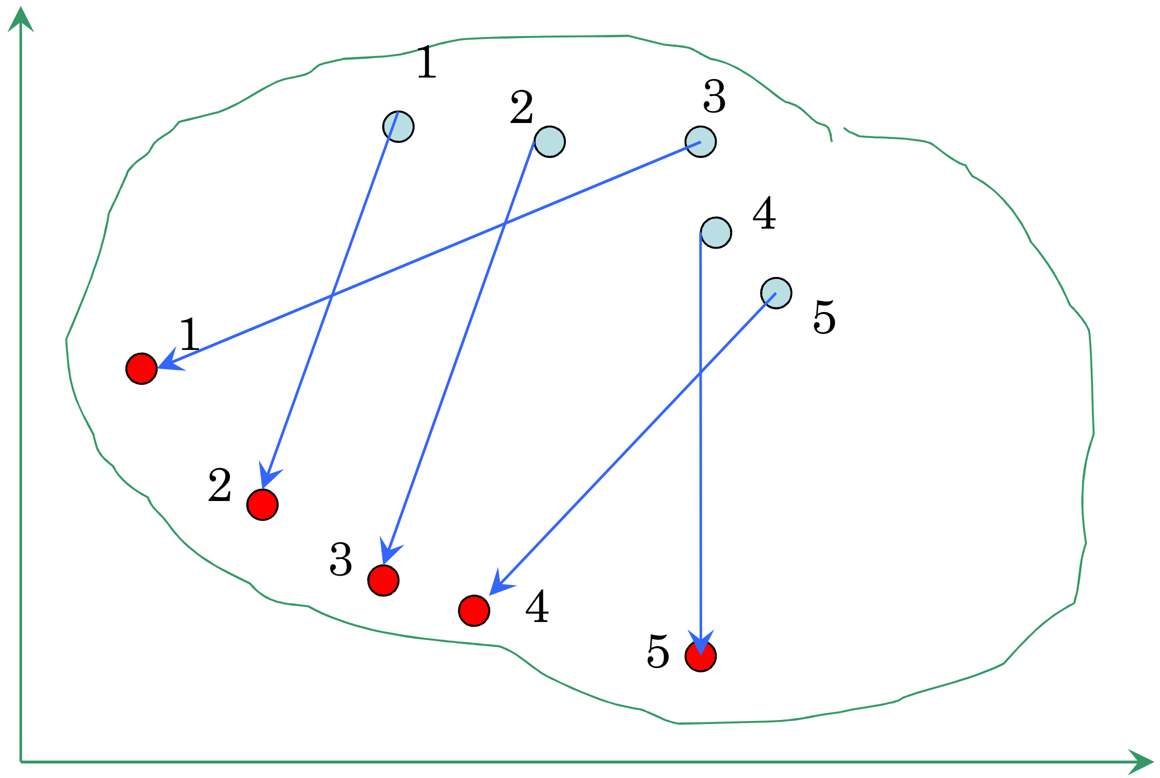
Search Directions in Serafini's MOSA



Idea: change the search direction randomly and slightly to explore the whole Pareto-optimal front

Weakness: easy to get stuck in the local part of the Pareto front

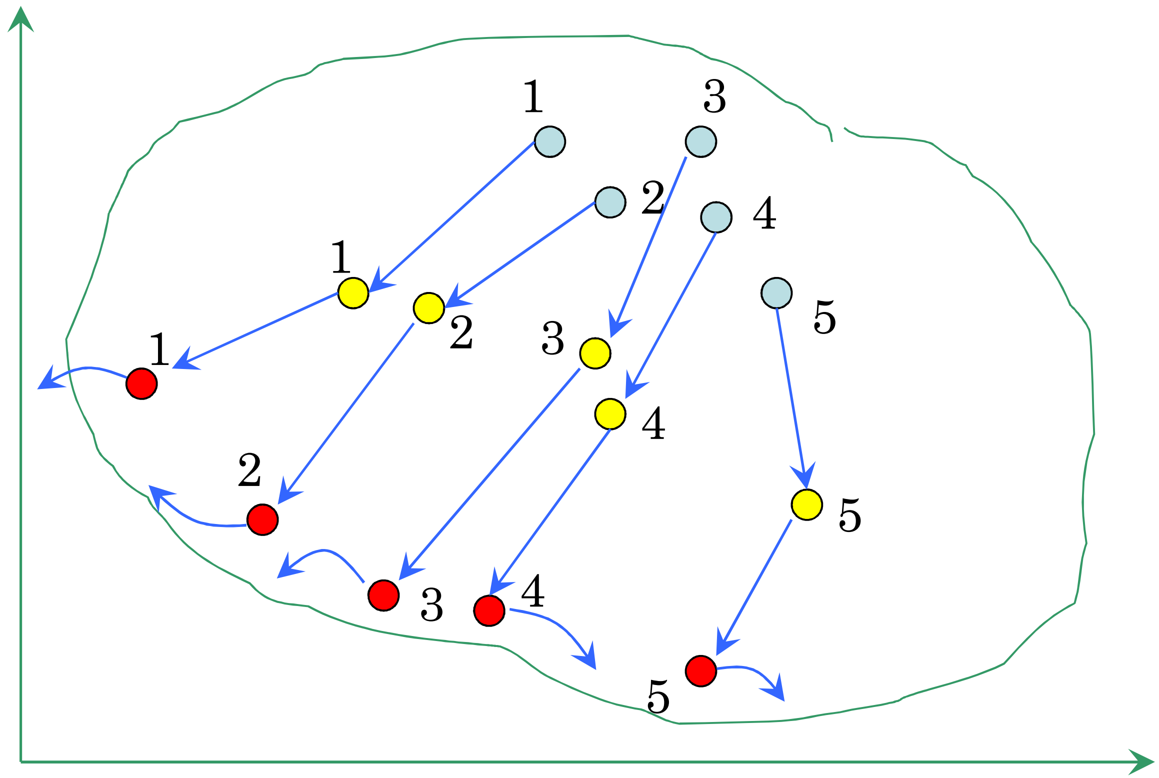
Search Directions in Ulungu's MOSA



Idea: maintain multiple fixed search directions and approximate corresponding Pareto-optimal solution separatively

Weakness: no ability to fill the gap between the solutions in the spare part of nondominated front

EMOSA: Two-stage Strategy of Tuning Search Directions



Stage 1: all search directions are fixed at each early temperature level when T is bigger than Tc

Stage 2: the search direction of each solution is tuned at each late temperature level which T is smaller than Tc

Framework of EMOSA

Step 0: Initialisation

Produce *pop* well-distributed weight vectors. For each weight vector, an initial solution is generated randomly. Update the external population (EP) with the nondominated solutions in the initial population. Set T:=T0.

Step 1: Local Search and Competition

For each individual x in the current population, repeat the follow steps K times.

1.1 find a neighbouring solution y

1.2 update the EP with y if it is not dominated by x

1.3 replace x with the probability P(w, x, y, T)

Compete with the other solutions with similar weight vectors to that of x

Step 2: Temperature Change

Lower the temperature value by using T:=T – alpha. If $T>=T_c$, adapt the weight vector of each individual in the population, otherwise, go to Step 4.

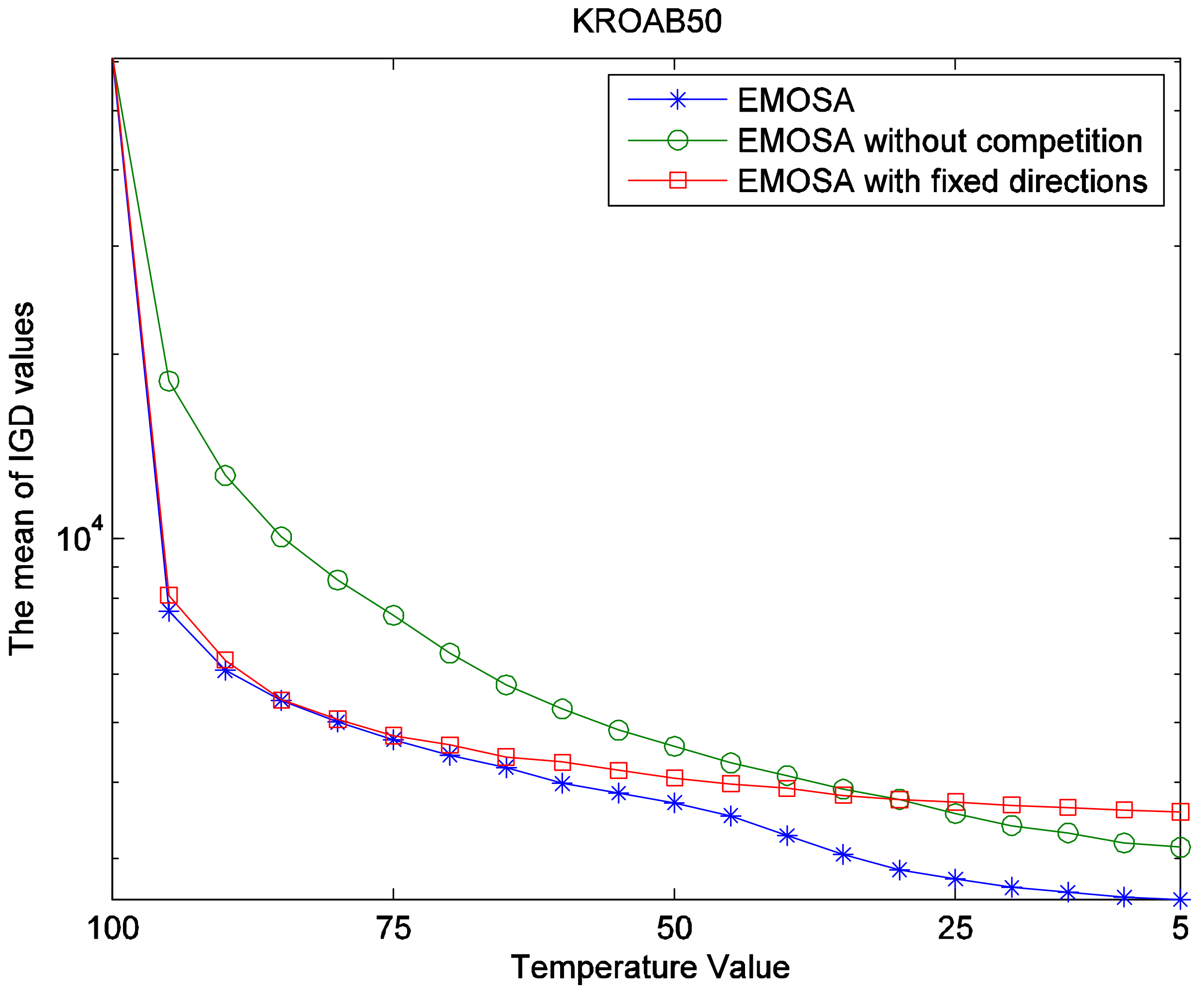
Step 3: Direction Adaptation

Modify each weight vector to move the current solution away from its nearest nondominated neighbours in the population.

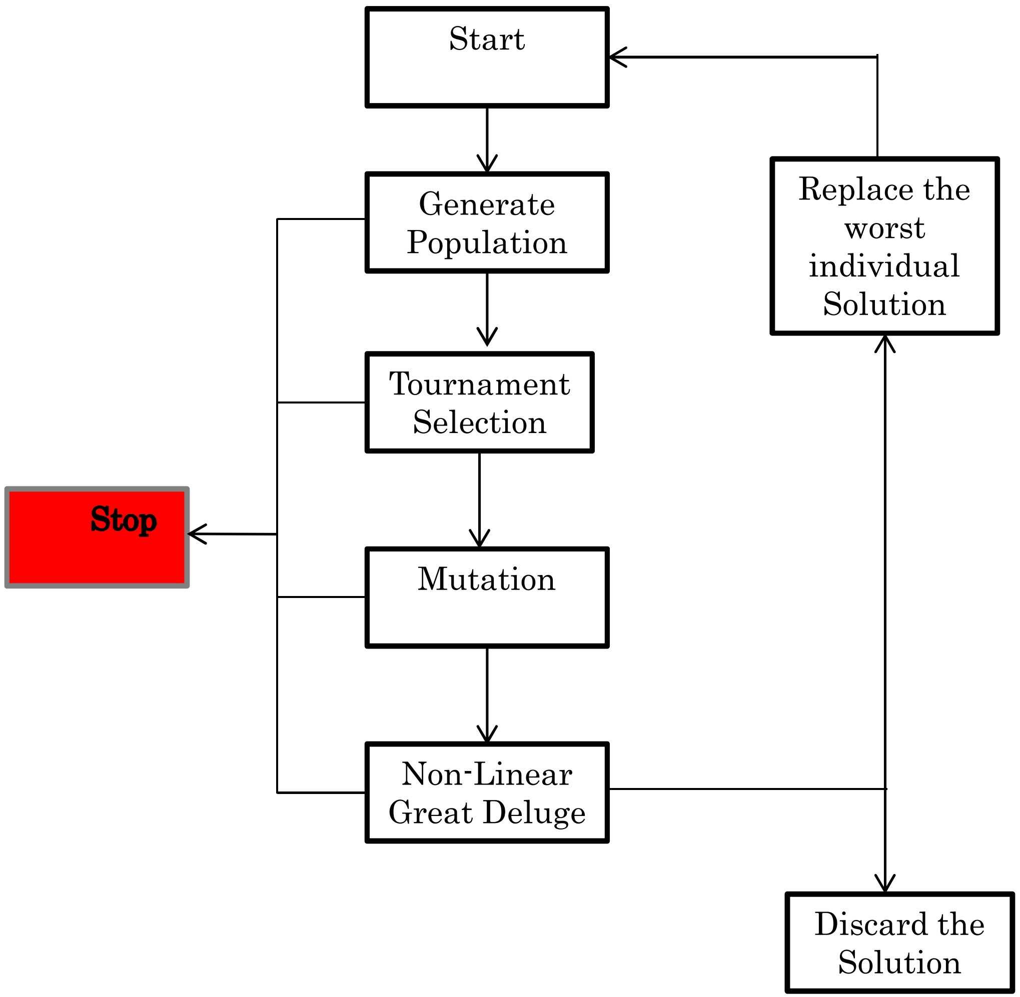
Step 4: Stopping Criteria

If $T<T_{min}$, then stop and return EP, Otherwise go to Step 1.

Impact of Adaptive and Competitive Search Directions



ENLGD: Evolutionary Non-linear Great Deluge



With probability 0.5, the mutation operator is applied to the solution selected from the tournament. The mutation operator selects at random 1 out of 3 types of neighbourhood moves while maintaining feasibility:

Move M1. Selects one event at random and assigns it to a feasible pair timeslot-room also chosen at random.

Move M2. Selects two events at random and swaps their timeslots and rooms while ensuring feasibility is maintained.

Move M3. Select three events at random and exchange the position of the events at random and ensuring feasibility is maintained.

Results Obtained by the ENLGD Approach

Instances	NGD	ENGD(-m)	ENGD	Best Known
Small1	3	3	0	**0** (VNS-T)
Small2	4	2	1	**0** (VNS-T)
Small3	6	2	0	**0** (CFHH)
Small4	6	3	0	**0** (VNS-T)
Small5	0	0	0	**0** (MMAS)
Medium1	140	157	**126**	146(CFHH)
Medium2	130	178	**123**	147(HEA)
Medium3	189	240	**185**	185(246)
Medium4	112	152	116	164.5(MMAS)
Medium5	141	142	**129**	130(HEA)
Large	876	995	821	**529**(HEA)

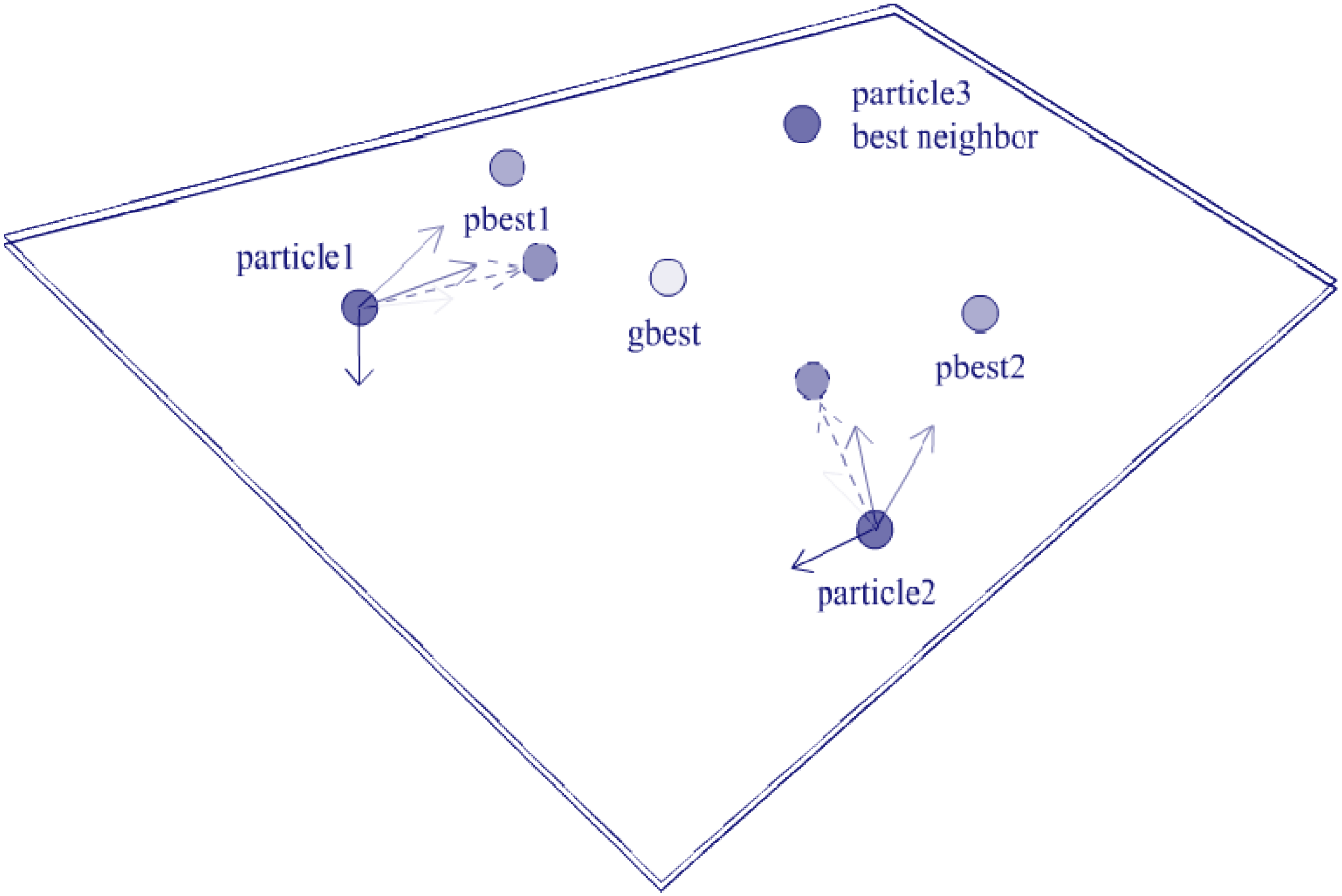
Particle Swarm Optimization

- Mechanism:
  - Set of particles initially in different positions. (solutions)
  - A particle knows: its position, its best position so far, best position in the neighborhood at the current iteration and the best position in the swarm.
  - Particles update their position by using two equations:

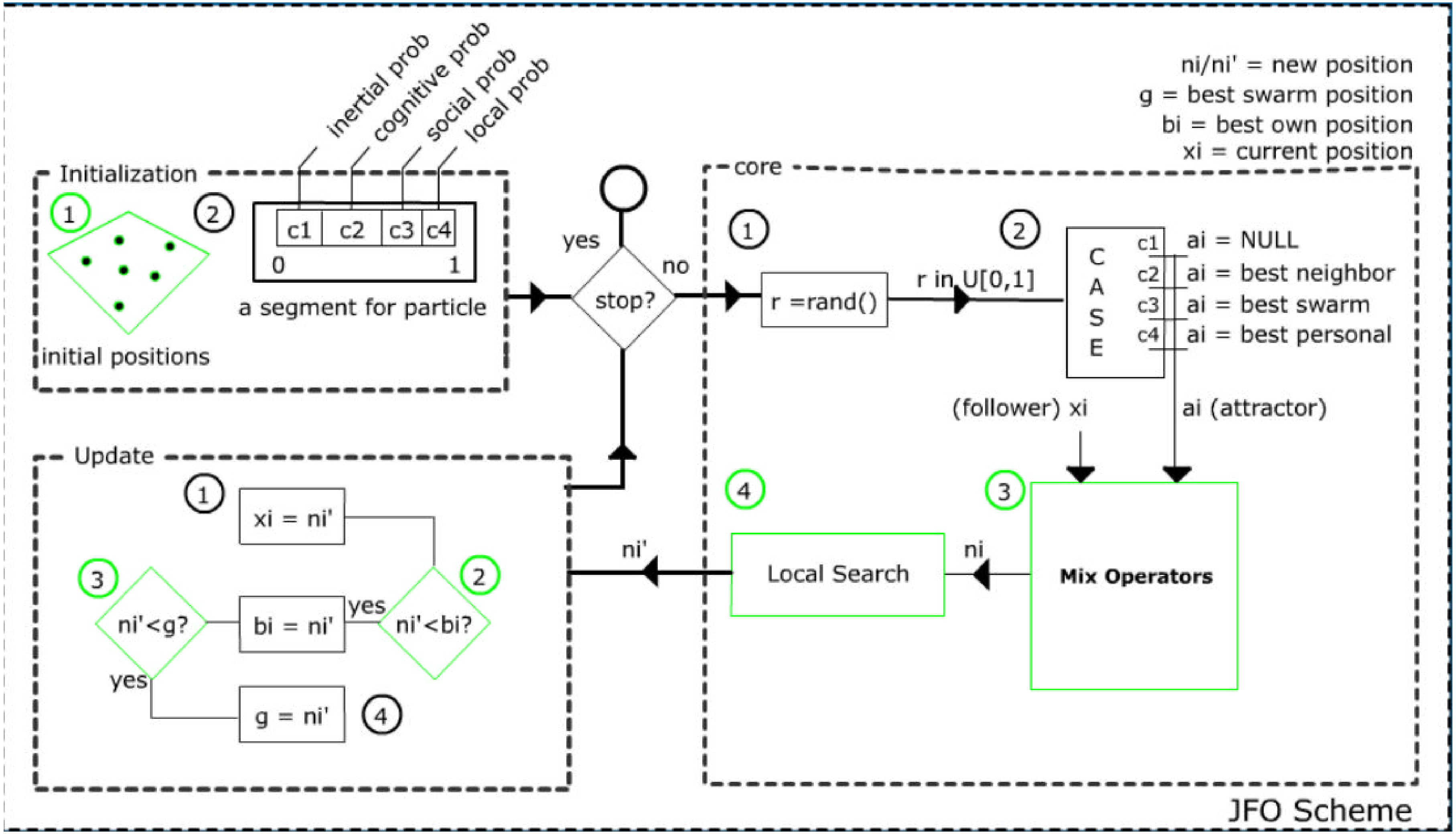
$$v_{i,j} \leftarrow c_0 v_i + c_1 r_1 \left(g_j - x_{i,j}\right) + c_2 r_2 \left(b_{i,j} - x_{i,j}\right) + c_3 r_3 \left(g_{i,j} - x_{i,j}\right)$$

$$x_{i,j} \leftarrow x_{i,j} + v_{i,j}$$

Set of Leaders Guide the Search in PSO



Implementation of a Discrete PSO (DPSO)



Additional Reading

Chapter3 of (Talbi, 2009)

Chapters 5 and 7 of (Michalewicz,2004)

Chapters 1,3 and 4 of (De Jong,2006)

A. Hertz, D. Klober. *A framework for the description of evolutionary algorithms*. European journal of operational research, 126(1), 1-12, 2000.

C. Reeves. *Genetic algorithms for the operations researcher*. INFORMS Journal on computing, Informs, 9(3), 231-250, 1997.

P. Chu, J. Beasley. *A genetic algorithm for the generalised assignment problem*. Computers and operations research, 24(1), 17-23, 1997.

Seminar Activity 5

The purpose of this seminar activity is to outline a number of evolutionary operators for the GAP.

Do the following:

1. Read some of the literature describing Evolutionary Algorithms. For example, section 3.3 of (Talbi, 2009), chapter 7 (Michalewicz and Fogel,2004), article Reeves 1997, Chu and Beasley 1997, etc.

2. Describe some ideas to implement the following operators in EAs for the subject problem:

   a) Selection Operators

   b) Mutation Operators

   c) Recombination Operators

   d) Replacement Strategies