Evolutionary and population algorithms

Andrzej Jaszkiewicz

Biological evolution

- Inheritance of traits/features
- Recombination of traits/ features between individuals and effectively in the population
- Mutations
- Natural selection
- Genetic drift

Biological evolution vs. optimization

- Individual - Solution
- Population - Set of solutions
- Genotype - Solution representation
- Phenotype - The value of the objective function and other parameters of interest (e.g. constraint values)
- Mutation - Construction of a new solution through a slight modification of another solution
- Crossover/recombination – Construction of a new solution by combining the features of two solutions
- Natural selection – Probabilistic selection of good solutions
- Fitness – Objective function
- Fitness improvement – Optimization of the objective functions
- Genetic Drift - Convergence and random walk of a population

General scheme of the evolutionary algorithms

Generate a randomized starting population X

repeat

X_1 := Crossover (X)

X_1 := Mutation (X_1)

X := Selection ($X \cup X_1$)

until stopping conditions are met

Return the best generated solution

Gene coding in biology

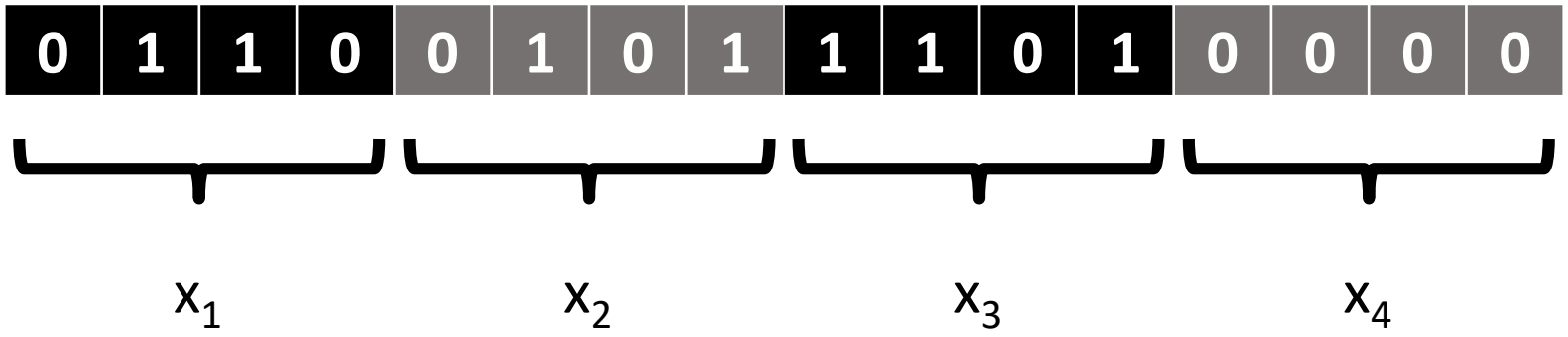
- DNA
- Four-letter alphabet – four nitrogenous bases A, G, T, C
- Three nitrogenous bases encode one amino acid in a redundant way, e.g. codons AAA i AAG encode the amino acid lysine

Genetic algorithms

- Evolutionary algorithms with binary encoding
- Solution represented as a sequence(string, vector) of zeros and ones
- Always possible (with some accuracy), but not always natural

Binary encoding of numeric variables

E.g.:



Binary encoding example for the knapsack problem

- One position for each item
- 0 – item not selected
- 1 – item selected

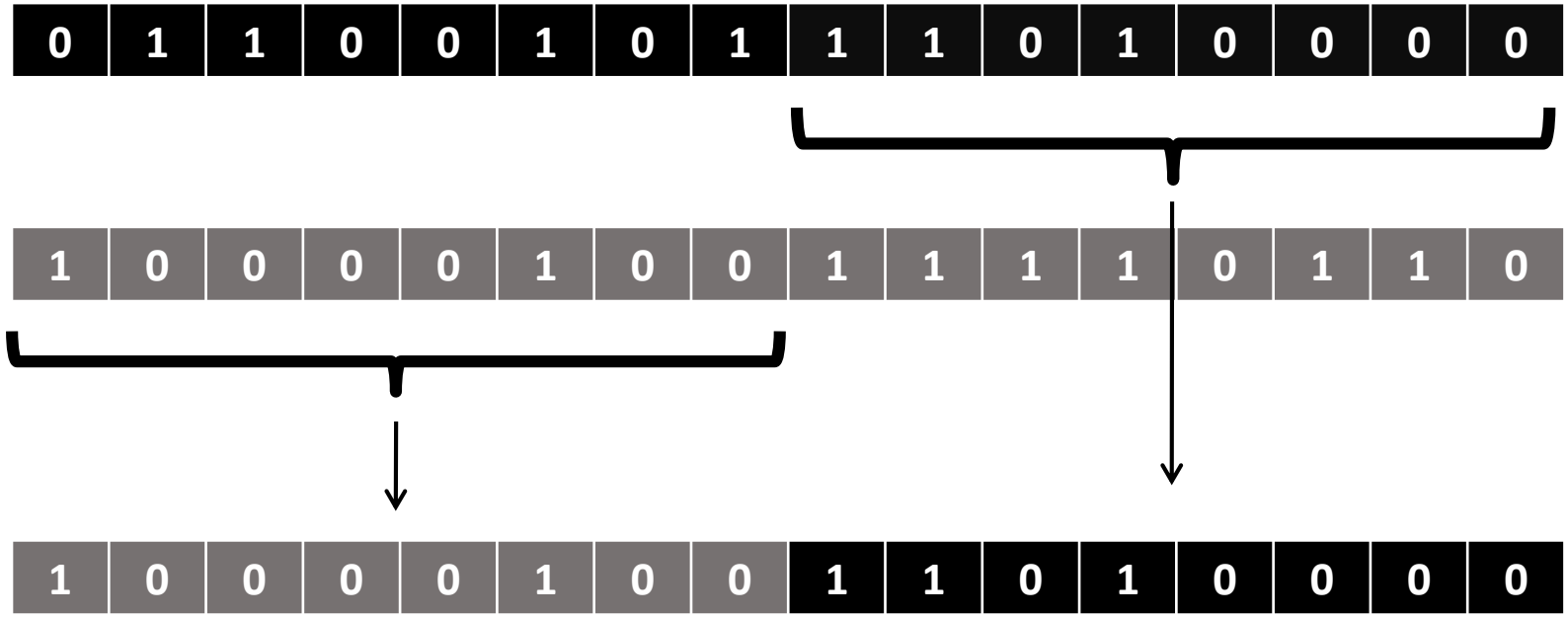
- Not every binary string encodes a feasible solution – the weight of the selected items may exceed the capacity

Example of binary encoding for traveling salesperson problem

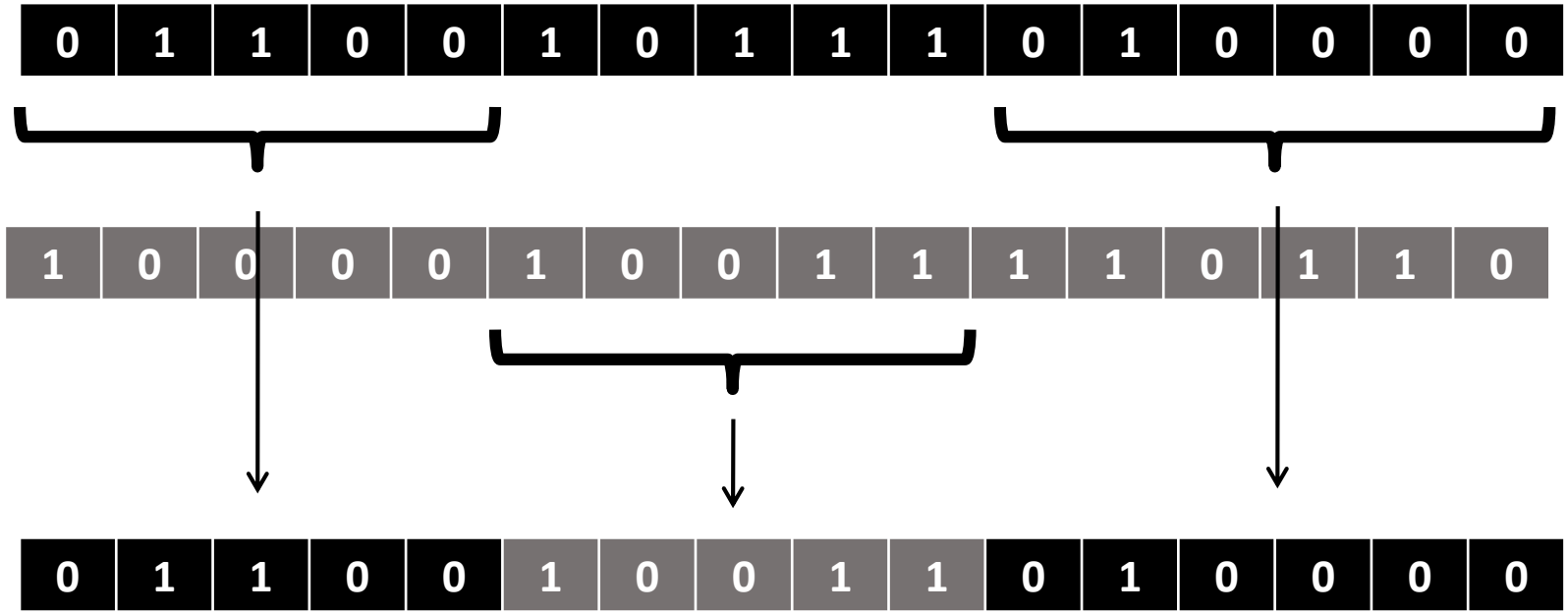
- One position for each edge
- 0 edge is not in solution
- 1 the edge is in the solution
- In other words, a coincidence matrix expanded into a vector

- Not every binary string encodes a feasible solution – edges may not form a Hamiltonian cycle
- Length of the sequence $O(n^2)$

Single-point crossover

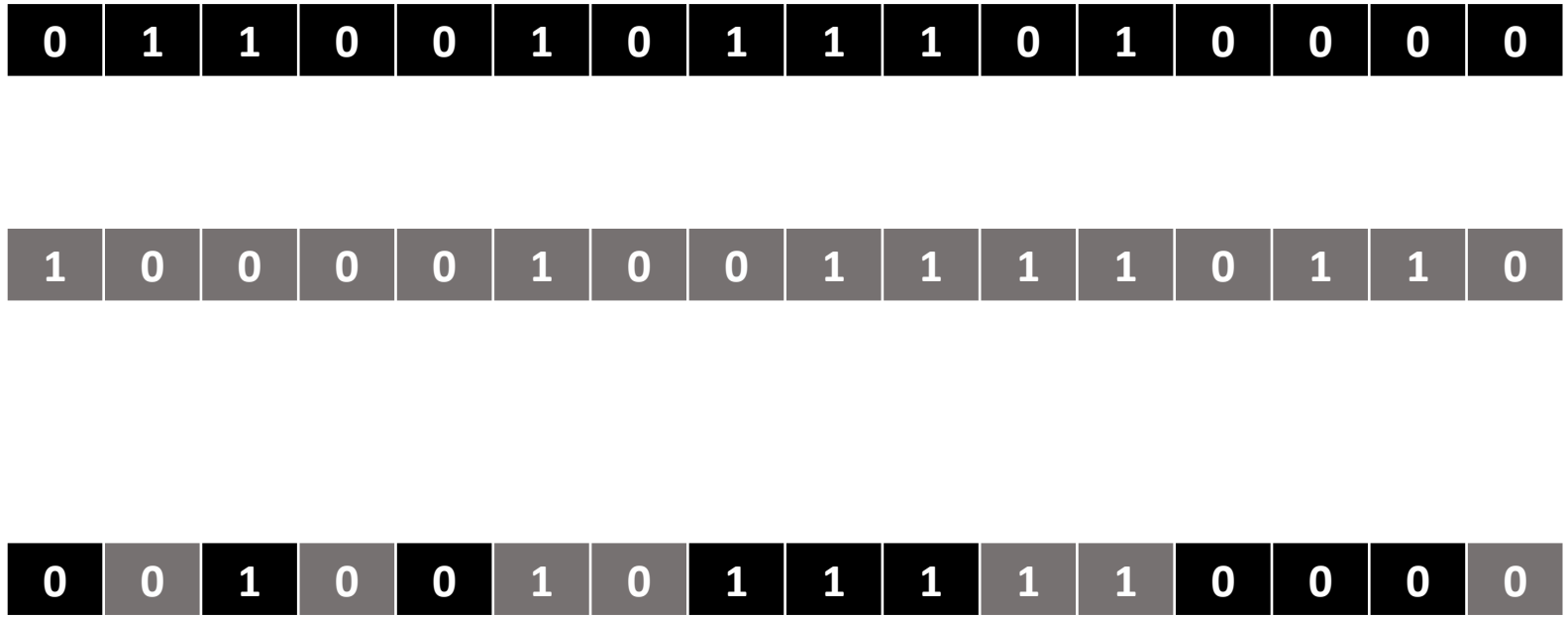


Multi-point crossover



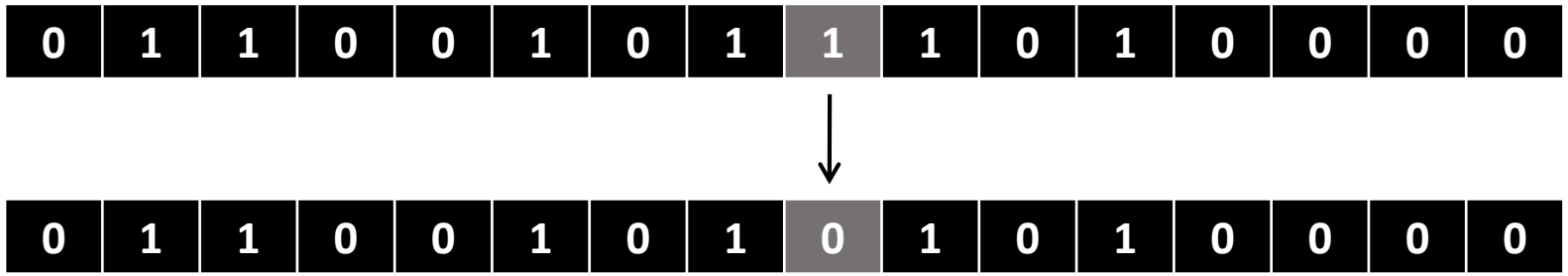
Uniform crossover

- Each position independently drawn at random from one of the parents



Mutation – random modification of a solution

- Single bit mutation



- Mutation probability – probability of changing one bit
  - For example, a probability of 0.01 means an average change of one bit per 100

Fitness

- Scaled (in general) objective function $\omega(\mathbf{x})$
- Usually maximized
- Often scaled to range [0, Max]

Roulette-wheel selection

- Calculate total fitness
  - $F_{tot} = \sum_{i=1}^{pop\_size} \omega(\mathbf{x})$
- Calculate the selection probability for each solution
  - $p_i = \frac{\omega(\mathbf{x})}{F_{tot}}$

Roulette-wheel selection algorithm

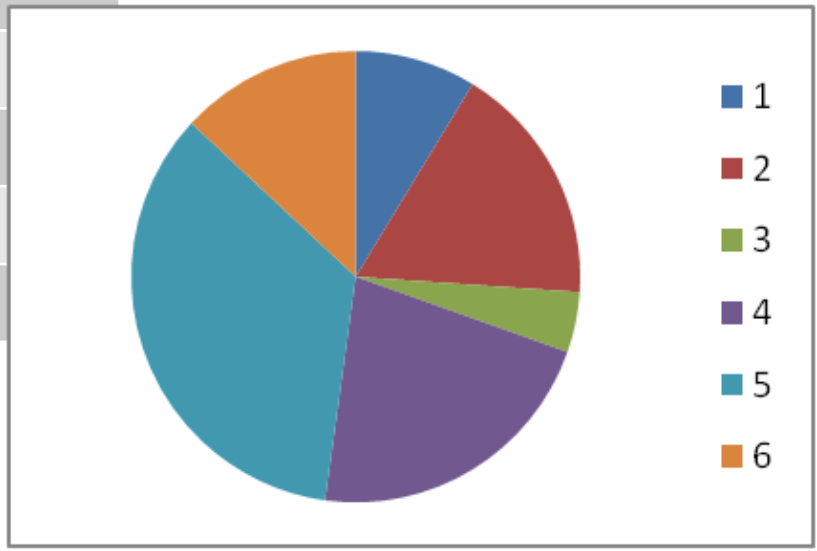
repeat *pop_size* times

generate a random number *r* from the interval [0,1) with uniform distribution

choose a solution *j* for which $\sum_{i=1}^{j-1} p_i \leq r \leq \sum_{i=1}^{j} p_i$ (repetitions are allowed)

Roulette-wheel selection example

Solution	Fitness	P_i
1	2	0,087
2	4	0,174
3	1	0,043
4	5	0,217
5	8	0,348
6	3	0,130
Total fitness	23	



Tournament selection - algorithm

repeat *pop_size* times

choose randomly *K* solutions for the tournament – uniform distribution

choose the best solution from the tournament group (repetitions are allowed)

- Often *K*=2
- The larger *K*, the greater the selection pressure (pressure to choose better solutions)

Elite selection

- Selection of *pop_size* best solution
- Possibly without repeating solutions – copies, clones
- Possible hybridization with other mechanisms (roulette, tournament) – partial elitism
  - Selection $L < P$ best solutions. Other solutions selected in other ways
- Often leads to (too) fast convergence

Premature convergence and genetic drift

- Premature convergence – convergence to solutions other than the global optimum
- Genetic drift
  - Iteratively repeated crossover/recombination leads to convergence of solutions in the population combined with random walk (even without selection)
  - One of the key driving forces of evolution

Preventing premature convergence – increasing population diversity

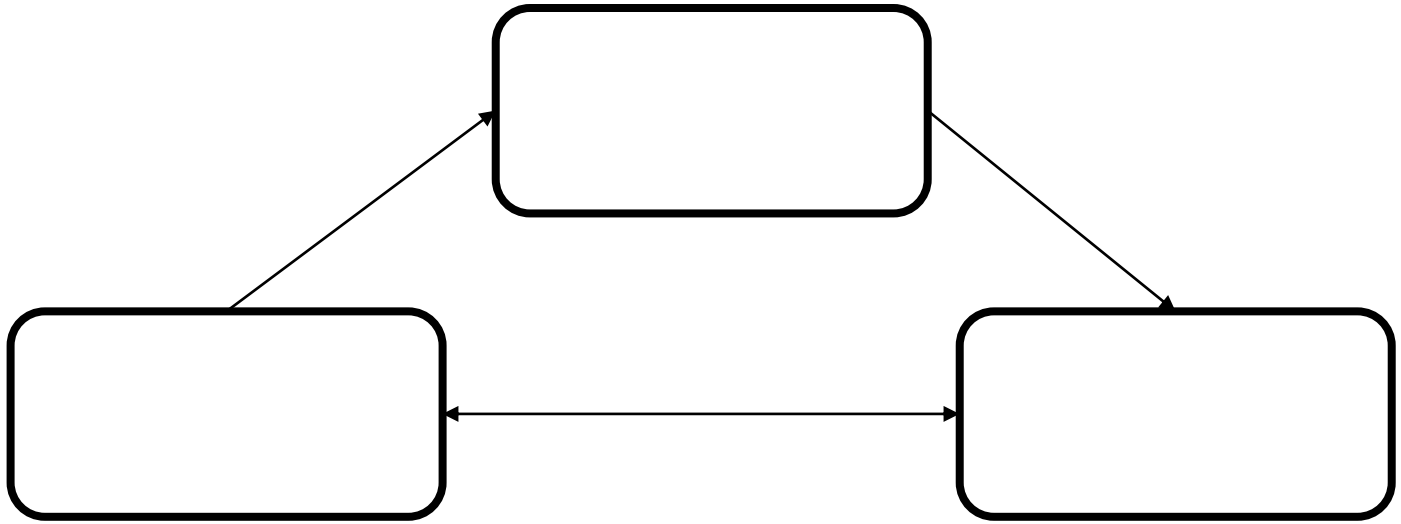
- Increasing population size
  - Increases calculation time
  - Rapid saturation – above a certain population size, there are no noticeable effects with further increase
- Mechanisms for increasing diversity - modification of selection mechanisms
- Island population models

Mechanisms for increasing diversity

- Eliminating (not accepting) copies/clones
  - The same solutions or solutions with the same values of the objective function
- Crowding – the offspring competes with its parents
- Restricted Tournament Selection – the offspring competes with the closest (in the sense of some measure of distance, e.g. Hamming distance) solution in the population
- Fitness sharing – similar solutions share "resources" and their fitness is reduced
- Clearing – the offspring competes with all solutions within a certain radius, only one winner survives
  - Radius setting problem

Island population models

- The population is divided into several disjoint populations evolving (almost) independently
- From time to time, certain solutions (e.g. the best of each population) migrate to another population (rather copying) – to another island



The concept of a schema

Common schema

String of length *n* matches 2^n schema

Order of a schema

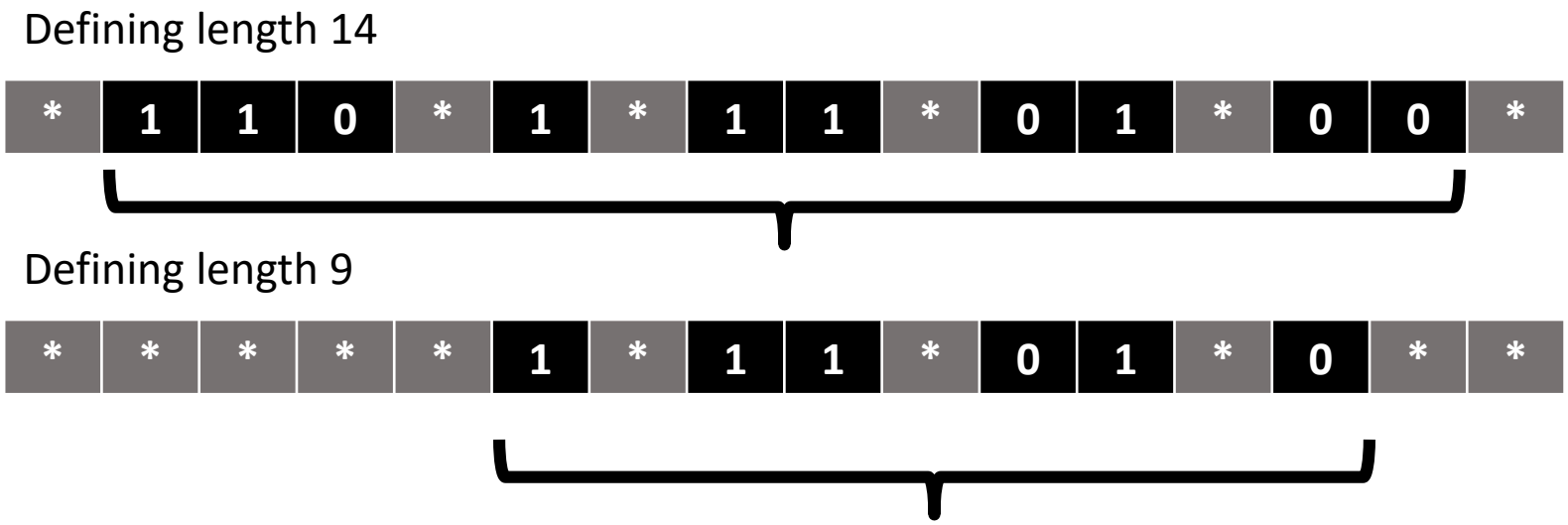
- Number of fixed positions - 0 or 1

Order 12

Order 8

Defining length of a schema

- The distance between the first and last fixed positions



Evolution of schema

- Evolutionary algorithm:
  - Evolution of solutions
  - Also the evolution of schema, of which there are much more than solutions
- Selection
  - For roulette-wheel selection, the average number of solutions matching schema *S* after the selection changes is proportional to the ratio of schema fitness (average fitness of matching solutions) to average population fitness
- Crossover
  - Shorter patterns have a better chance of "survival". Multi-point crossover, and especially uniform crossover, destroys schema to a greater extent
- Mutation
  - Schema of a smaller order have a better chance of "survival"
- An evolutionary algorithm makes sense if there are good and bad schema

Holland schema theorem

Short, low-order and well-adapted schema spread through subsequent generations according to the exponential law of growth

Evolutionary algorithms – natural encoding

- More letters of the alphabet
- Natural and floating point numbers
- Lists, sequences, permutations
- Matrices
- Trees
- Sets
- … any data structures

Advantages of natural coding

- Easier crossover – recombination
  - Creation of a new (or several) solution (offspring) on the basis of two parents combining features of parents
- More "sensible" recombination – preserving important features of parent solutions
- Easier (often guaranteed) feasibility

Order crossover (OX) for lists

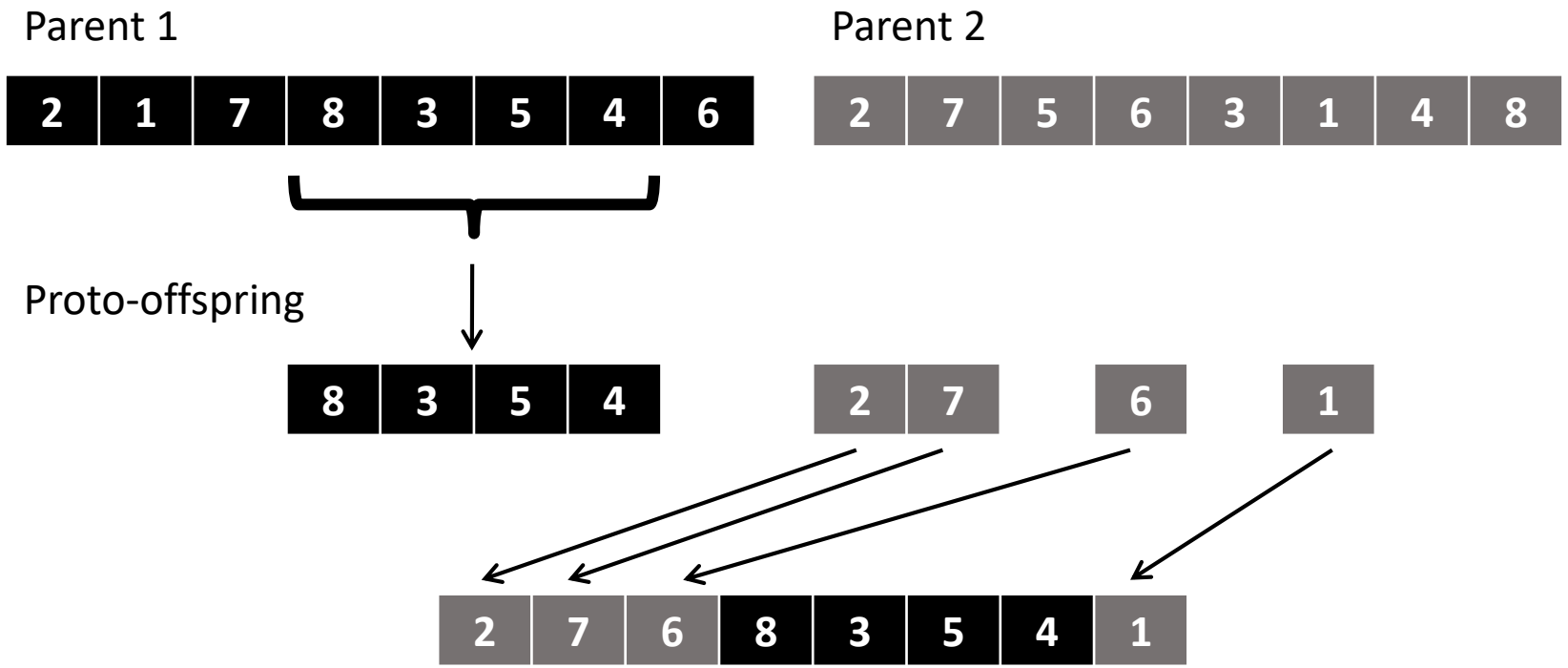
Choose a substring from parent 1

Copy the selected substring to the offspring

Remove elements already placed in the offspring from the parent 2

Add remaining elements from parent 2 to the offspring in the order of their occurrence in parent 2

Order crossover (OX) for lists

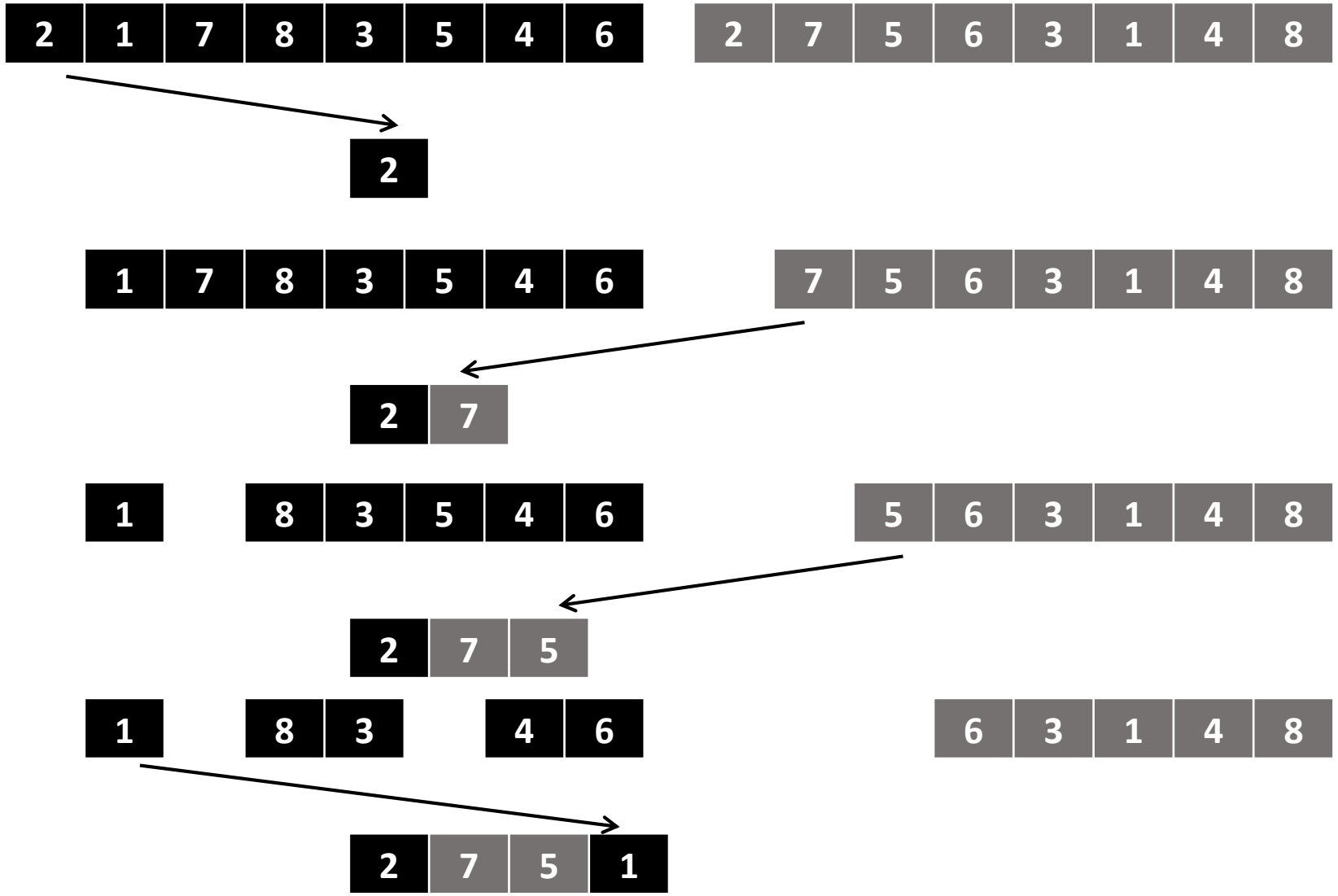


Another example of recombination for lists

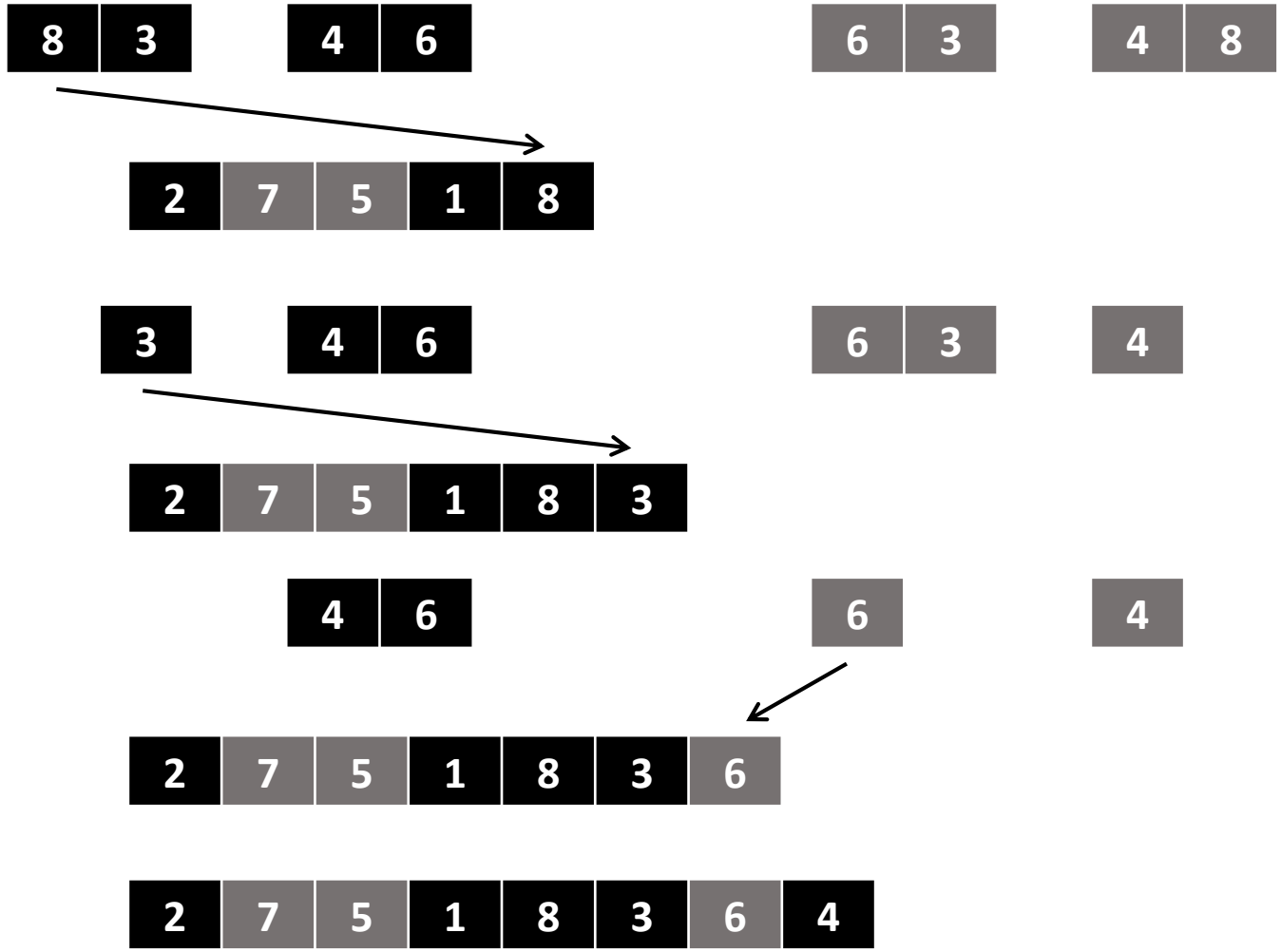
repeat for each position

- draw at random parent 1 or 2
- choose the first element from the selected parent
- delete the selected element from both parents
- add the selected element at the end of the offspring

Example


2 1 7 8 3 5 4 6
2 7 5 6 3 1 4 8
2
1 7 8 3 5 4 6
7 5 6 3 1 4 8
2 7
1 8 3 5 4 6
5 6 3 1 4 8
2 7 5
1 8 3 4 6
6 3 1 4 8
2 7 5 1

Example continued



Other recombination operators for lists

- Partially mapped crossover (PMX)
  - As in OX start by copying a substring from Parent 1, then insert elements from Parent 2 trying to keep their absolute positions (first insert those that can be inserted into free positions)
- Edge recombination crossover (ERX)
  - Select a random element. The next element is subsequent element from one (randomly selected) parent. If both subsequent elements are already selected, the next element is selected randomly
- Cycle crossover (CX)
  - Choose the first element from Parent 1 and insert it into the position that it has in Parent 2. Then select the element from Parent 2, which was in the position of the last element selected from Parent 1, etc.

The recombination operator may be quite a complex algorithm – e.g. route-based recombination for VRP

Offspring solution *O* ← empty solution

Select at random parent *p*

repeat

Choose from *p* the route *r* with the largest number of common, not yet allocated vertices

Remove from *r* vertices that have already been added to *O*

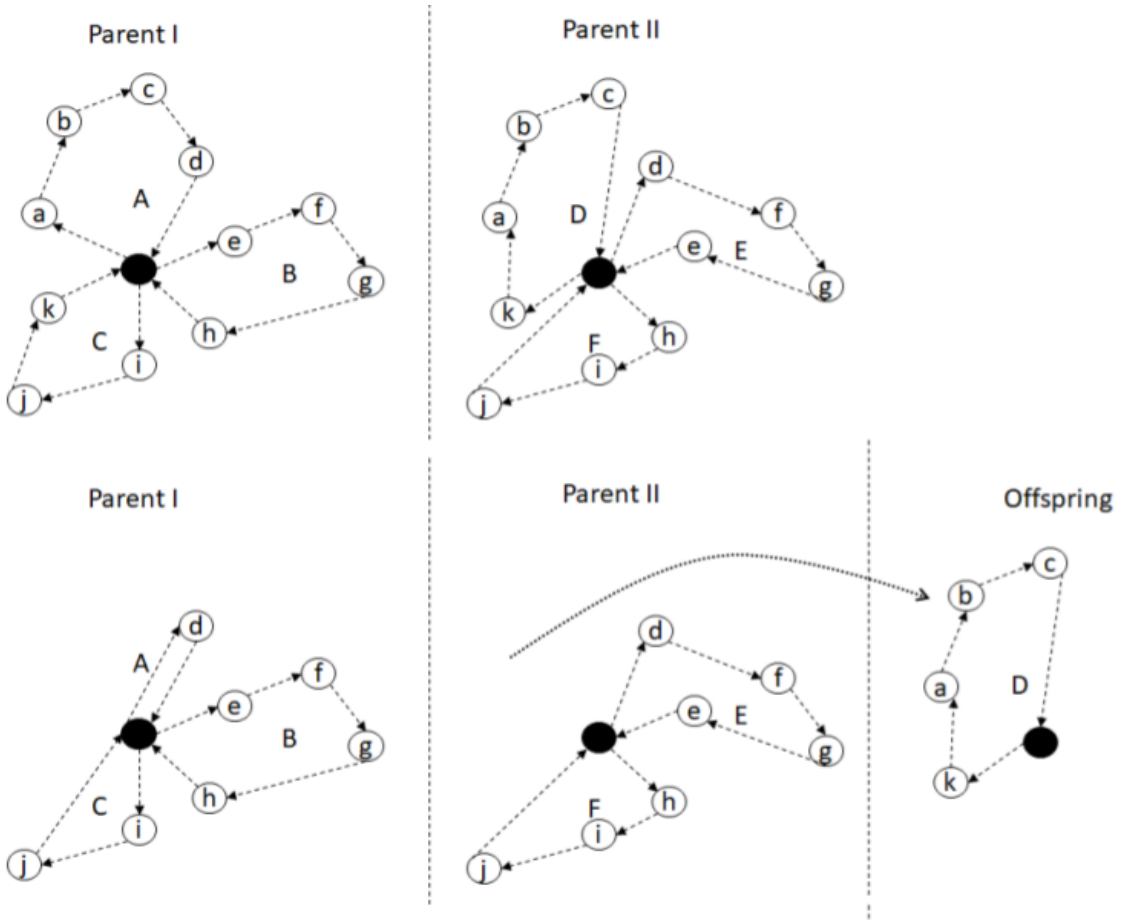
Add route *r* to *O*

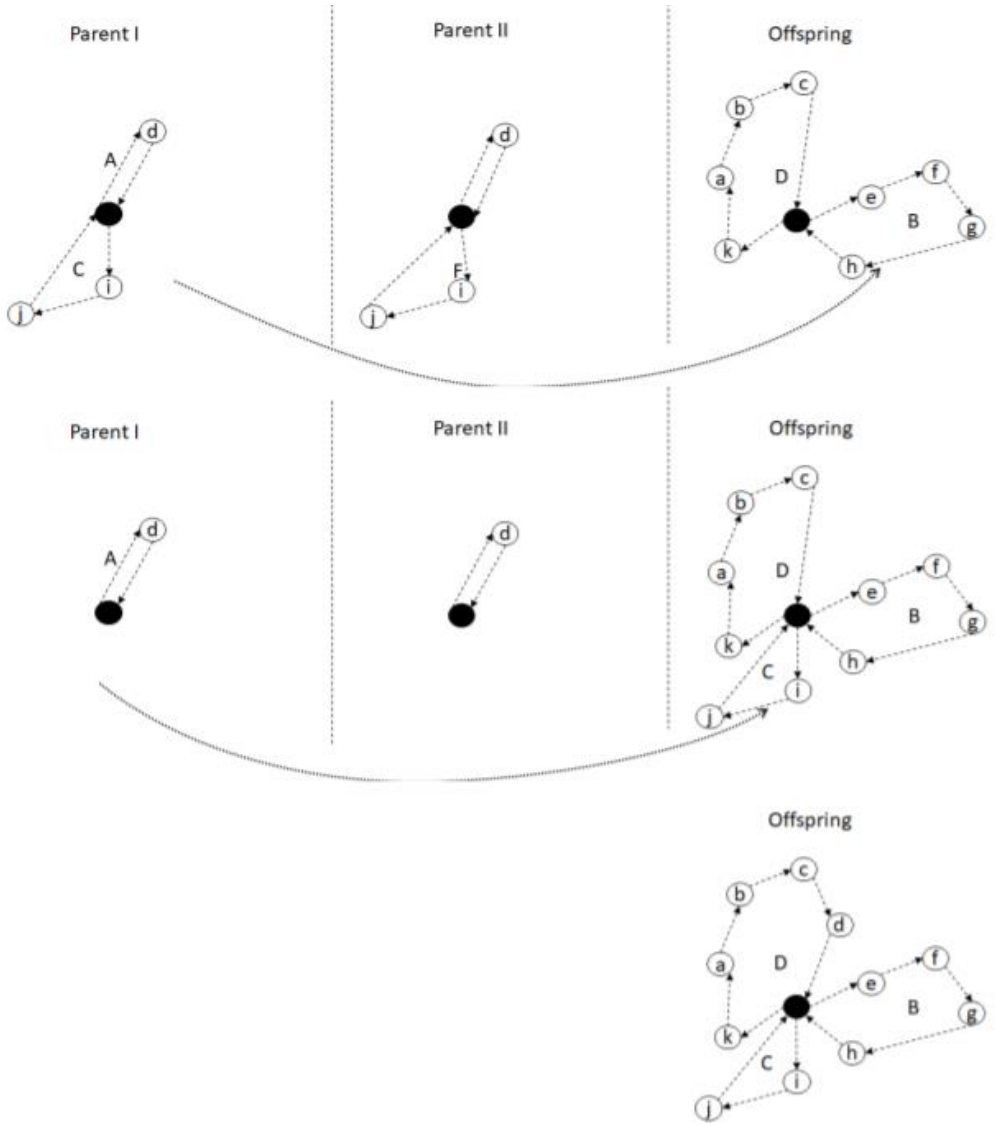
until the number of routes in the O child is equal to the number of routes in *p*

Lock all common arcs and edges in the offspring

Insert unallocated vertices using a greedy procedure and return *O*

Route-based recombination for VRP





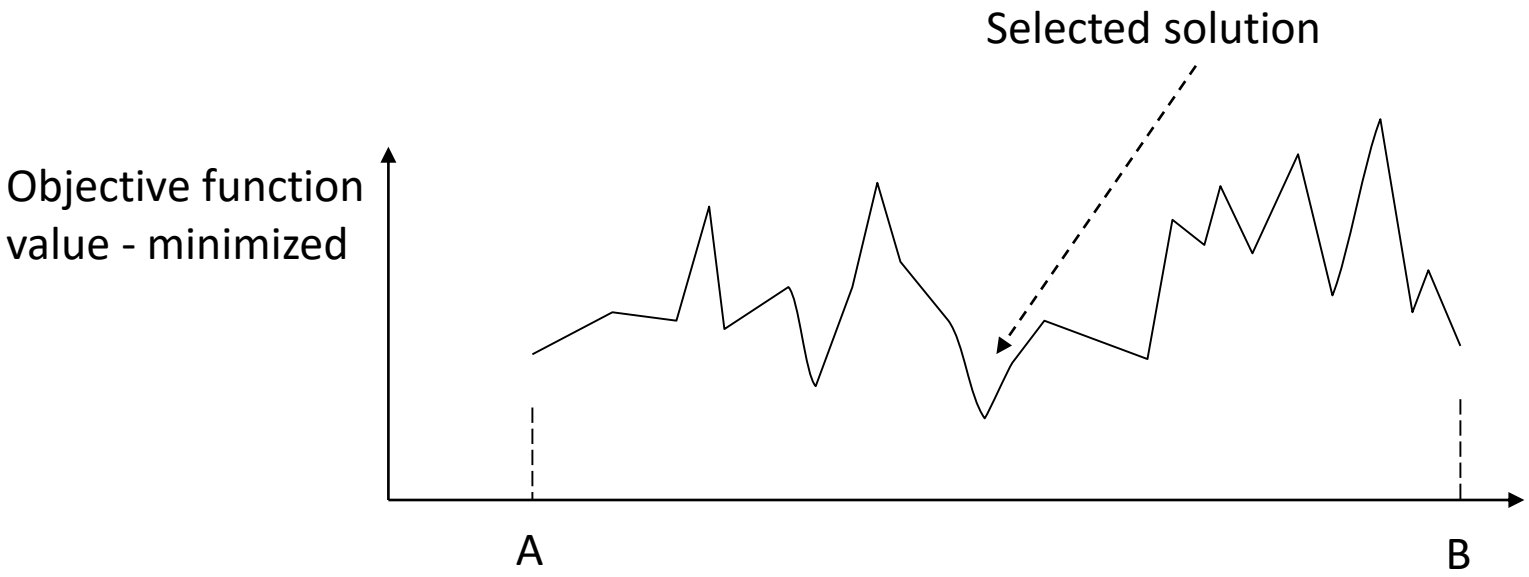
Another example for VRP - Modified Selective Route Exchange Crossover

- Treat the routes of both parents as sets of vertices
- Define the problem of maximum coverage – selection of a set of routes (of size such as in one of the parents) with a lexicographic criterion:
  - Maximization of covered vertices, then minimization the sum of the lengths of the selected routes
- Solve the problem of maximum coverage with a simple hybrid evolutionary algorithm with a local and global list of tabu (prohibited) solutions (HAE inside recombination)
- For redundantly allocated vertices (to two routes), randomly select one of them (i.e. remove from the other)
- Insert unallocated vertices using a greedy procedure

Path relinking

- Type of recombination
- Go from parent A to B by making simple moves
  - Could be a local search starting from A, where the objective function is the distance to B
  - Lexicographically, the value of the original objective function can also be taken into account, if two or more moves give the same distance to B, choose the best solution in terms of the original objective function
- Return the best solution on this path
- May lead to (too) rapid convergence. It is worth using additional mechanisms of population diversification

Path relinking



Encoding vs recombination

- The same effect can often be achieved by using:
  - Simpler encoding and more complex recombination
    - or
  - More complex encoding and simpler recombination

Hybrid evolutionary algorithms (HAE)

- Other names
  - Memetic algorithms
  - Genetic local search
  - …
- Hybridization of evolutionary algorithms and local search
  - or more generally local heuristics
- The goal is to combine the advantages:
  - Speed and ability to locally improve solutions by local search
  - Globality of evolutionary algorithms

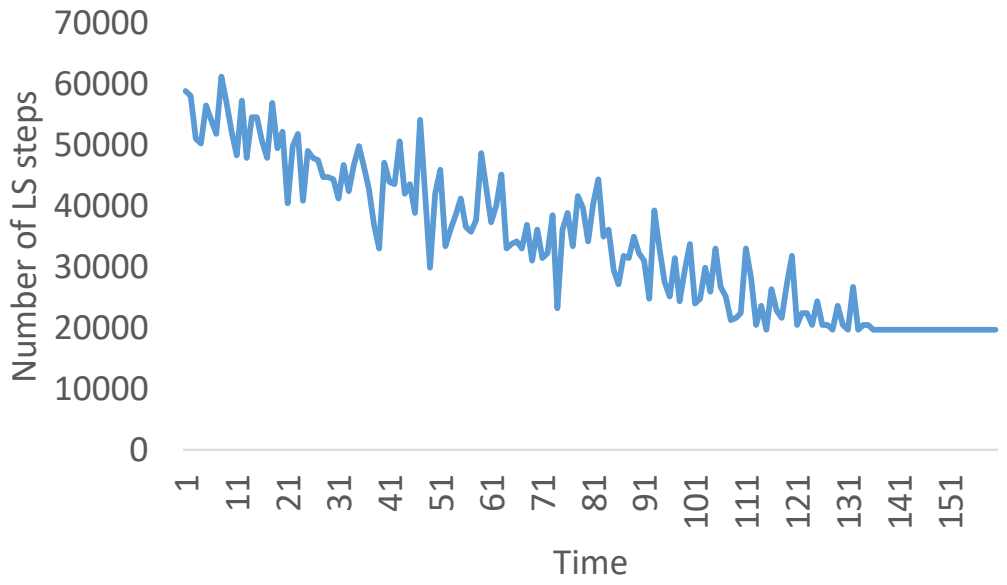
Different ways of hybridization

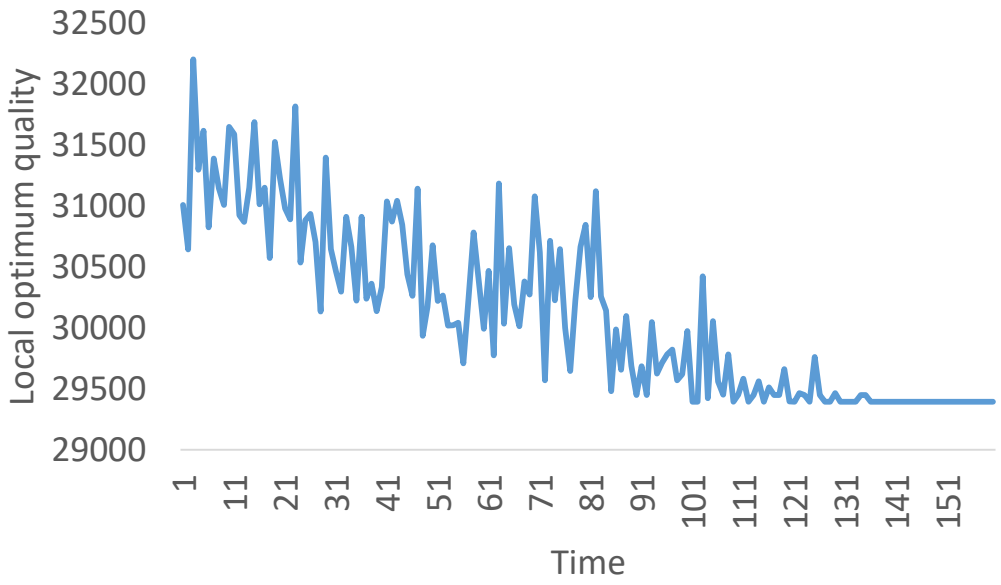
- First LS then EA
  - EA starts with a very good population of local optima
- First EA then LS
  - EA finds a globally good region explored more thoroughly by LS
- **LS during EA**
  - **E.g. LS after each (or some) recombination and mutation**

Two points of view (justification of efficiency) of HAE

- EA point of view
  - HAE is an EA working on a limited set of solutions – only a set of local optima
  - Limited search space results in more efficient operation
  - Local search ensures that EA search space is effectively limited to very good solutions
- LS point of view
  - Recombination of good solutions provides a good starting point for local search, which in turn gives:
    - Faster convergence to a local optimum
    - Convergence to better solutions

Example of local search operation during a HAE run for TSP – win-win in terms of time and quality





In HAE, elite selection is often used

- For ordinary EAs, elite selection leads to (too) rapid convergence
- Adding local search often eliminates this problem – LS introduces additional diversification of solutions obtained after recombination
  - Often no explicit mutation is needed, which can also be built into the randomized recombination operator
  - It is also important that due to rather long LS running time, fewer EA iterations are performed (less recombinations/mutations)

Steady State evolutionary algorithms

- No generation(s)
- An offspring may be added to the population (if it meets the appropriate conditions) immediately after construction

HAE with elite selection and steady state - an example

Generate an initial population **X**

repeat

Draw at random two different solutions (parents) using uniform distribution

Construct an offspring solution by recombining parents

y := Local search (**y**)

if y is better than the worst solution in the population and (sufficiently) different from all solutions in the population

Add **y** to the population and remove the worst solution

until the stopping conditions are met

A more general approach – a population algorithm generalizing methods such as HAE, ILS, LNS

Generate an initial population **X**

repeat

Draw at random two different solutions (parents) using uniform distribution

Construct a new solution **y** using a given operator – e.g. recombination, perturbation, destroy-repair

y := Local search (**y**) (optionally)

if y is better than the worst solution in the population and (sufficiently) different from all solutions in the population

Add **y** to the population and remove the worst solution

until the stopping conditions are met

Genetic Programming Hyper-Heuristic

- The rules/heuristics/code for creating solutions (not the solutions themselves) evolve
- These heuristics are learned on a set of learning instances, evaluated on a set of test instances, and applied to new instances

Parameters in evolutionary algorithms

- Number of generations/iterations – running time
- Population size
  - Smaller population – faster convergence
  - Larger population – better quality in long term
- Type of selection and updating of the population
  - The level of selection pressure – intensification vs diversification

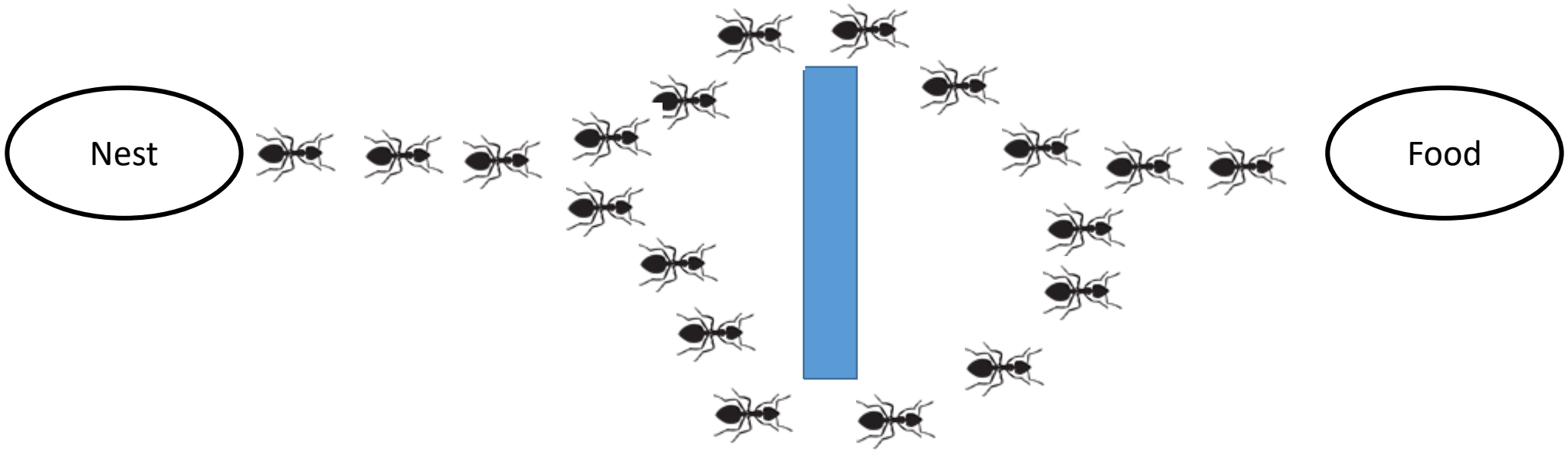
Biologically inspired algorithms and more…

- **Evolution-based**
  - Differential evolution, Evolution strategies, Genetic/evolutinary algorithms, Genetic programming…
- **Swarm-based**
  - Ant colony optimization, Artificial Bee Colony, Bat Algorithm, Crow search algorithm, Cuckoo search, Firefly Algorithm, Flower Pollination Algorithm, Grey Wolf Optimizer, Krill Herd Algorithm, Moth-Flame Optimization Algorithm, Particle Swarm Optimization, Social Spider Optimization, Whale Optimization Algorithm…
- **Physics-based**
  - Electromagnetism-like mechanism, Gravitational Search Algorithm, Simulated annealing, Sine Cosine Algorithm, States of matter search…
- **Human-based**
  - Fireworks Algorithm, Harmony Search, Imperialist Competitive Algorithm, Tabu search…

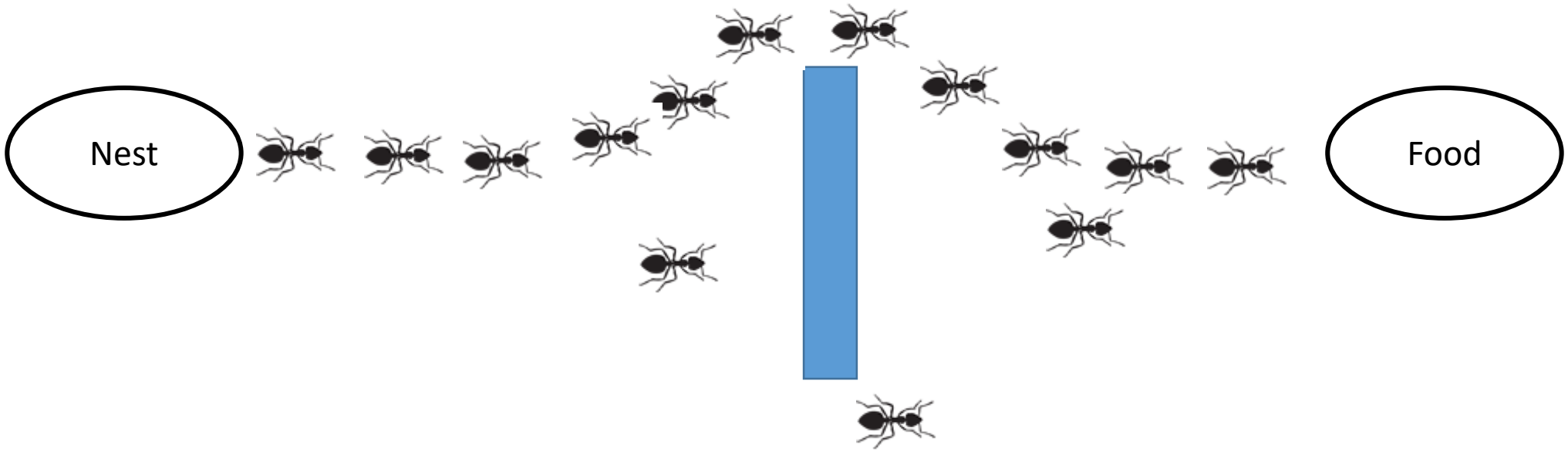
The ZOO of (nature-inspired) metaheuristics

- Fausto, Fernando; Reyna-Orta, Adolfo; Cuevas, Erik; et al., From ants to whales: metaheuristics for all tastes, ARTIFICIAL INTELLIGENCE REVIEW Volume: 53 Issue: 1 Pages: 753-810 Published: JAN 2020
- Fathollahi-Fard, Amir Mohammad; Hajiaghaei-Keshteli, Mostafa; Tavakkoli-Moghaddam, Reza, Red deer algorithm (RDA): a new nature-inspired meta-heuristic, SOFT COMPUTING Volume: 24 Issue: 19 Pages: 14637-14665 Published: OCT 2020
- Torabi, Shadi; Safi-Esfahani, Faramarz, Improved Raven Roosting Optimization algorithm (IRRO), SWARM AND EVOLUTIONARY COMPUTATION Volume: 40 Pages: 144-154 Published: JUN 2018

Ant colony algorithms



After some time, most ants choose the shortest route



Pheromone trails

- Ants moving leave pheromone trails perceptible by other ants
- Pheromones gradually evaporate - the trails disappear
- Ants leave a stronger trail on the paths leading to food
- On shorter paths, evaporation is slower
- Strong pheromone trail attracts other ants

The idea of an ant colony algorithm for TSP

- Artificial ant – an agent who moves from vertex to vertex
- Ants prefer vertices connected by edges with a large amount of pheromone
- Ants start from randomly selected cities
- They move to new verticies, modifying the pheromone trails on the traversed edges (local trail updating)
- After completing all routes, the ant whose route was the shortest modifies the edges belonging to its route by adding the amount of pheromone inversely proportional to the length of the route (global trail updating)
- Pheromone trail gradually disappears

Hybridization of ant colony algorithms with local search

- Hybrid methods give the best results
- After a ant builds a solution, local search is triggered

Constraints in evolutionary/metaheuristic algorithms

- Most EA/MH are defined as methods for problems without constraints
- In practice, operators of e.g. neighborhood, crossing, mutation, may lead to infeasible solutions
- E.g. for a knapsack problem – exceeding the capacity of the knapsack
- For example, for TSP – a set of edges that does not form a Hamiltonian cycle

Possible ways of constraints handling

- Problem/operators re-formulation to an unconstrained version
- Rejection of infeasible solutions
- Repair of infeasible solutions
- Intermediate encoding and decoding
- Penalty functions
- Multiobjective approach

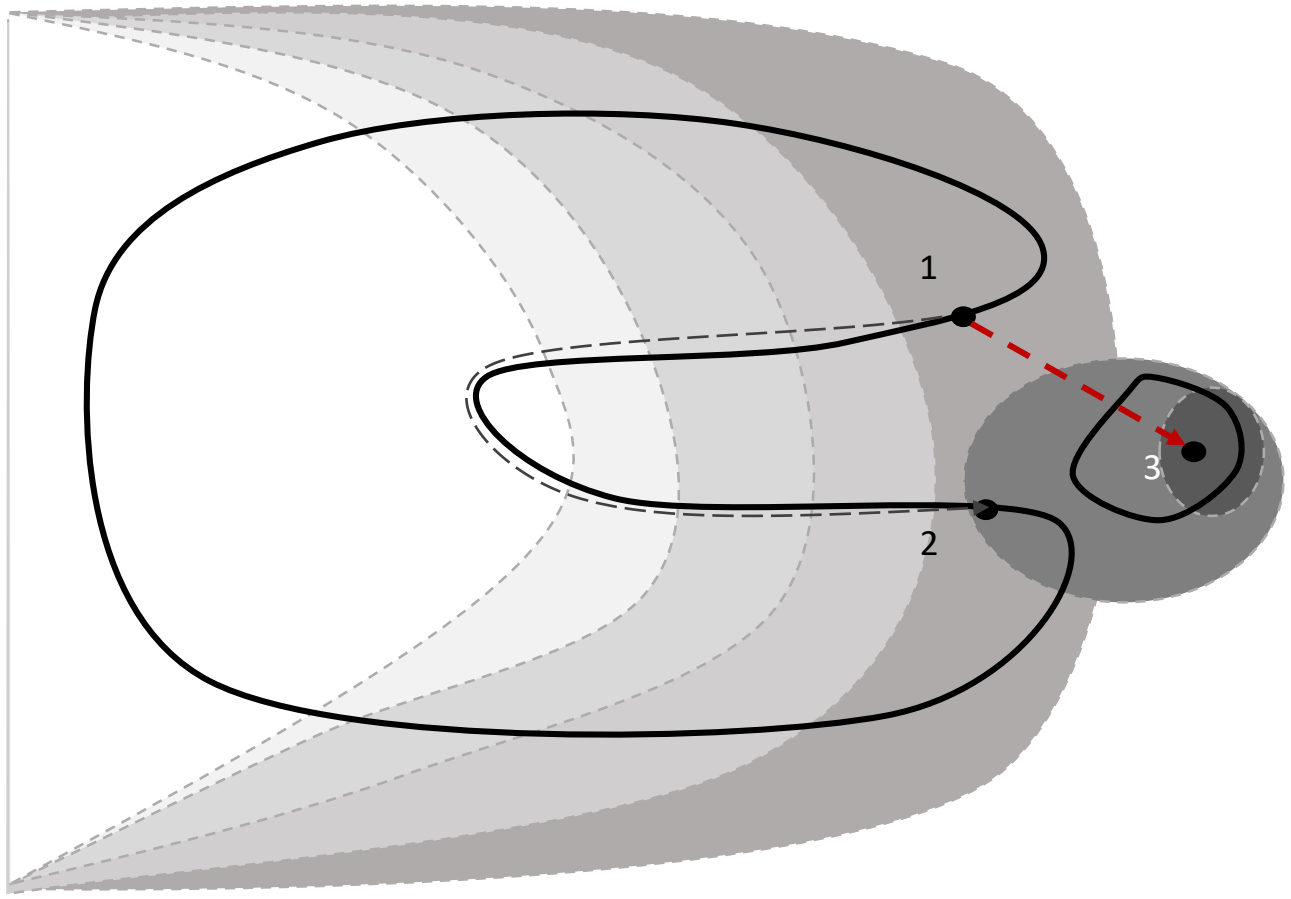
Problem/operators re-formulation

- For example, TSP has constraints if solutions are encoded as sets of arcs/edges, but if the solutions are encoded as permutations of all vertices, then any permutation defines a feasible solution
- It is easy to design operators for permutations always constructing new permutations
- Approach possible with relatively simple constraints (e.g. difficult to apply in VRP)

Rejection of infeasible solutions

- Infeasible solutions are immediately rejected
- Ineffective for highly constrained problems where it is difficult to obtain a feasible solution
- Changes the landscape of the objective function, e.g. the transition from a given feasible solution to a better feasible solution may be impossible or very difficult

Difficulties in rejection of infeasible solutions

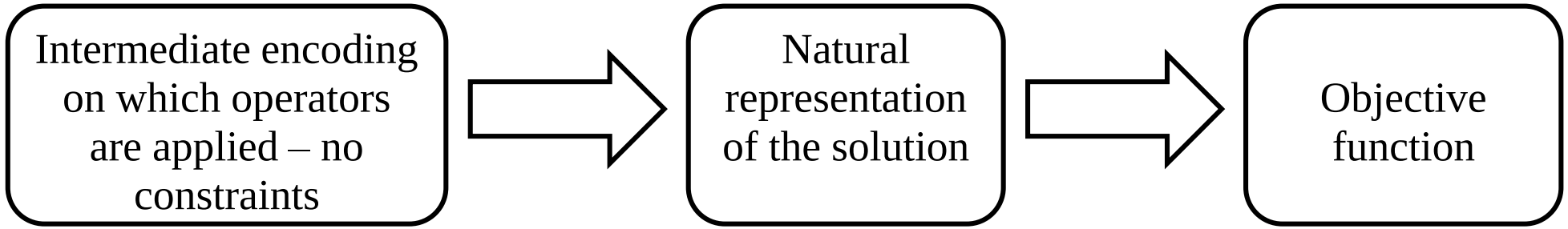


Repair procedures

- An infeasible solution resulting from the use of an operator is repaired by a dedicated heuristic
- The repair procedure should not change the solution too much
- The repair procedure may be guided by the objective function
- E.g. knapsack problem
  - Delete items until the solution is feasible
  - Option – remove items with the worst value-to-weight ratio
- E.g. TSP
  - Rebuilding the Hamilton cycle while maintaining as many edges as possible
  - Option –shortest possible edges are preferred

Intermediate encoding (representation)

Solution decoding: Heuristic
or exact transformation



Intermediate encoding for knapsack problem

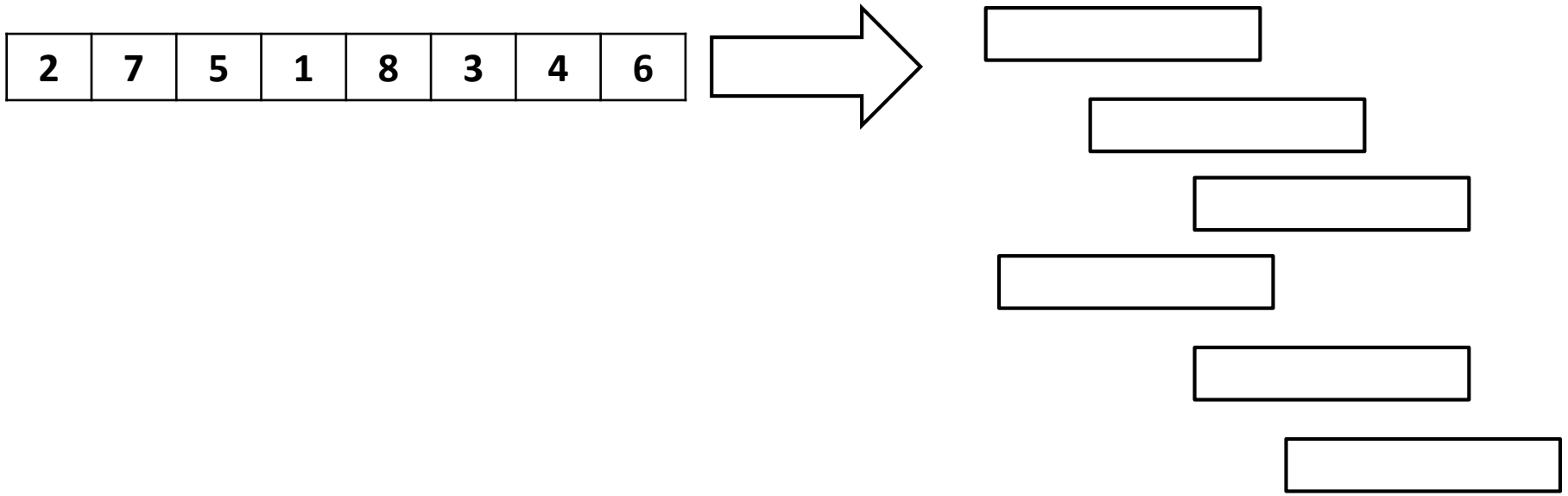
- Intermediate representation: an ordered list of items – operators that act on lists
- Decoding – selection of items from the list one by one until the knapsack is full

Intermediate encoding for task scheduling

Intermediate encoding – priority list of tasks

Priority heuristic

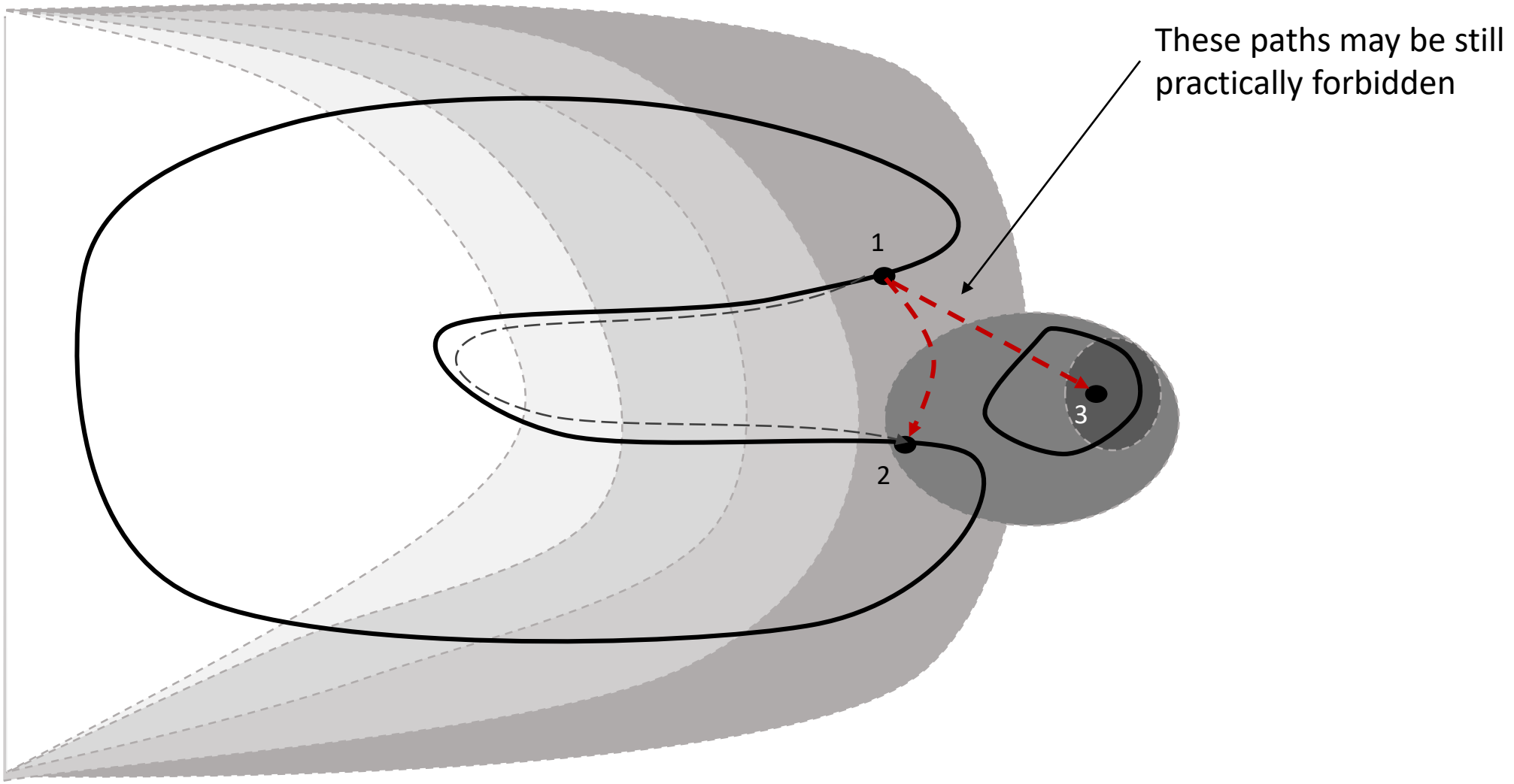
Schedule



Penalty function(s)

- For each constraint (type of constraint) a measure of violation is defined (0 if the constraint is met), e.g. by excess of the size of the knapsack - penalty
- Infeasible solutions are treated as feasible, but the value of the objective function is reduced by the value of the penalties(s)
- Difficulties
  - Too small penalty may lead evolution towards infeasible solutions
  - Too large penalty may make it difficult to reach good solutions – in fact, it changes the landscape of the objective function – solutions that strongly exceed the constrains are in practice unattainable
- Rejection may be interpreted as the use of the infinite weight of penalty

Difficulties when applying the penalty function

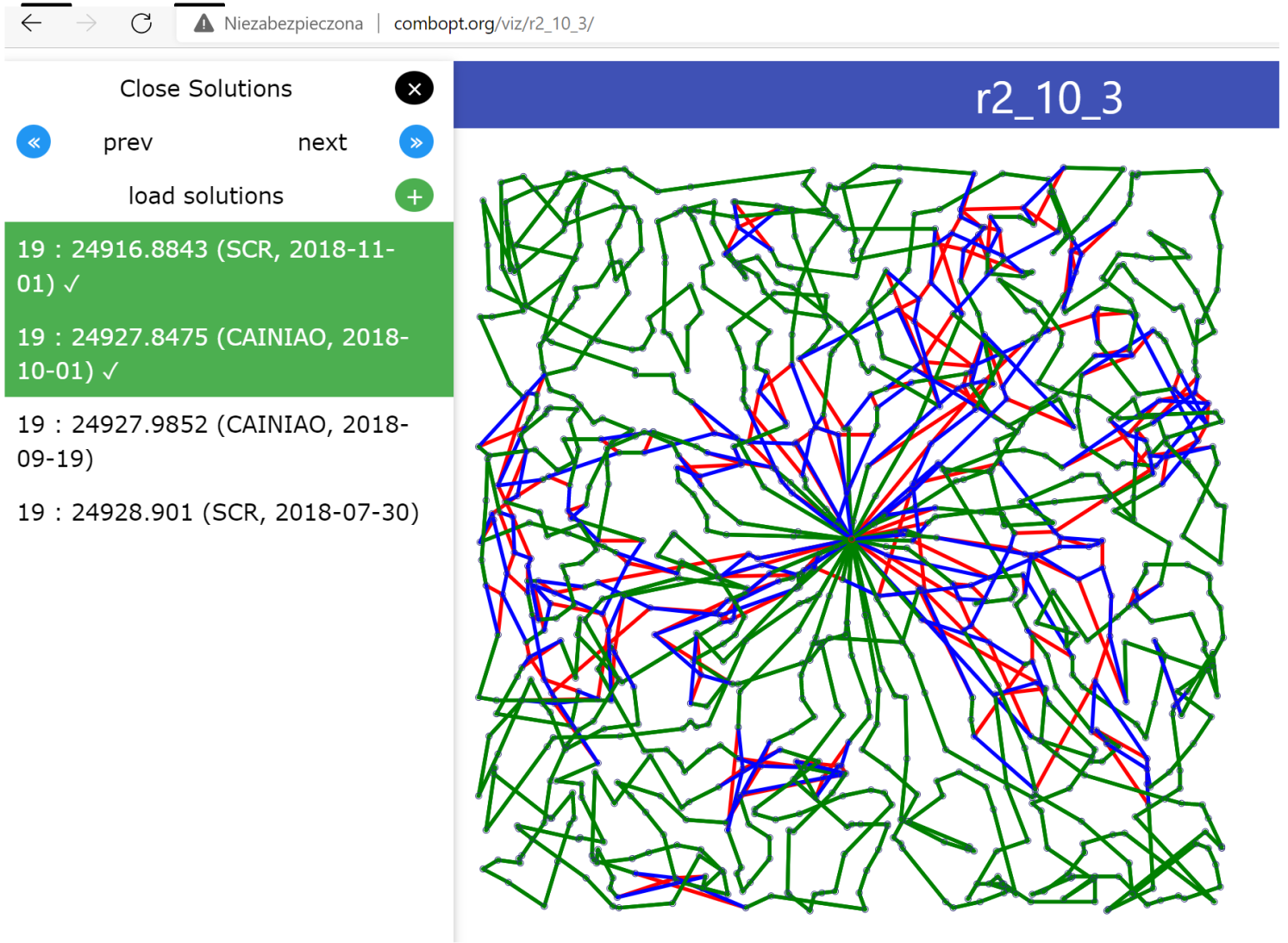


New method for setting the weights of the penalty function

- Beling, Cybula, Jaszkiewicz, Rogalski, Sielski – over 100 records for Capacitated Vehicle Routing with time windows (www.sintef.no/vrptw), as well as PDPTW, TOPTW
- The main idea: in order to improve the best feasible solution, it is worth going deeper into the area of infeasible solutions
- Improvement and feasibility restoration phases
- Improvement phase – the weights of penalties are reduced until the best solution is improved at the cost of feasibility
- Feasibility restoration phase – penalty weights are increased to achieve a feasible solution with the least possible deterioration of the value of the objective function

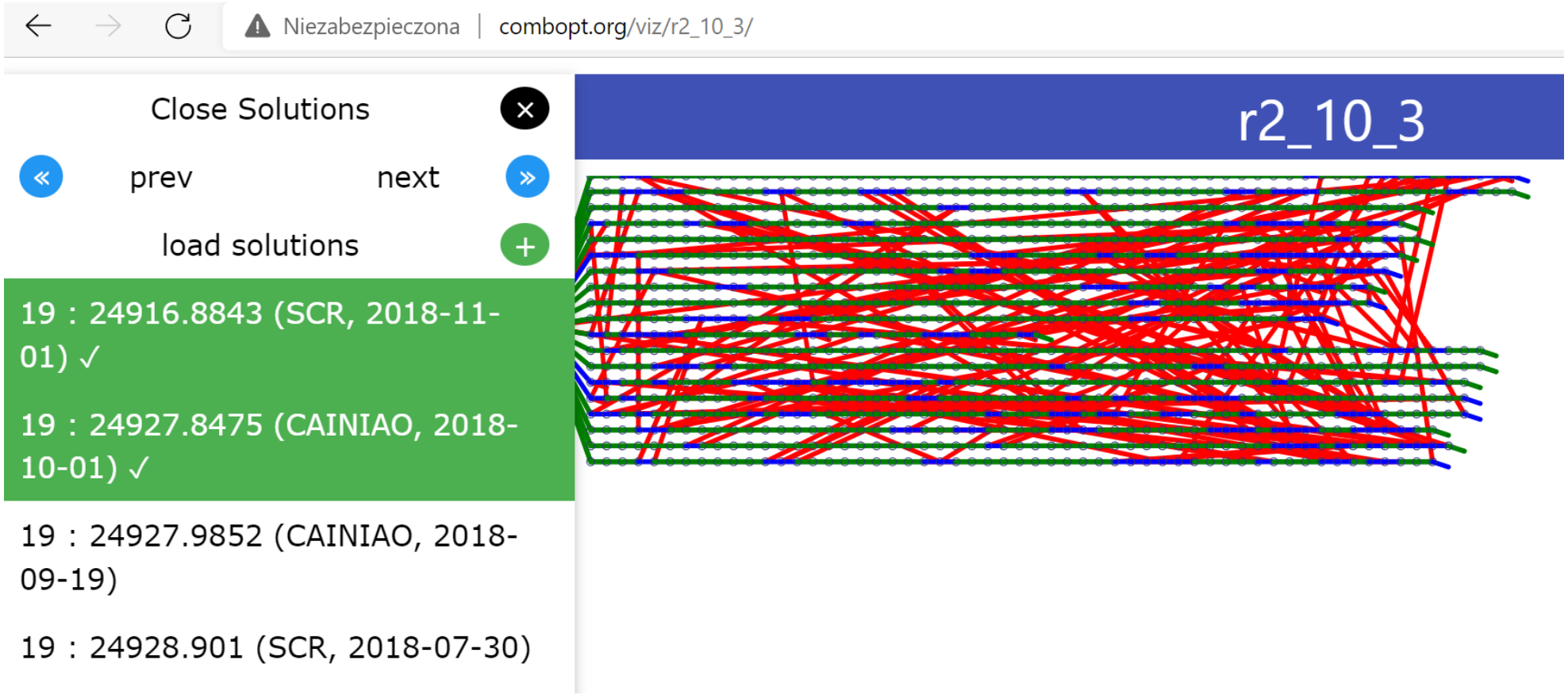
- Deep Infeasibility Exploration Method for Vehicle Routing Problems, Piotr Beling, Piotr Cybula, Andrzej Jaszkiewicz, Przemysław Pełka, Marek Rogalski, Piotr Sielski, EvoCOP 2022, s. 62-78.

Comparison of the two best known solutions for r2_10_3 VRP instances

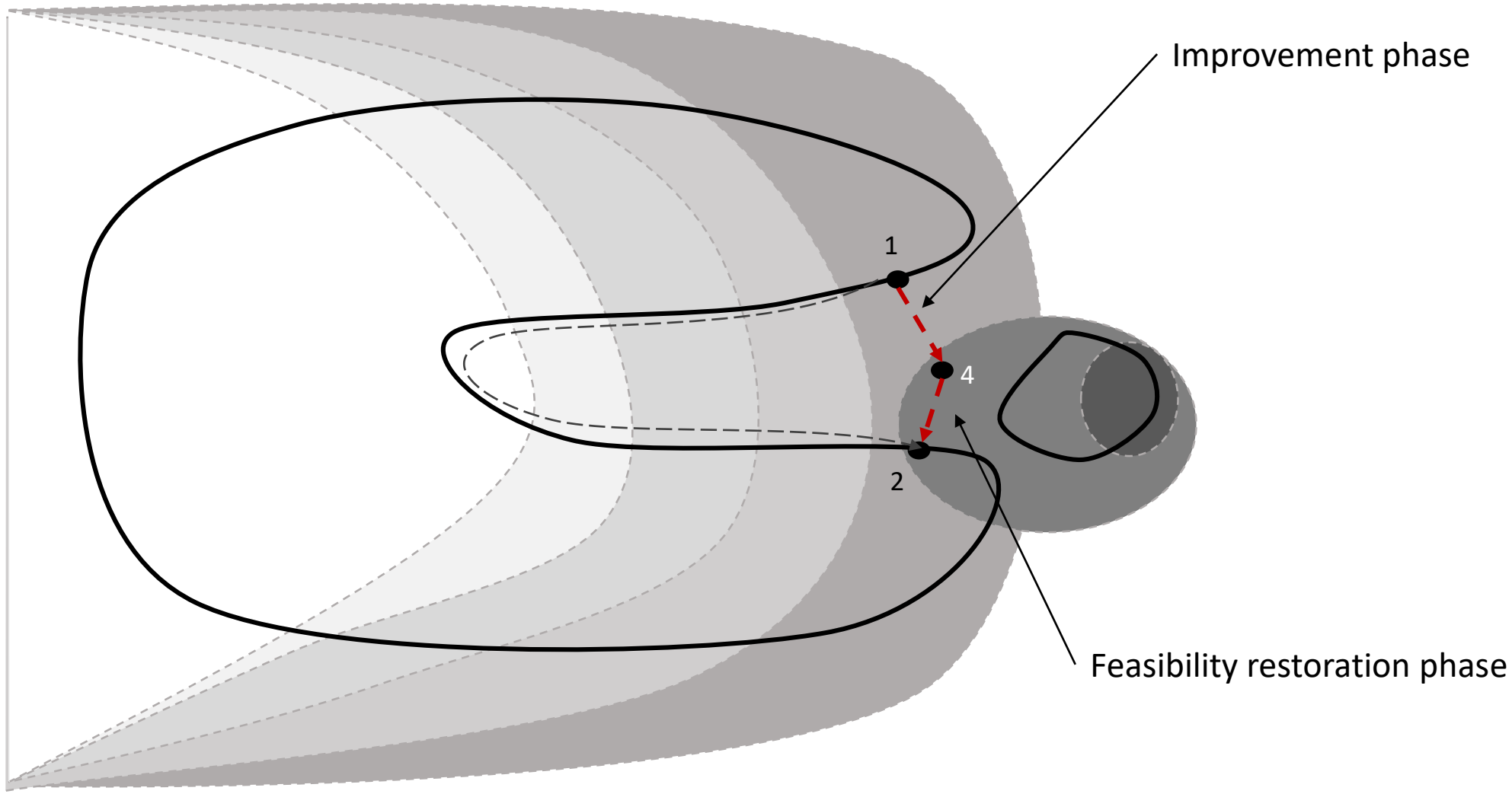


http://combopt.org

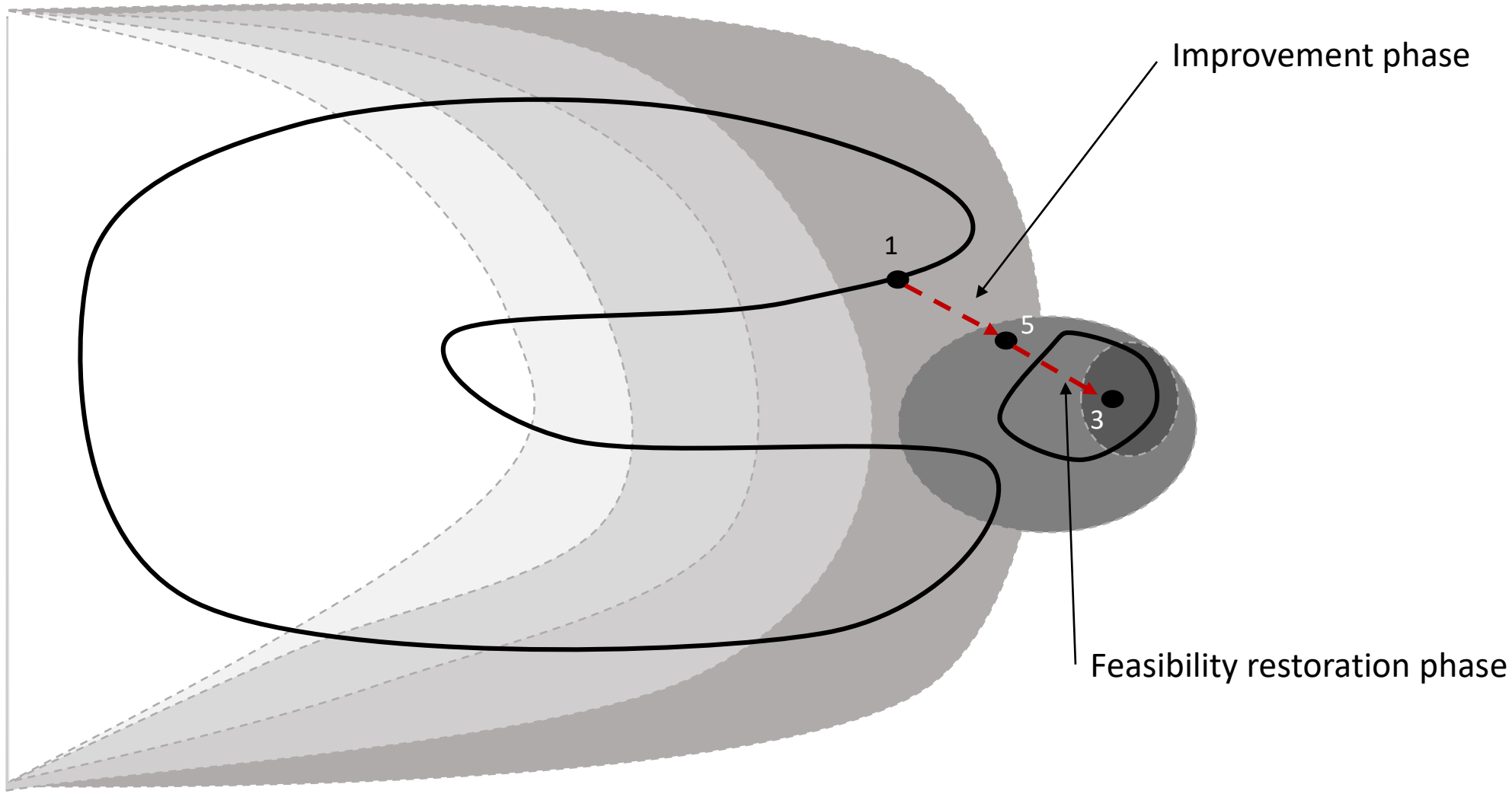
Comparison of the two best known solutions for r2_10_3 VRP instances



Two alternating phases



Two alternating phases



Multiobjective approach

- Objective function and level of constraints violation are treated as two separate optimization criteria
  - The typical penalty function can be interpreted as the application of the weighted sum of the objective function and the level of constraints violation
- Multiobjective metaheuristics are used, e.g. generating a set of Pareto-optimal solutions for these criteria