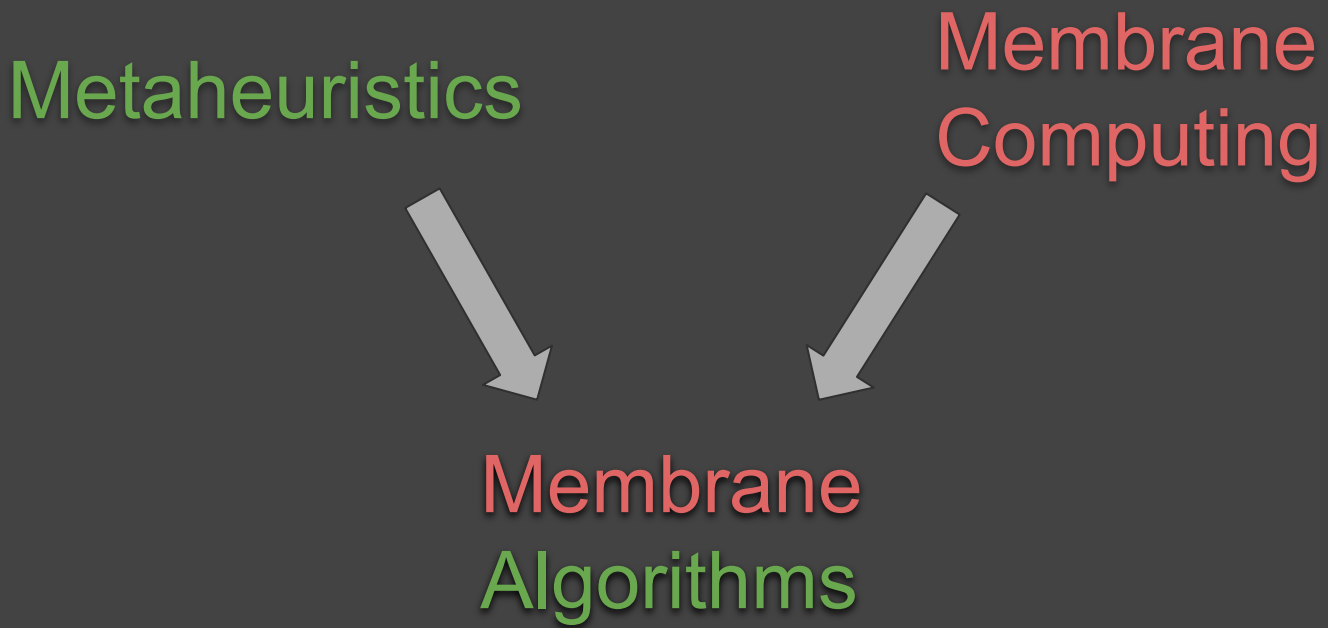
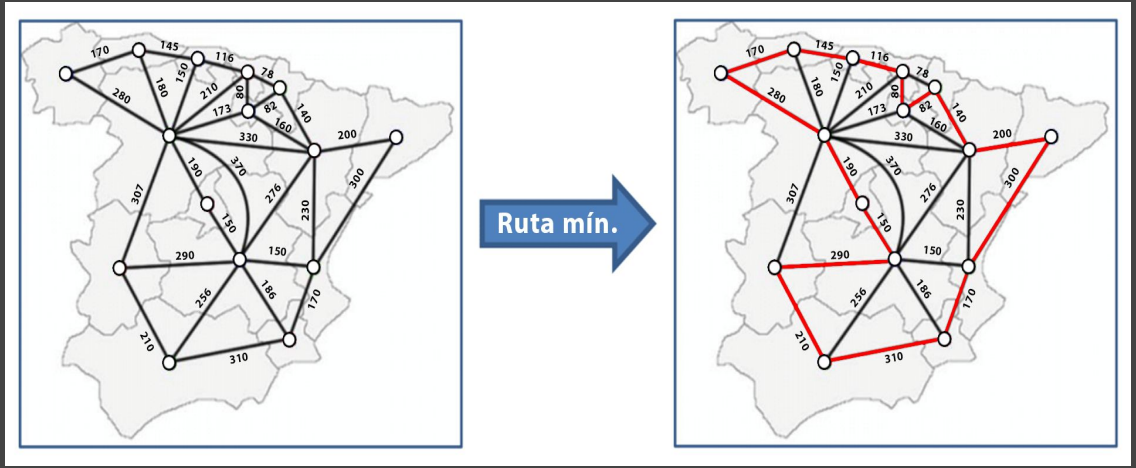
An introduction to Membrane Algorithms

José Antonio Andreu Guzmán

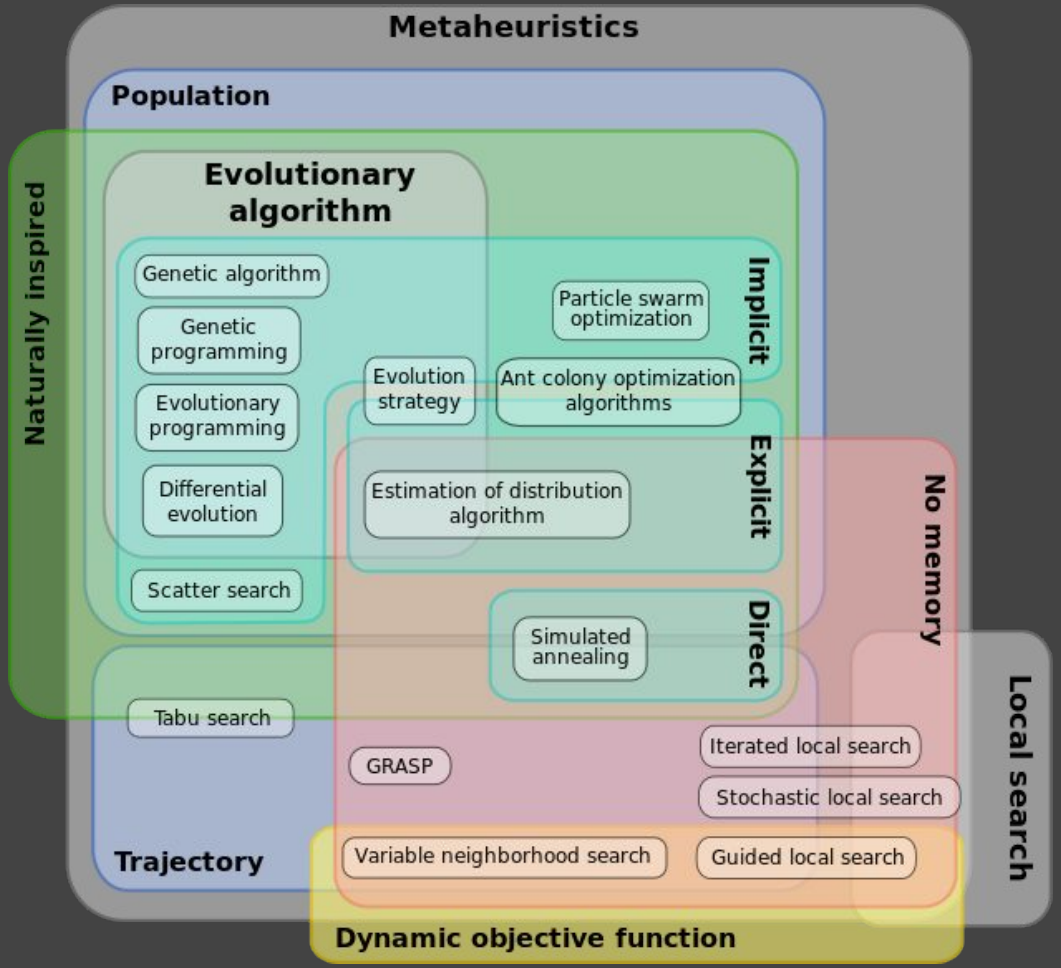
20th Brainstorming Week on Membrane Computing


Metaheuristics
Membrane Computing
Membrane Algorithms

Travelling Salesman Problem

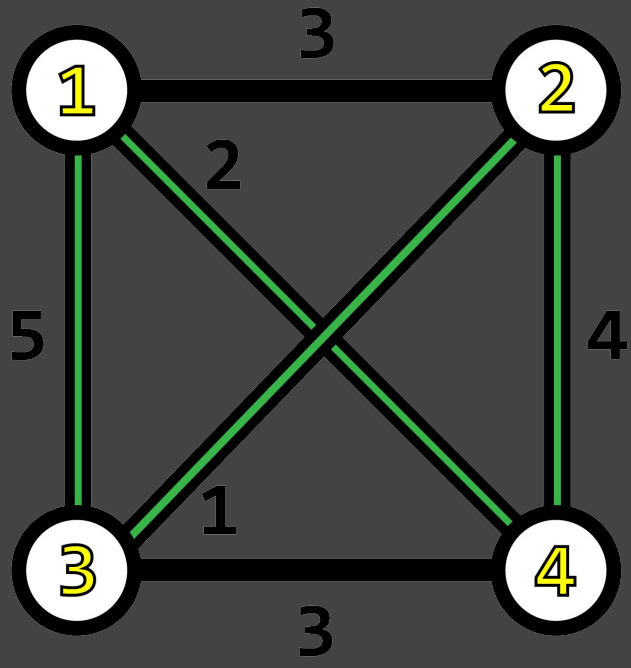


n	O(n!)
1	0.0001 seg
10	6.048 min
100	$2.95 \cdot 10^{146}$ años


Metaheuristics
Population
Evolutionary algorithm
Naturally inspired
Genetic algorithm
Genetic programming
Evolutionary programming
Differential evolution
Scatter search
Particle swarm optimization
Evolution strategy
Ant colony optimization algorithms
Estimation of distribution algorithm
Simulated annealing
Implicit
Explicit
Direct
No memory
Local search
Tabu search
GRASP
Iterated local search
Stochastic local search
Variable neighborhood search
Guided local search
Trajectory
Dynamic objective function


SOLUTIONS
OPERATORS

Chromosome

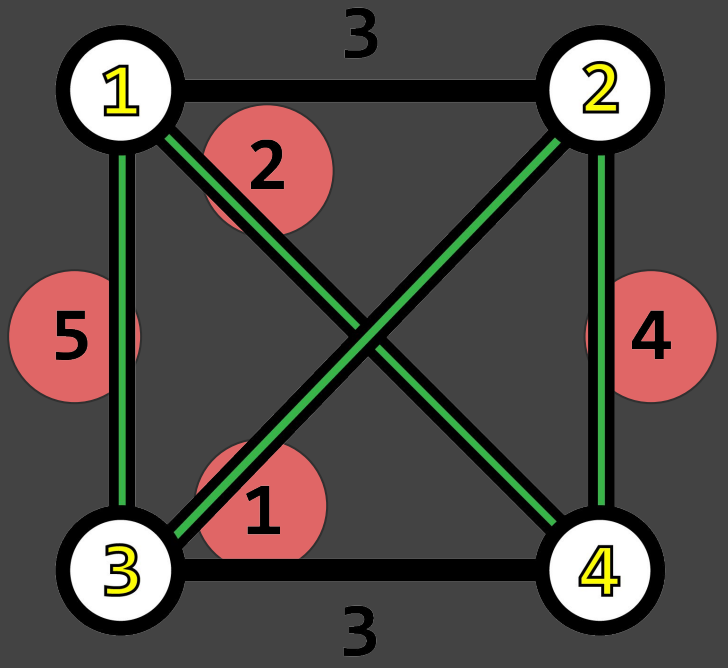


Phenotype

1	4	2	3
1	2	3	4

Genotype

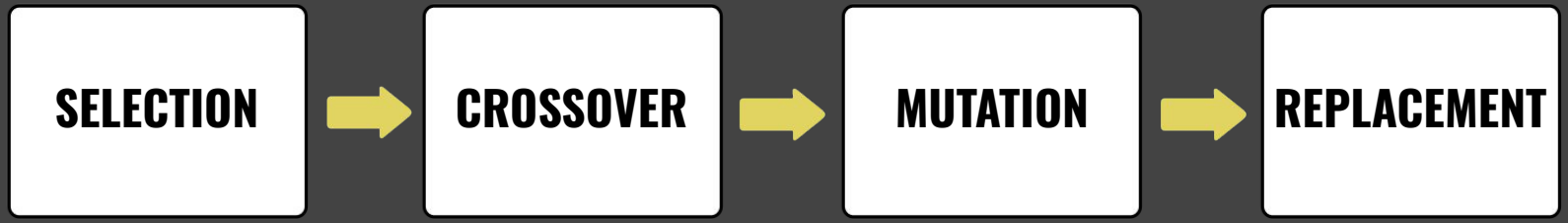
Fitness



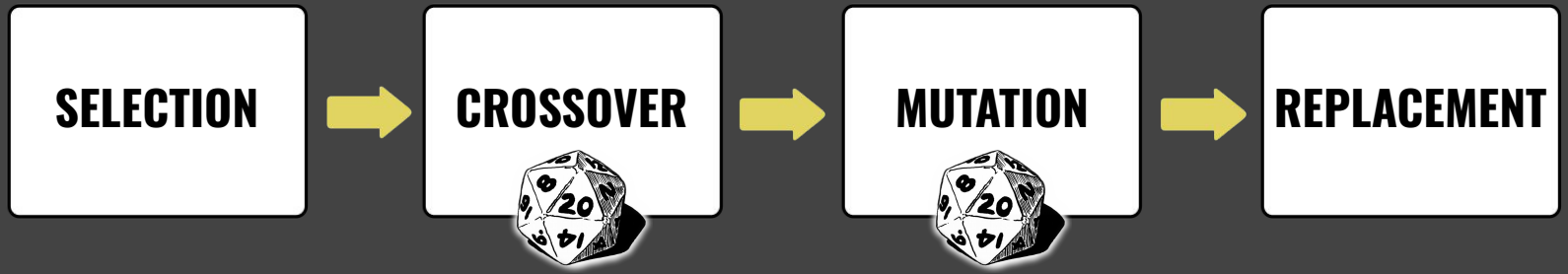
2 + 4 + 1 + 5 = **12**

1	4	2	3
1	2	3	4

Genetic Algorithm operators

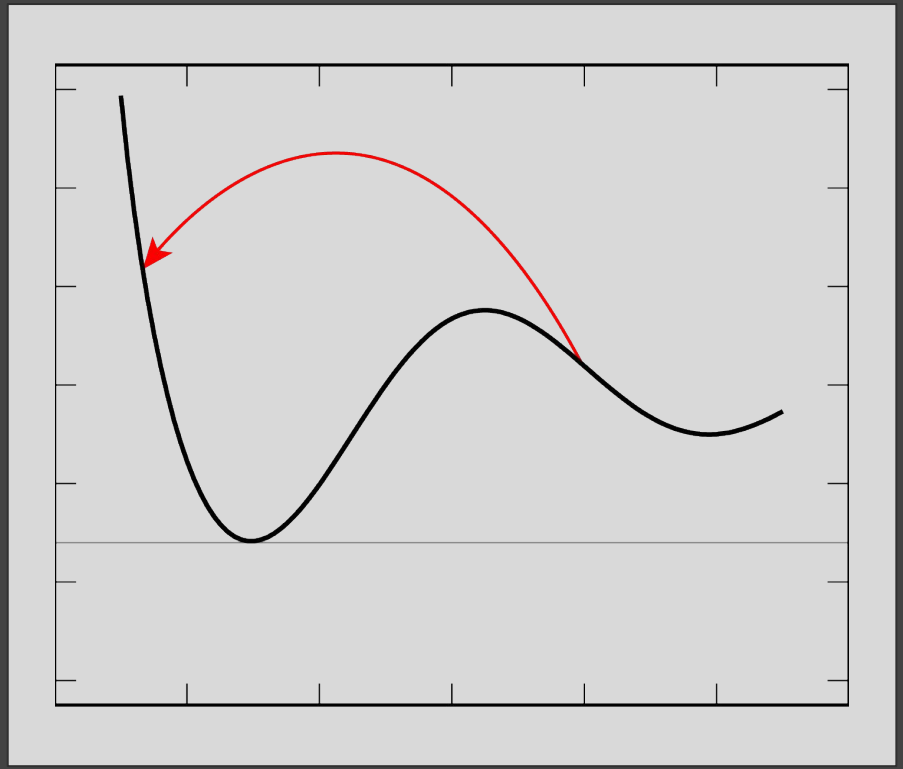


Genetic Algorithm operators



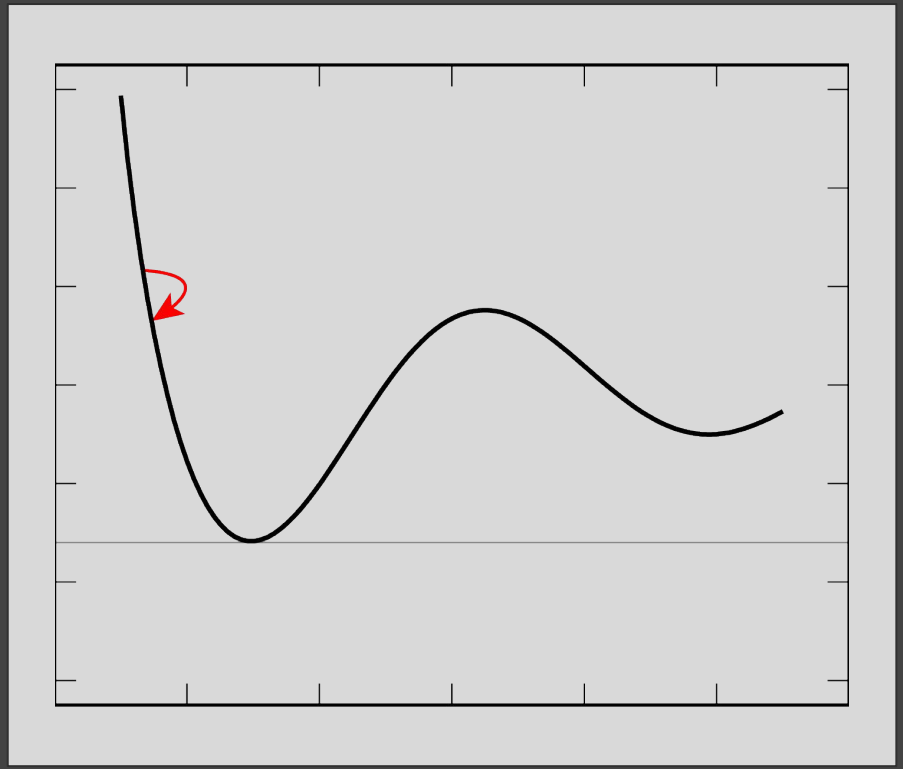
Crossover

- High probabilities
- Exploration > Exploitation

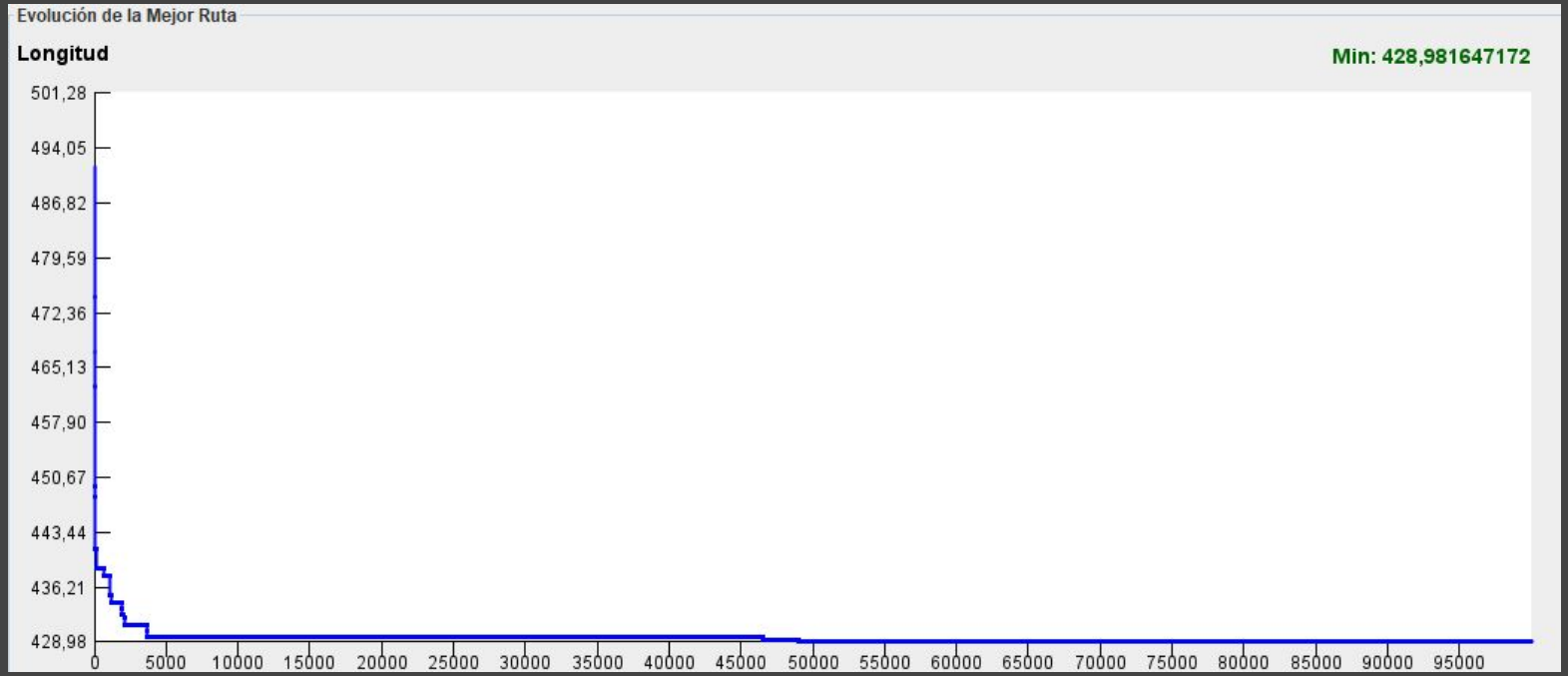


Mutation

- Low probabilities
- Exploration < Exploitation

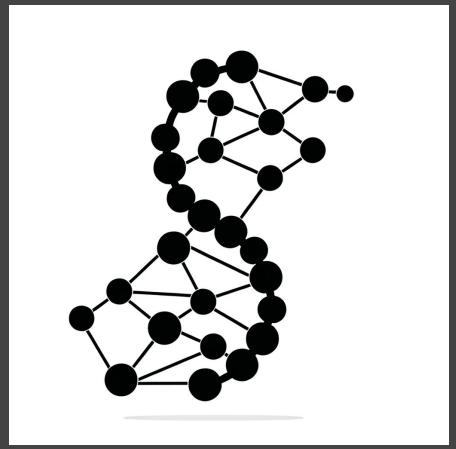


Premature convergence



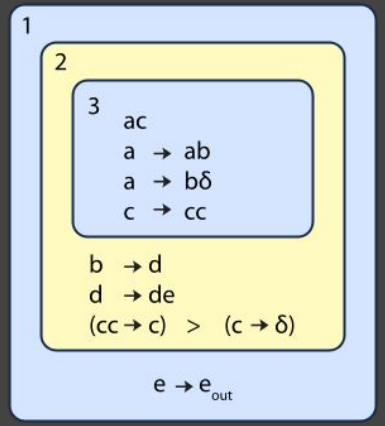
Now yes! Membrane Algorithms

Metaheuristics



Membrane Computing

Membrane Algorithms



Membrane Algorithm

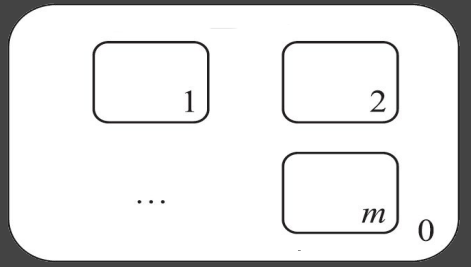
Membrane structure

Objects

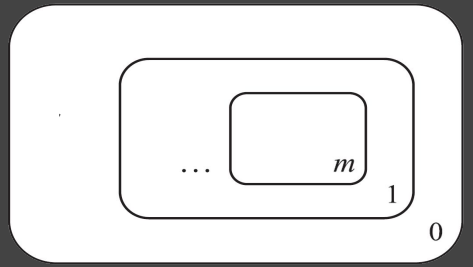
Communication rules

Evolution rules

OLMS



NMS



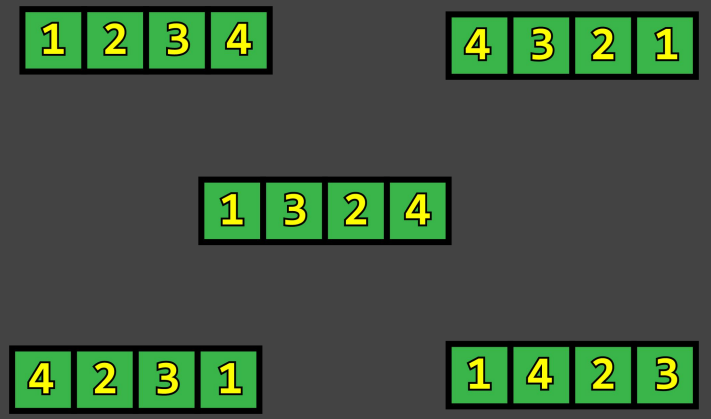
Membrane Algorithm

Membrane structure

Objects

Communication rules

Evolution rules



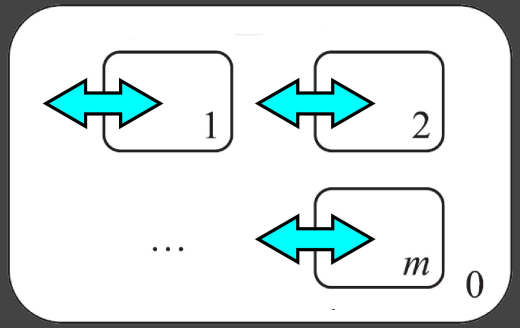
Membrane Algorithm

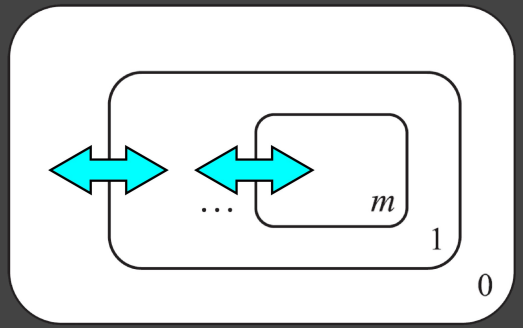
Membrane structure

Objects

Communication rules

Evolution rules





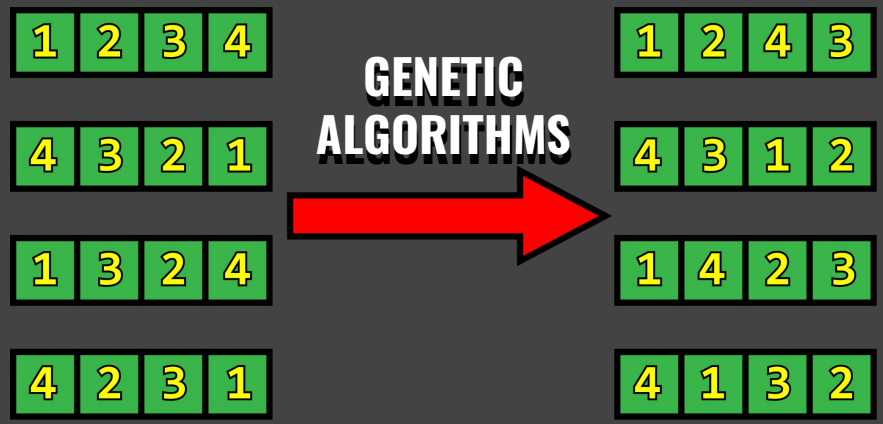
Membrane Algorithm

Membrane structure

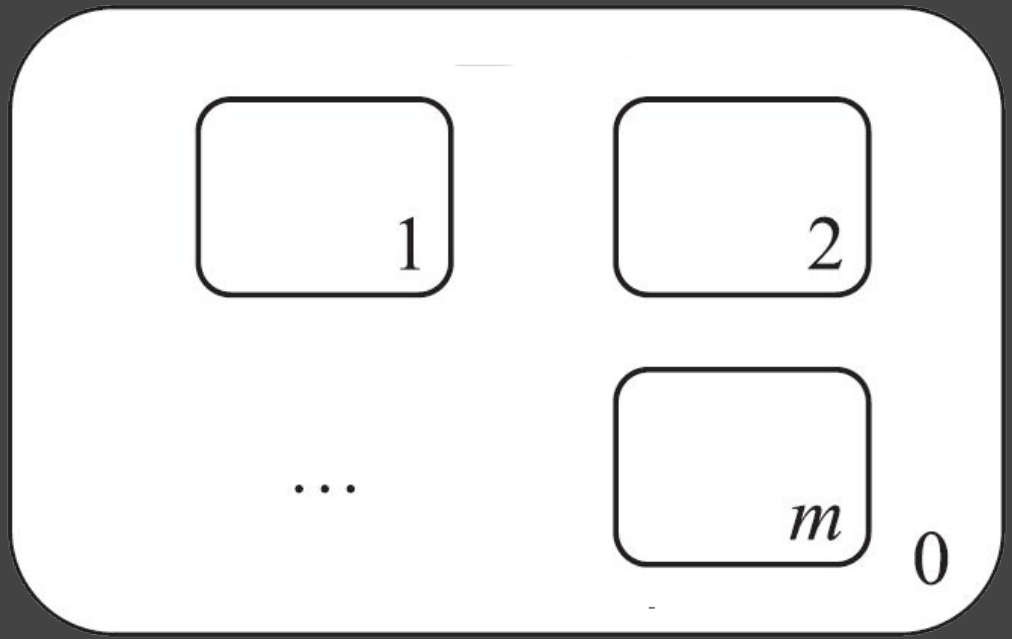
Objects

Communication rules

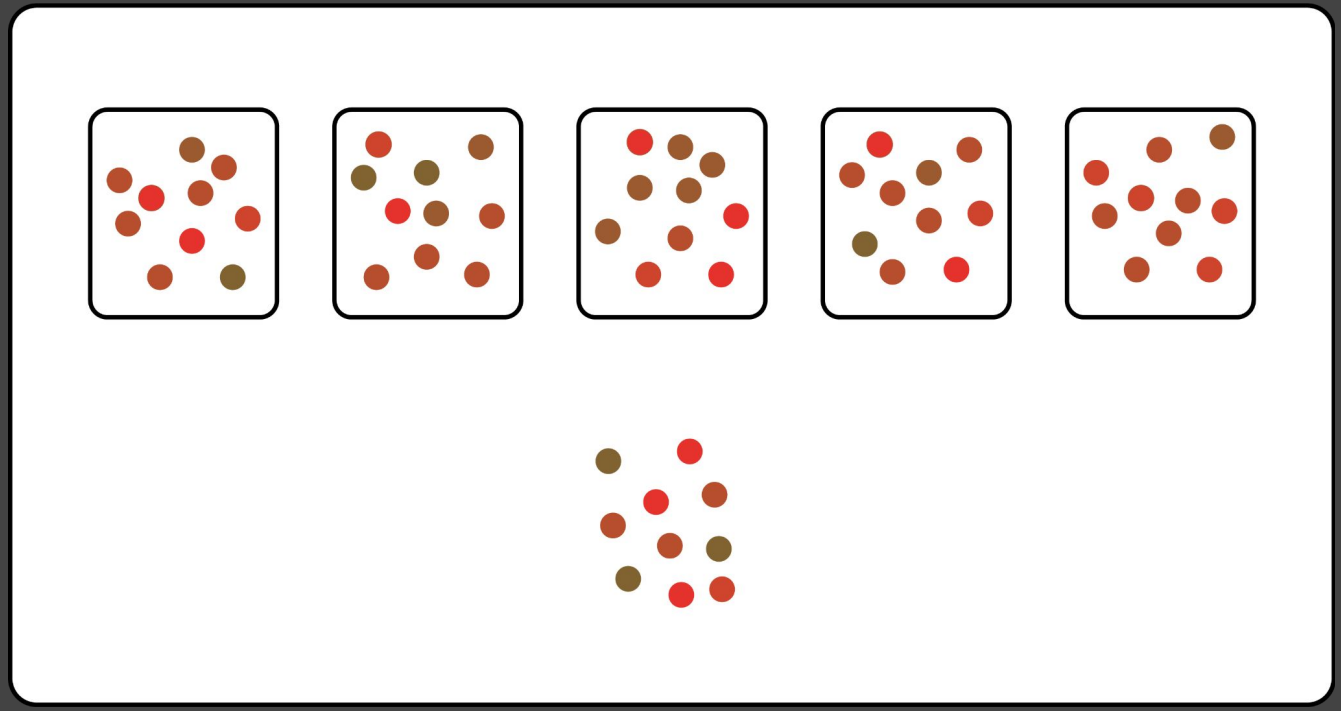
Evolution rules



One **L**evel **M**embrane **S**ystem

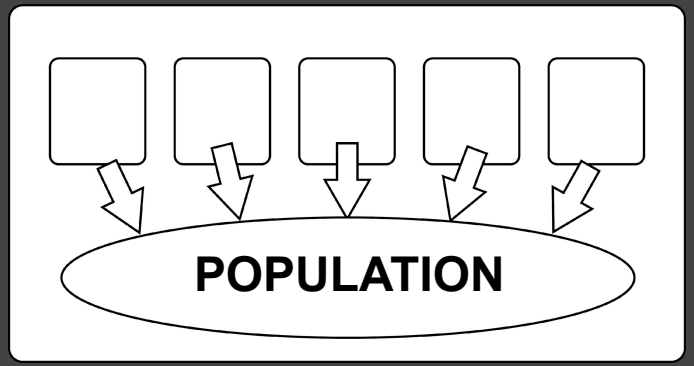


Population of chromosome objects

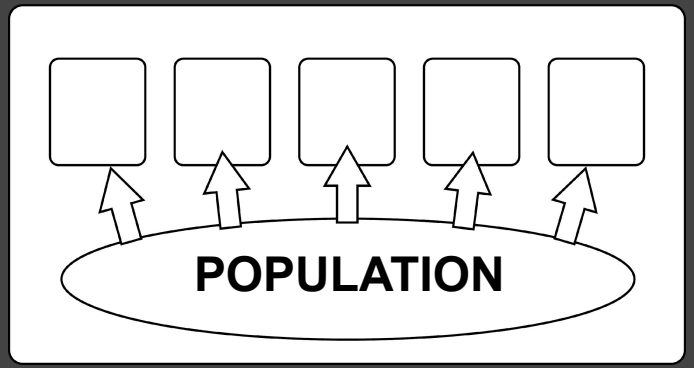


Communication rules

Send-out rule: Proportional



Send-in rule: Copy



Evolution rule: Genetic Algorithm

Genetic Algorithm Performance with Different Selection Strategies in Solving TSP

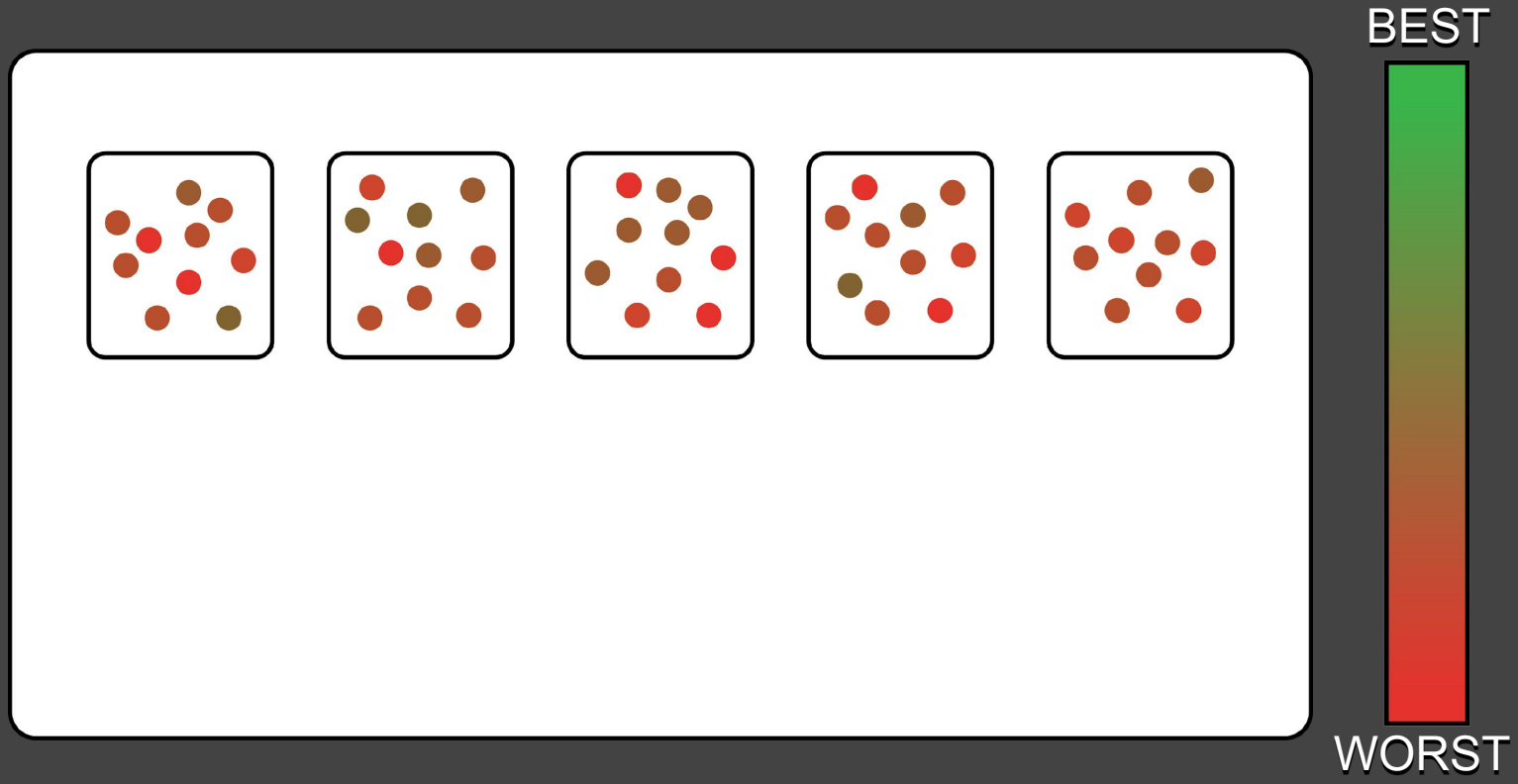
Noraini Mohd Razali, John Geraghty

***Abstract*—A genetic algorithm (GA) has several genetic operators that can be modified to improve the performance of particular implementations. These operators include parent selection, crossover and mutation. Selection is one of the important operations in the GA process. There are several ways for selection. This paper presents the comparison of GA performance in solving travelling salesman problem (TSP) using different parent selection strategy. Several TSP instances were tested and the results show that tournament selection strategy outperformed proportional roulette wheel and rank-based roulette wheel selections, achieving best solution quality with low computing times. Results also reveal that tournament and proportional roulette wheel can be superior to the rank-based roulette wheel selection for smaller problems only and become susceptible to premature convergence as problem size increases.**

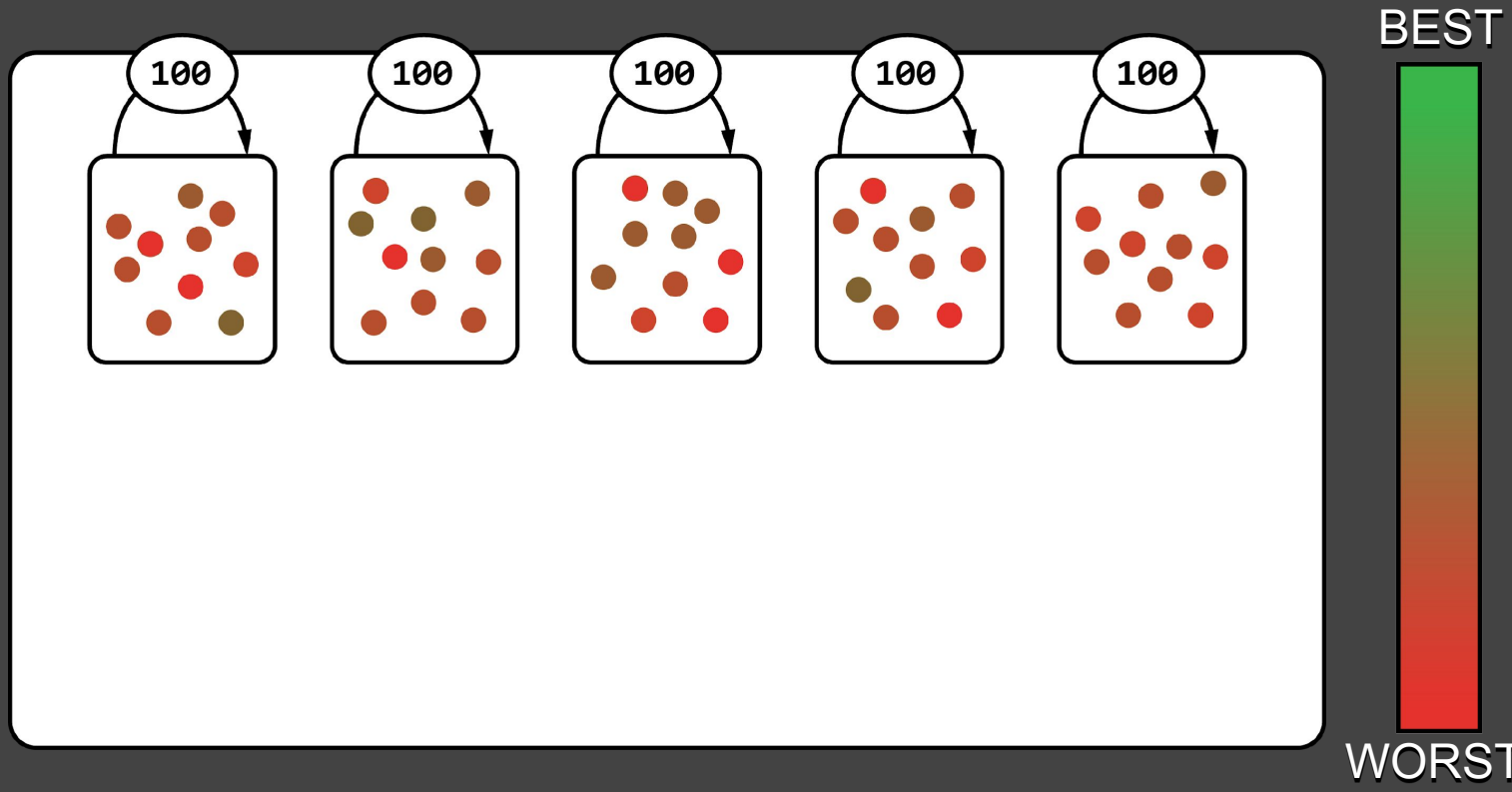
***Index Terms*— Genetic algorithm, Selection, Travelling salesman problem, Optimization**

The different selection strategy used in the GA process will significantly affect the performance of the algorithm differently. This study is intended to examine the performance of GA when using different selection strategy specifically in solving the travelling salesman problem (TSP). TSP is a classical example of a NP-hard combinatorial optimization problem. Many production and scheduling problems can be reduced to a simple concept that there is a salesman who must travel from city to city, visiting each city exactly once and returning to the home city [2]. It is possible for the salesman to select the orders of the cities visited so that the total distances travelled in his tour is as small as possible which will apparently save him time and money [2]. Although TSP is conceptually simple, it is difficult to obtain an optimal solution. The main difficulty of this problem is the enormous number of possible tours; $(n-1)!/2$ for symmetric n cities tour. As the number of cities in the problem increases, the numbers of permutations of valid

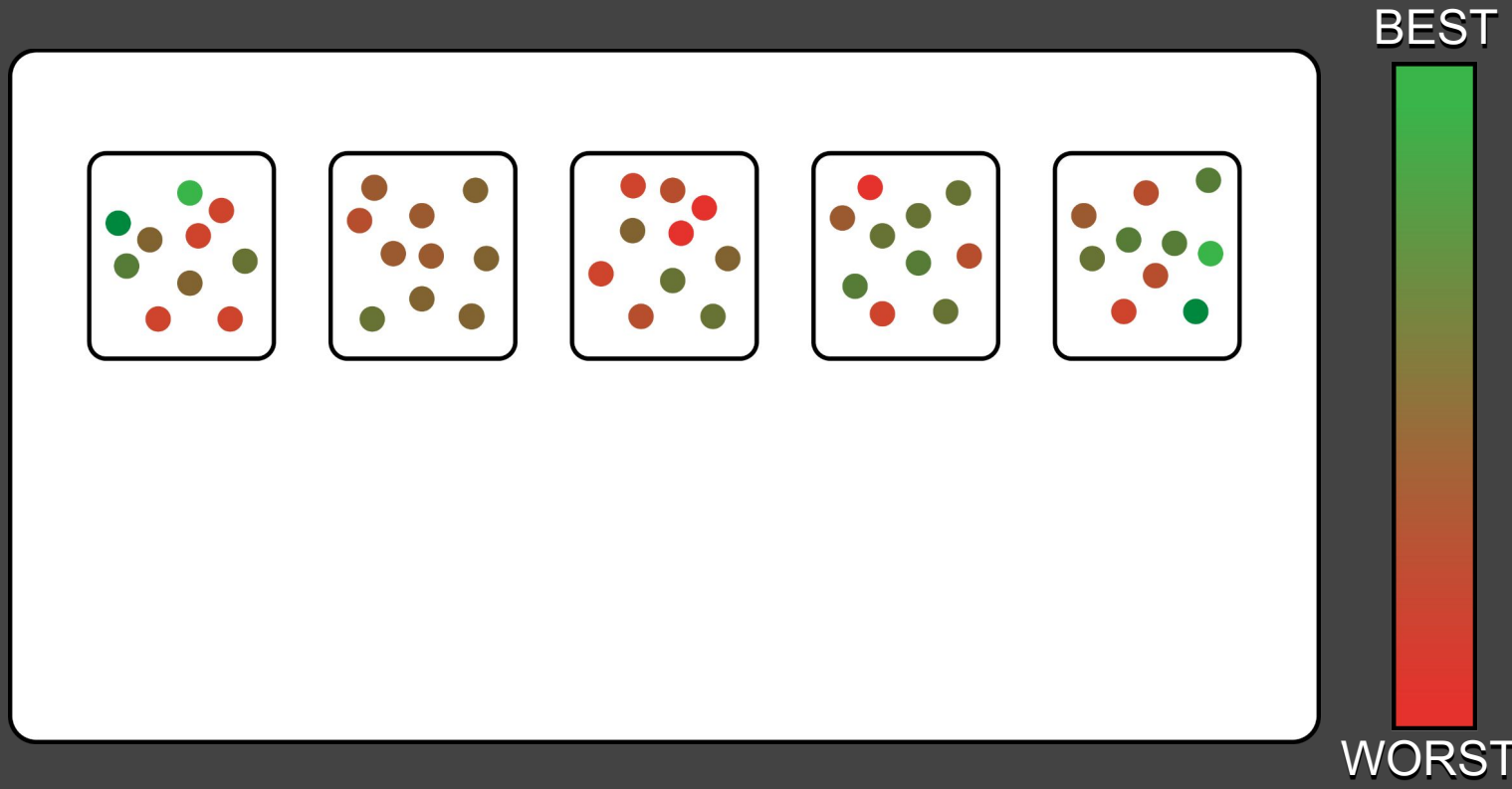
Generate initial population



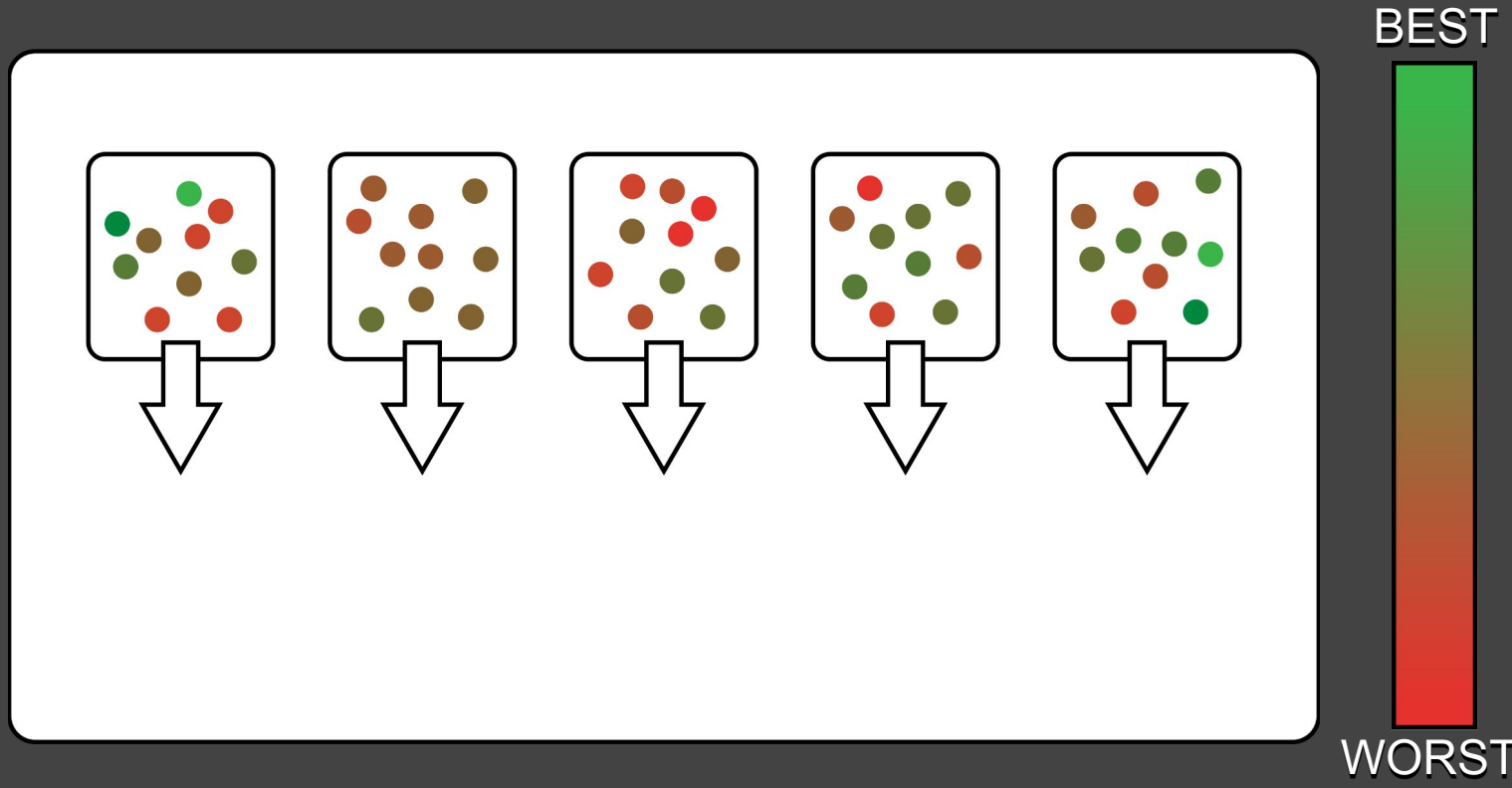
Main loop



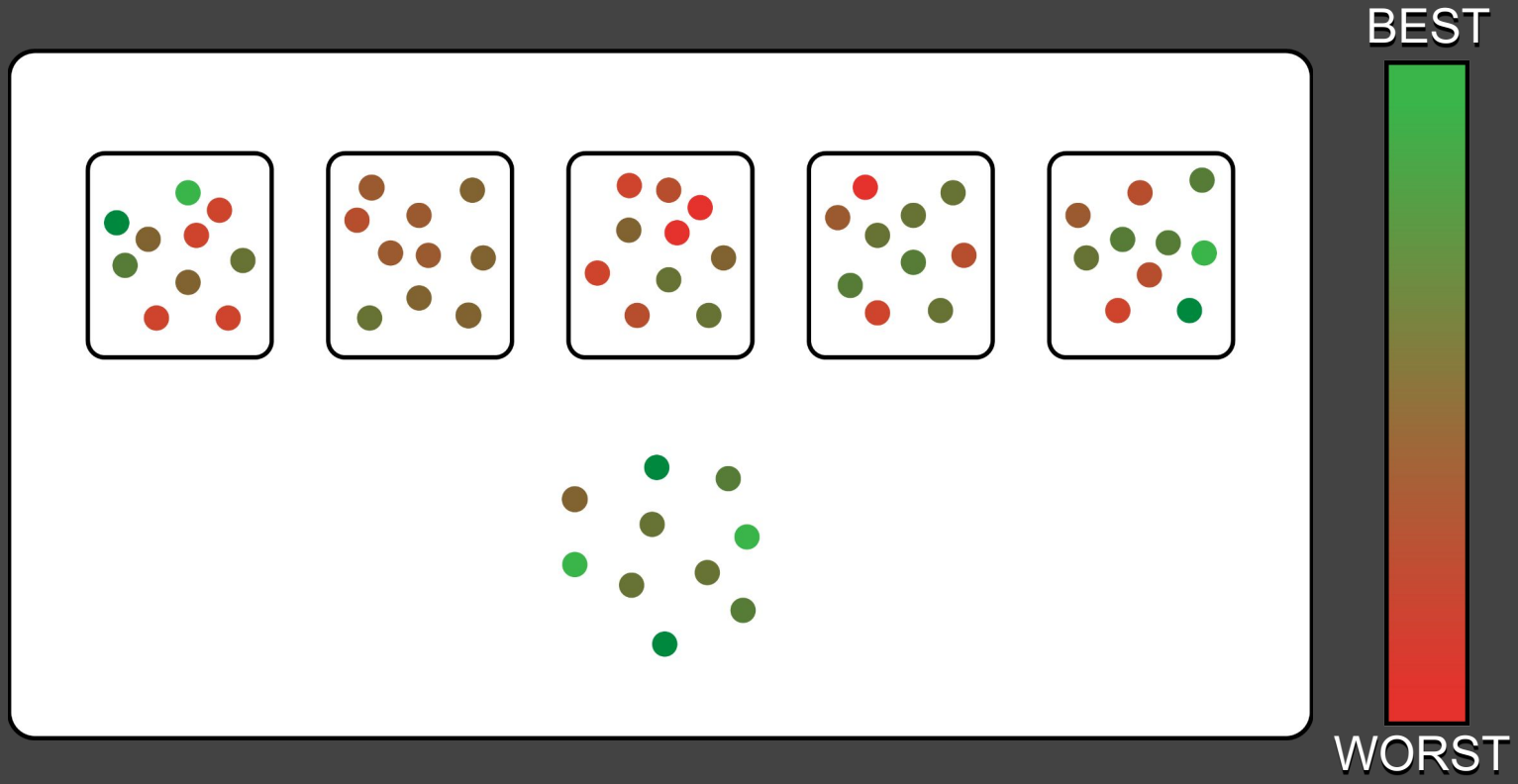
Main loop



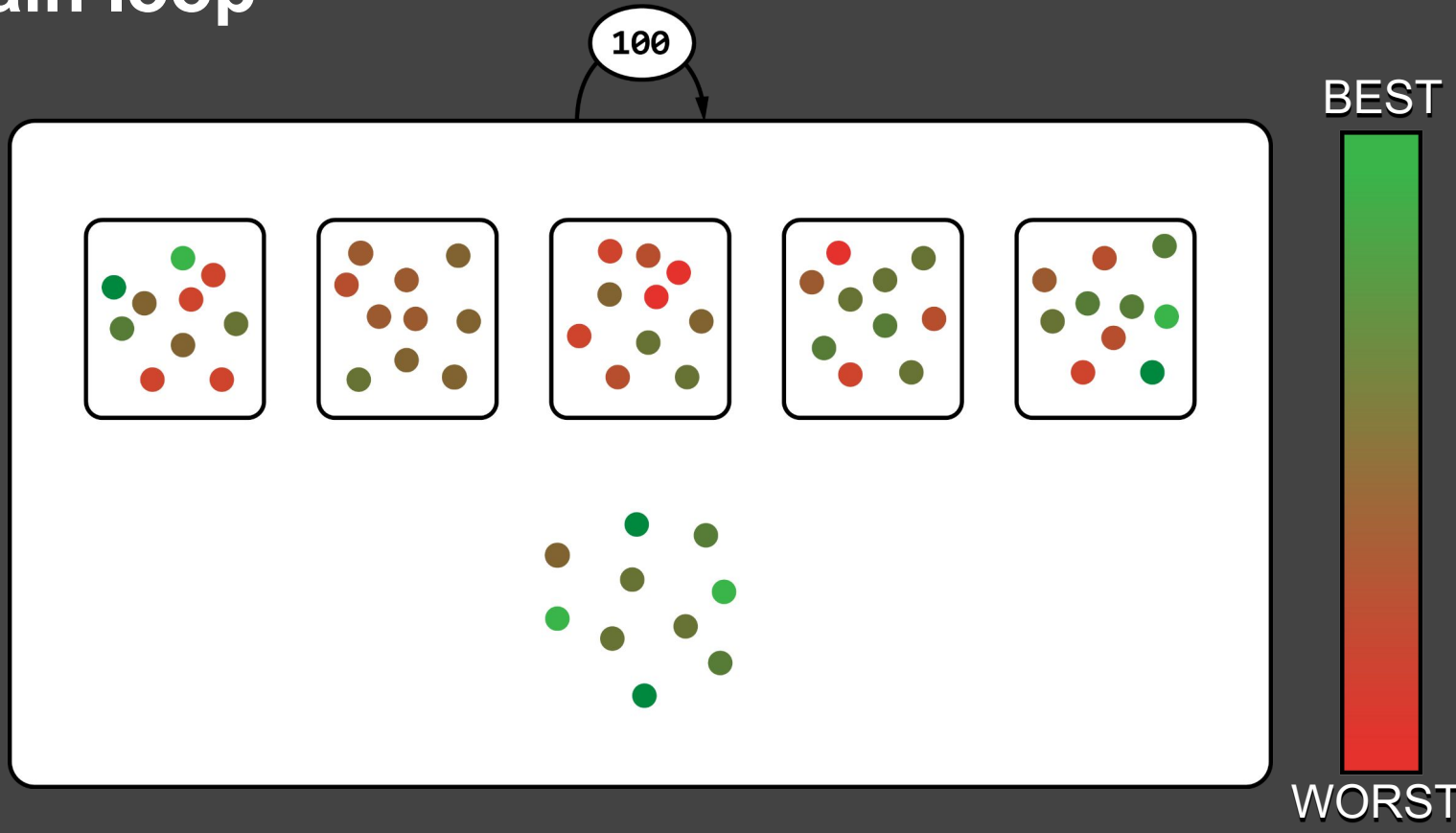
Main loop



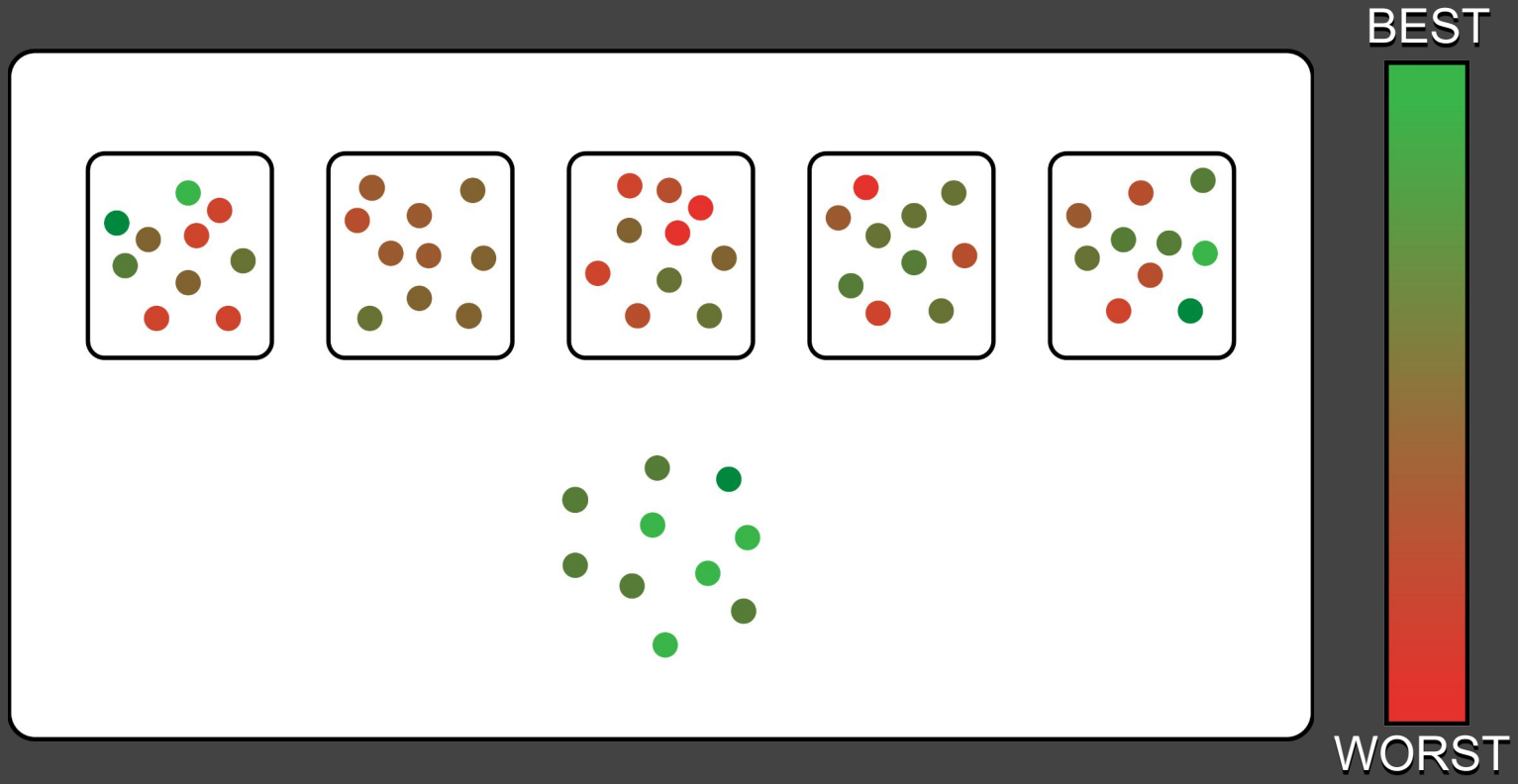
Main loop



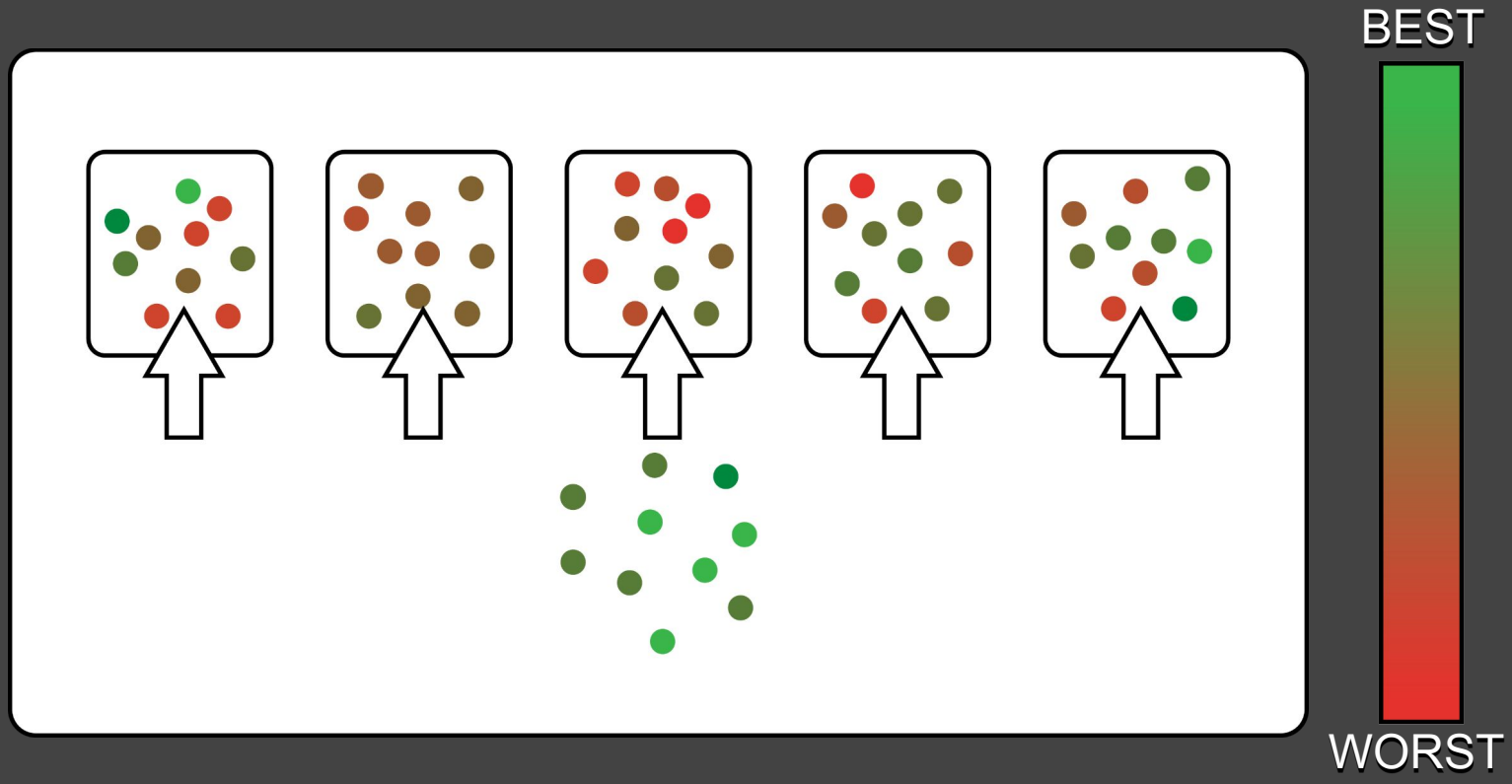
Main loop



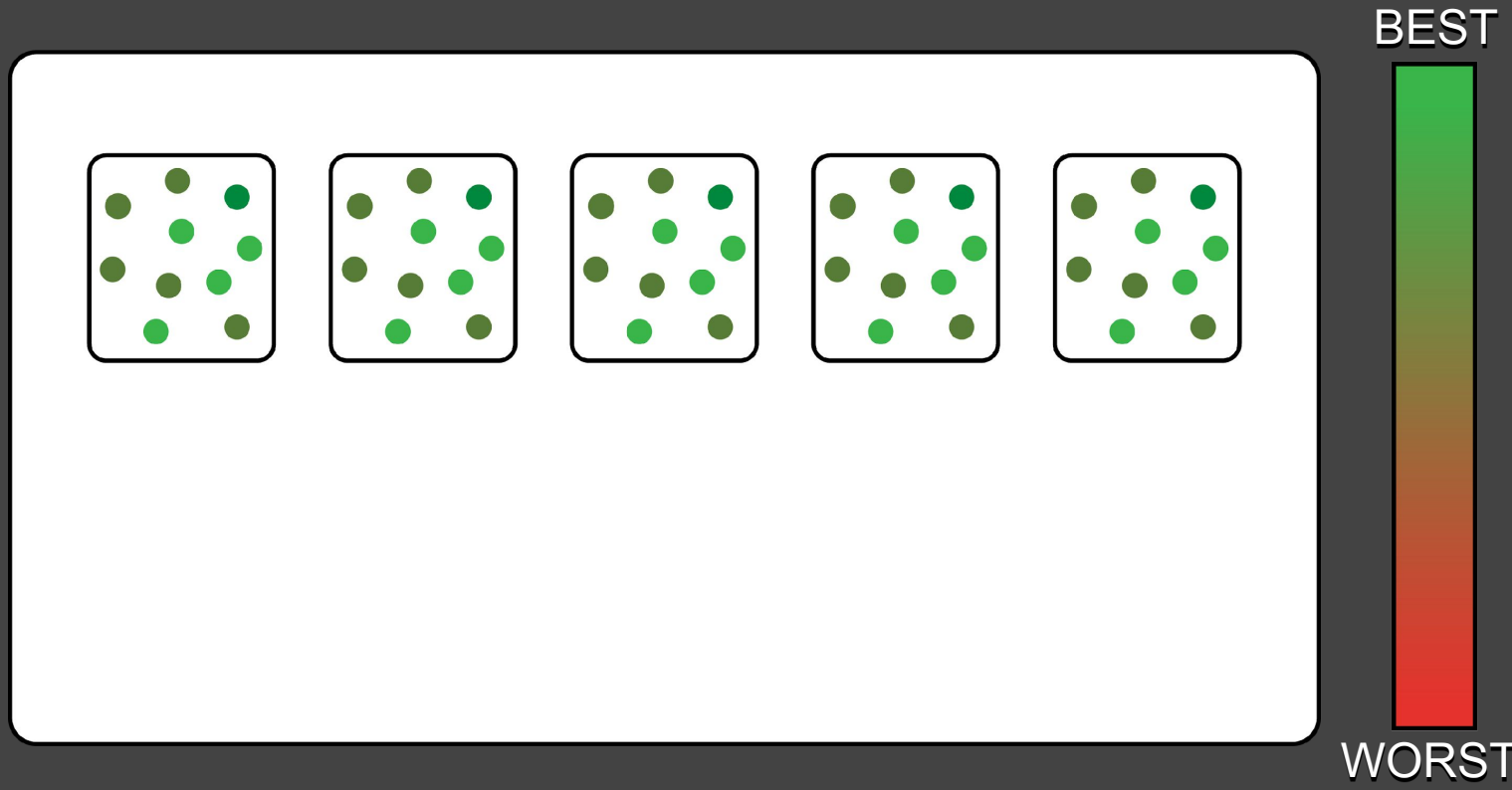
Main loop

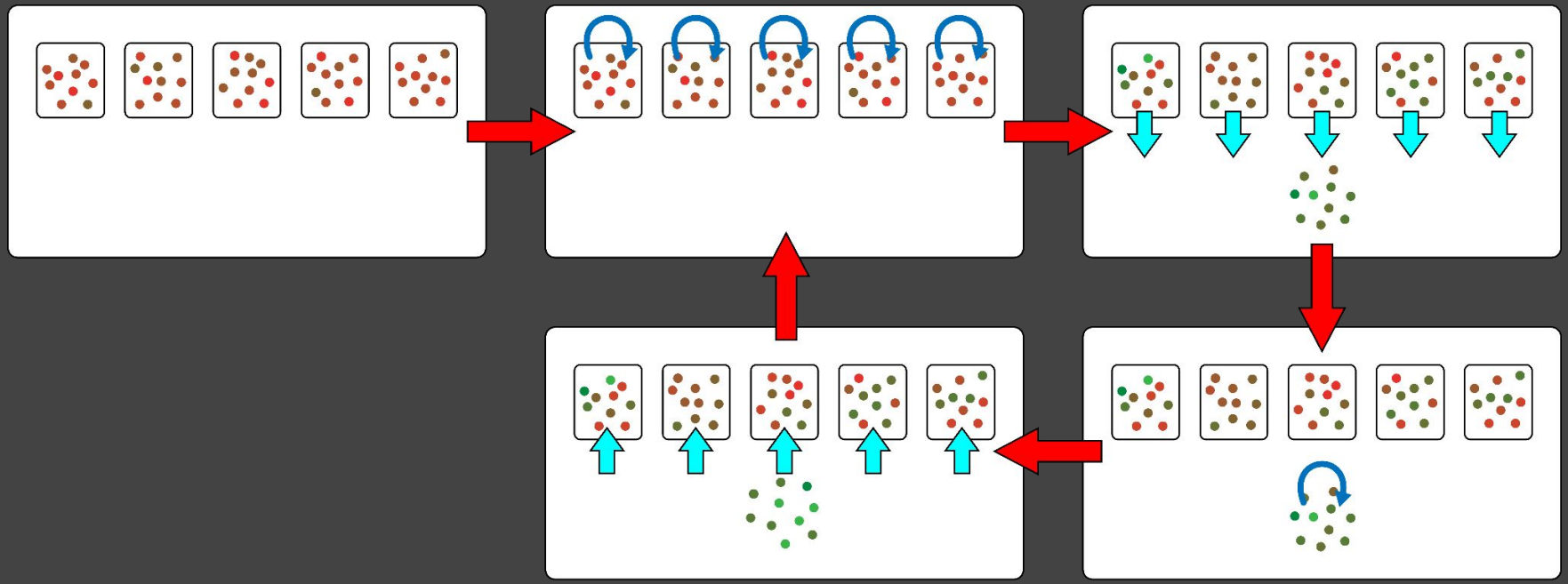


Main loop



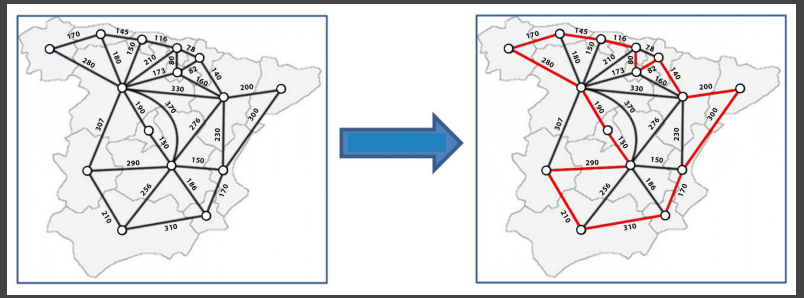
Main loop

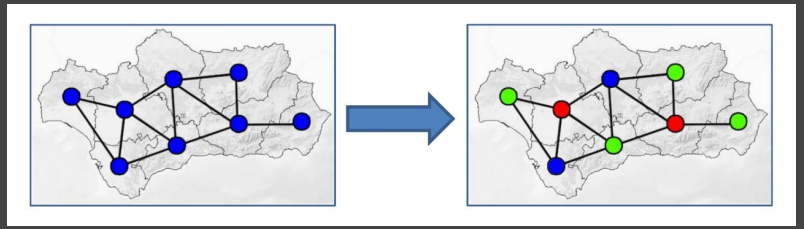




As objectives of my thesis

- Problems
- Membrane Structures
- Number of membranes
- Size of population
- Iterations
- Crossover and mutation probabilities
- Communication rules
- Evolution rules
- Develop a framework

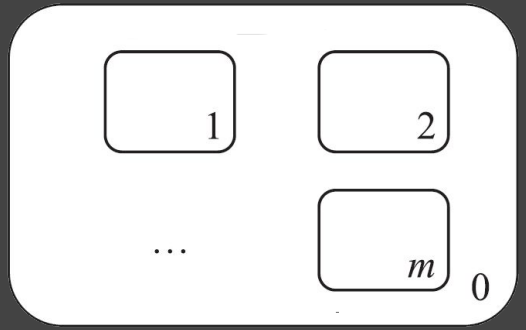




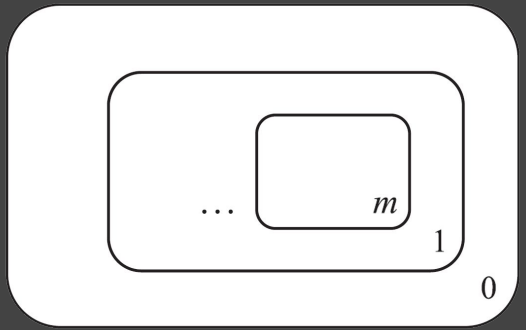
As objectives of my thesis

- Problems
- Membrane Structures
- Number of membranes
- Size of population
- Iterations
- Crossover and mutation probabilities
- Communication rules
- Evolution rules
- Develop a framework

OLMS

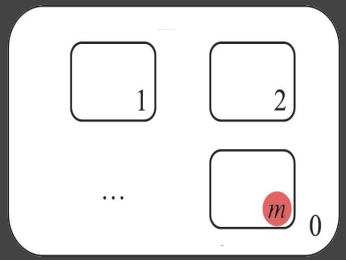


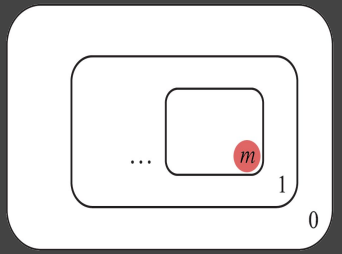
NMS

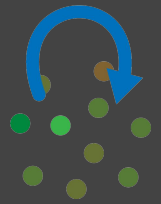


As objectives of my thesis

- Problems
- Membrane Structures
- Number of membranes
- Size of population
- Iterations
- Crossover and mutation probabilities
- Communication rules
- Evolution rules
- Develop a framework





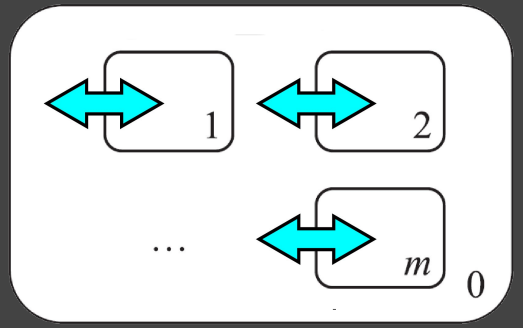


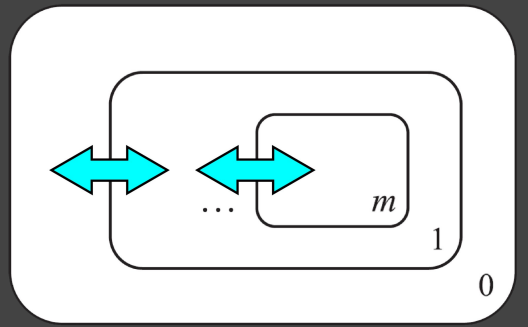
As objectives of my thesis

- Problems
- Membrane Structures
- Number of membranes
- Size of population
- Iterations
- Crossover and mutation probabilities
- Communication rules
- Evolution rules
- Develop a framework

As objectives of my thesis

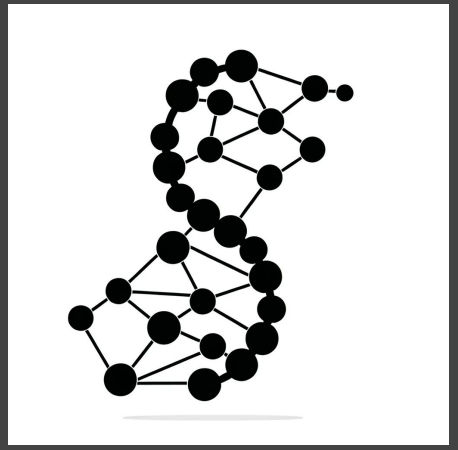
- Problems
- Membrane Structures
- Number of membranes
- Size of population
- Iterations
- Crossover and mutation probabilities
- Communication rules
- Evolution rules
- Develop a framework

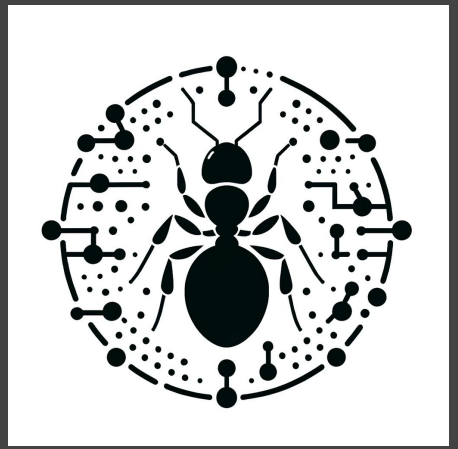




As objectives of my thesis

- Problems
- Membrane Structures
- Number of membranes
- Size of population
- Iterations
- Crossover and mutation probabilities
- Communication rules
- Evolution rules
- Develop a framework





As objectives of my thesis

- Problems
- Membrane Structures
- Number of membranes
- Size of population
- Iterations
- Crossover and mutation probabilities
- Communication rules
- Evolution rules
- Develop a framework

Papers presented

Towards a General Framework for Membrane Algorithms

A novel solution for GCP based on an OLMS membrane algorithm with dynamic operators

José Antonio Andreu-Guzmán[1] · Luis Valencia-Cabrera[1]

Future work

- Different configurations of genetic algorithms in each membrane.
- Incorporation other rules of Membrane Computing.
- Parallelising and optimise with hardware.

Ideas?

Thank you

jandreu@us.es