Population-based algorithm

Genetic algorithm with the following operators:

- Selection:
  - Ranking based on objective function
  - Randomly selected according to ranking position

$$p(x_i) = \frac{\frac{1}{r_i}}{\sum_{j=1}^{n} \frac{1}{r_j}}$$

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[0 0 0 0 0 0 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[**0 1 1 0** 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[0 0 0 0 0 0 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[**0 1 1 0** 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[**0 1 1 0** 0 0 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 **0** 1 1 1]	
[0 1 1 0 0 0 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 **0** 1 1 1]	
[0 1 1 0 **0** 0 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 **1** 1 1]	
[0 1 1 0 0 0 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 **1** 1 1]	
[0 1 1 0 0 **1** 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 **1** 1]	
[0 1 1 0 0 1 0 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 **1** 1]	
[0 1 1 0 0 1 **1** 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[0 1 1 0 0 1 1 0]	

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[0 1 1 0 0 1 1 0]	[0 0 1 0 1 0 1 0]

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[0 1 1 0 0 1 1 0]	[0 0 1 0 1 0 1 0]

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[0 1 1 0 0 1 1 0]	[0 1 1 0 1 0 1 0]

Population-based algorithm

Genetic algorithm with the following operators:

- Crossover:

[0 1 1 0 1 0 1 0] + [0 0 1 0 0 1 1 1]	
[0 1 1 0 0 1 1 0]	[0 1 1 0 1 0 1 0]

Population-based algorithm

Genetic algorithm with the following operators:

- Mutation: swap a 0 and a 1

  [0 1 1 0 1 0 1 0]

Population-based algorithm

Genetic algorithm with the following operators:

- Mutation: swap a 0 and a 1

  [0 1 **1** 0 1 0 1 **0**]

Population-based algorithm

Genetic algorithm with the following operators:

- Mutation: swap a 0 and a 1

  [0 1 **0** 0 1 0 1 **1**]

Population-based algorithm

Genetic algorithm with the following operators:

- Mutation: swap a 0 and a 1

  [0 1 0 0 1 0 1 1]

Population-based algorithm

Main algorithm:

1. Create uniform random population of *n* individuals
2. Select 2 individuals and cross them over
3. Repeat until *n* offsprings are generated
4. Mutate each one with probability *p*
5. Merge parents and offsprings and rule out the worst *n* individuals
6. Repeat until there is no improvement

Results

Algorithm	$n = 10$		$n = 50$		$n = 100$	
	Value	Time	Value	Time	Value	Time
Local search	**11.42**	**0.05**	**59.74**	**133.55**	**120.47**	727.67
Genetic	**11.42**	62.38	61.99	149.30	123.55	**301.64**

Combinatorial optimization: Number partitioning problem

Imanol Echeverría, Andoni Angulo