A Biased Random Key Genetic Algorithm for solving the α-neighbor p-center problem

Sergio Pérez-Peló (sergio.perez.pelo@urjc.es)

Jesús Sánchez-Oro (jesus.sanchezoro@urjc.es)

Abraham Duarte (abraham.duarte@urjc.es)

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

AGENCIA ESTATAL DE INVESTIGACIÓN

Universidad Rey Juan Carlos

Outline

Universidad Rey Juan Carlos

grafo RESEARCH

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

AGENCIA ESTATAL DE INVESTIGACIÓN

Introduction

- **Facility location problems** have been extensively studied in the literature.
- **Different variants** of this problem have appeared **over the years**.
- Facility location problems have **multiple real-world applications**.

Introduction

- In this work we address the **α-neighbor p-center** problem.
- A **customer** must be **assigned to α facilities**.
- Focuses on **providing a robust** failover **solution**.
  - Situations in which the **nearest facility is unavailable** for any reason.
- In this work, **all the points** available in the instance **can act as facilities or as clients**.

Introduction

- The **objective function** is to **minimize** the **maximum distance** between each client and its assigned αth facility.
- In mathematical terms, for each client:

$$adistance(a, Z) = min_{S \in Z}\{max_{b \in S} d(a, b)\}$$

where Z is the set of p points selected as facilities; S represents any subset of Z of size α; a and b are points in the plane; and $d(a, b)$ is de **Euclidean distance** from a to b.

Introduction

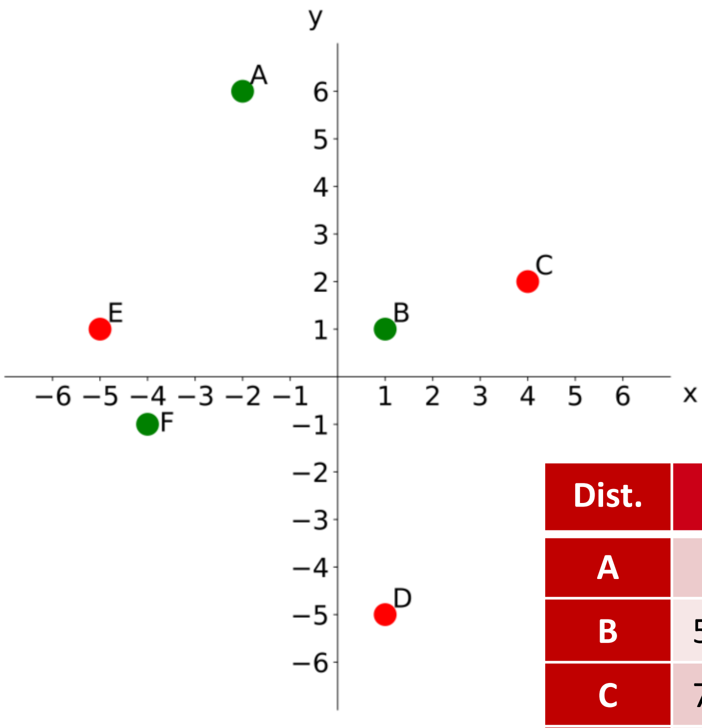
- For a **whole solution**, the objective function value is calculated as:

$$OF(Z,K) = \ max_{\alpha \in K \setminus Z} \alpha distance(a,Z)$$

where K represents the set of points in an instance.

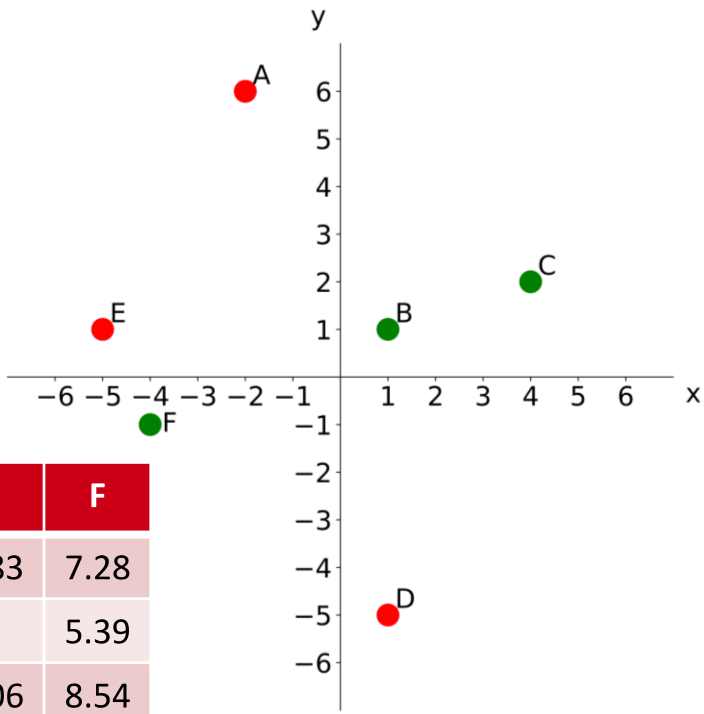
Introduction

$$n = 6, p = 3, \alpha = 2$$



$Z_1 = \{C, D, E\}$

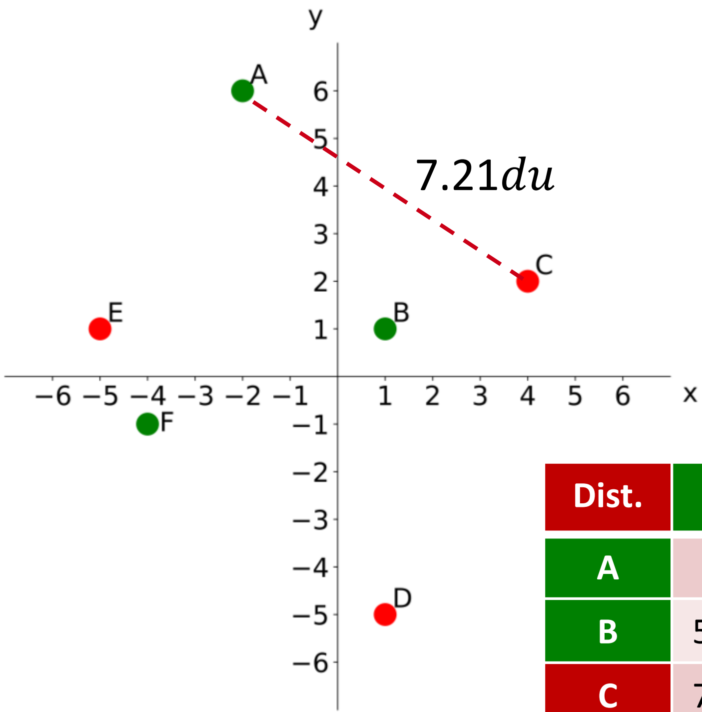
Dist.	A	B	C	D	E	F
A	0	5.83	7.21	11.4	5.83	7.28
B	5.83	0	3.16	6	6	5.39
C	7.21	3.16	0	7.62	9.06	8.54
D	11.4	6	7.62	0	8.49	6.4
E	5.83	6	9.06	8.49	0	2.24
F	7.28	5.39	8.54	6.4	2.24	0



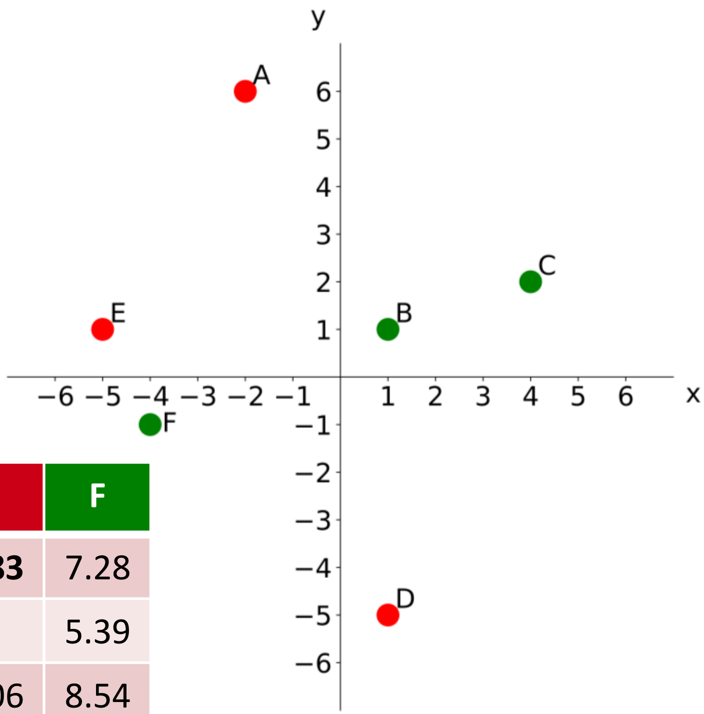
$Z_1 = \{A, D, E\}$

Introduction

$$n = 6, p = 3, \alpha = 2$$



$Z_1 = \{C, D, E\}$

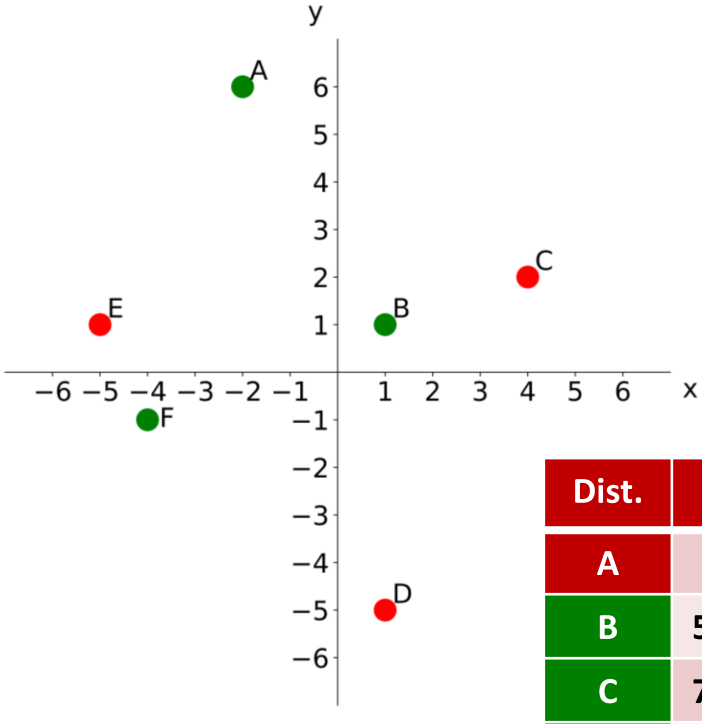


$Z_2 = \{A, D, E\}$

Dist.	A	B	C	D	E	F
A	0	5.83	**7.21**	11.4	**5.83**	7.28
B	5.83	0	**3.16**	**6**	6	5.39
C	7.21	3.16	0	7.62	9.06	8.54
D	11.4	6	7.62	0	8.49	6.4
E	5.83	6	9.06	8.49	0	2.24
F	7.28	5.39	8.54	**6.4**	**2.24**	0

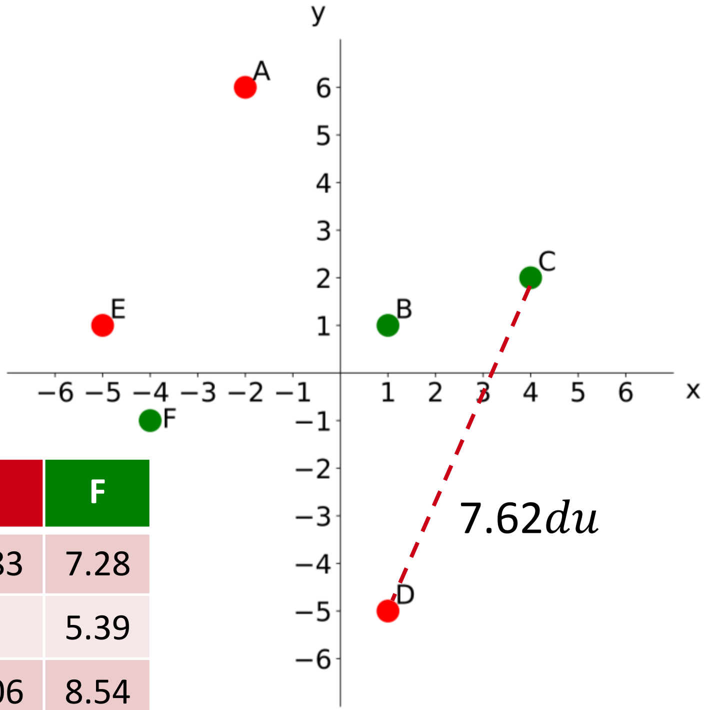
Introduction

$$n = 6, p = 3, \alpha = 2$$



$Z_1 = \{C, D, E\}$

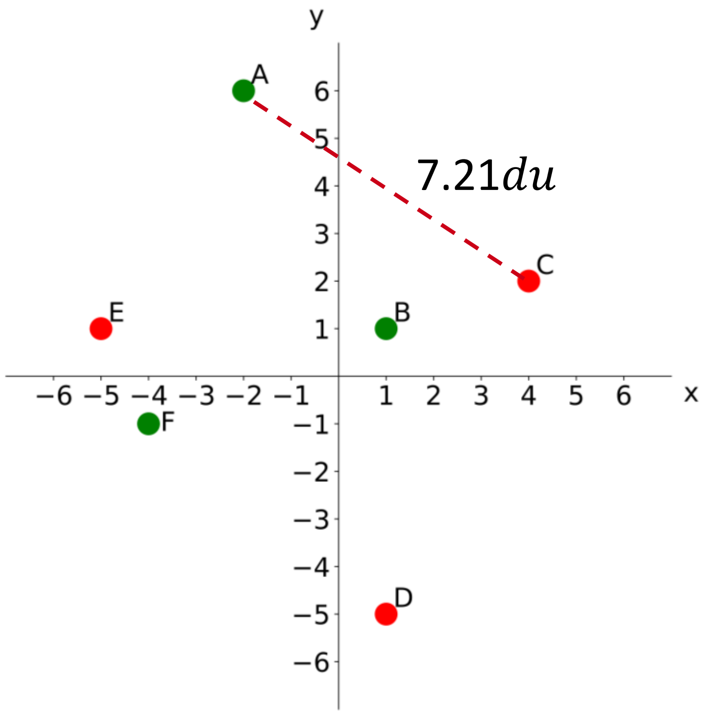
Dist.	A	B	C	D	E	F
A	0	5.83	7.21	11.4	5.83	7.28
B	**5.83**	0	3.16	**6**	6	5.39
C	**7.21**	3.16	0	**7.62**	9.06	8.54
D	11.4	6	7.62	0	8.49	6.4
E	5.83	6	9.06	8.49	0	2.24
F	7.28	5.39	8.54	**6.4**	**2.24**	0



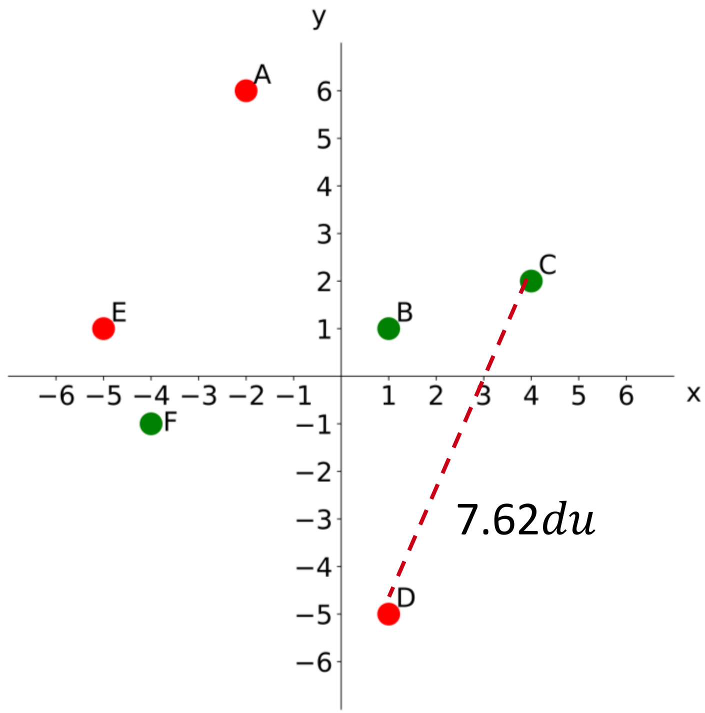
$Z_2 = \{A, D, E\}$

Introduction

$$n = 6, p = 3, \alpha = 2$$



$\boldsymbol{Z_1} = \{\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}\}$



$Z_2 = \{A, D, E\}$

Methodology
BRKGA

- **B**iased **R**andom **K**ey **G**enetic **A**lgorithm (Gonçalves and Resende, 2011).
  - **Based on** the behavior of **classical genetic algorithms** (chromosomes representing solutions encoded in some way).

Methodology

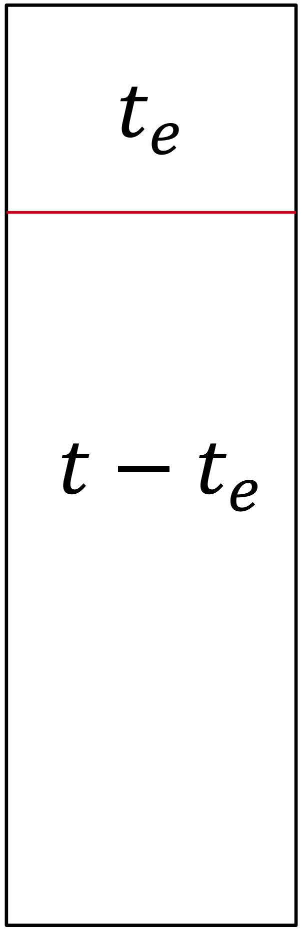
BRKGA

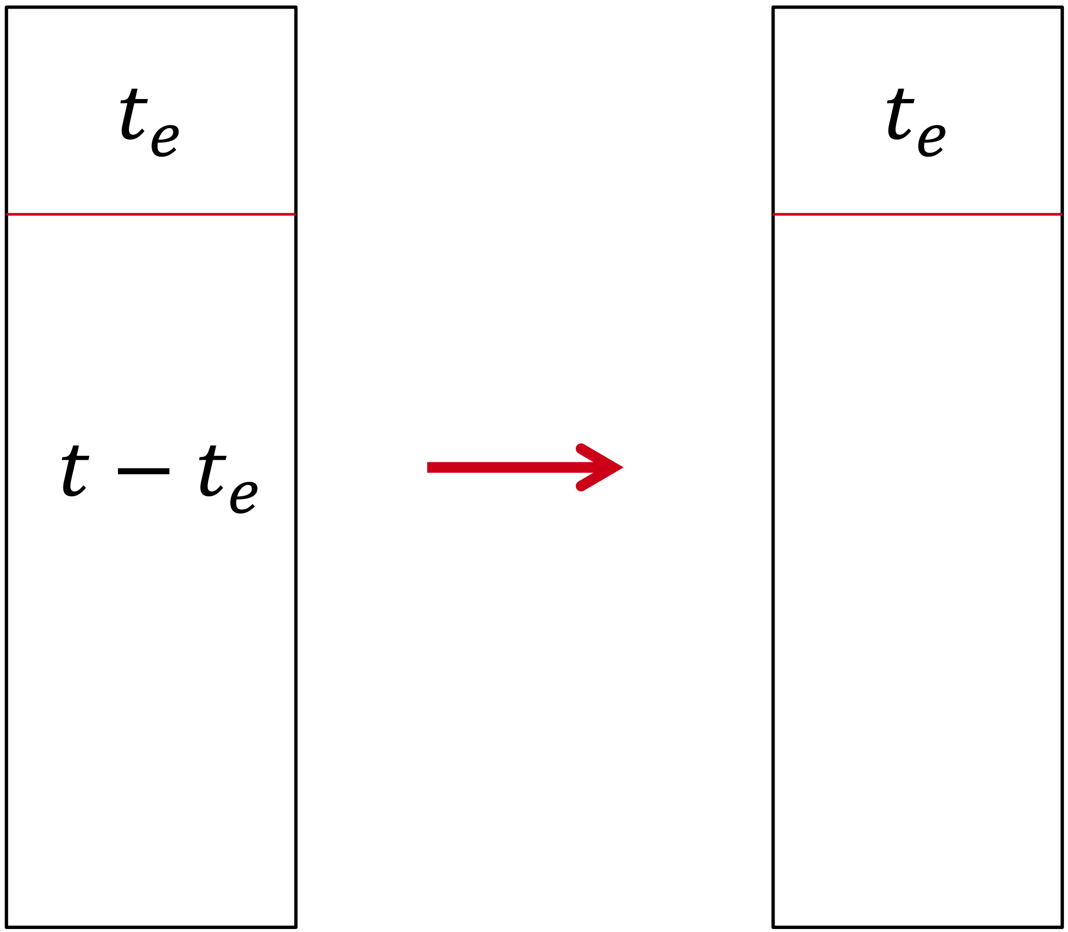
- **Chromosomes** encoded by vectors of real numbers in the interval [0, 1].
  - These numbers are commonly known as **alleles**.
  - Represent the **random keys**.
- They must be translated into the context of the solution in a **decoding phase**.

t

Methodology

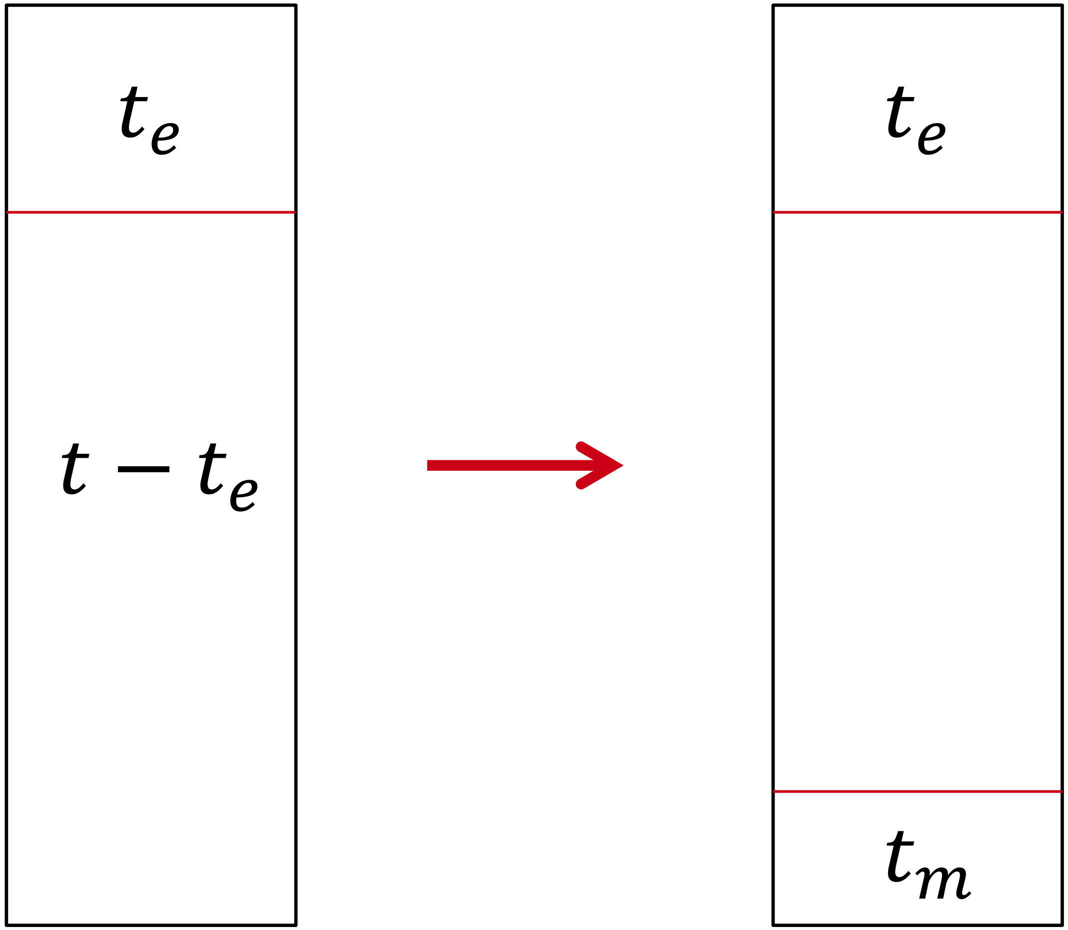
BRKGA





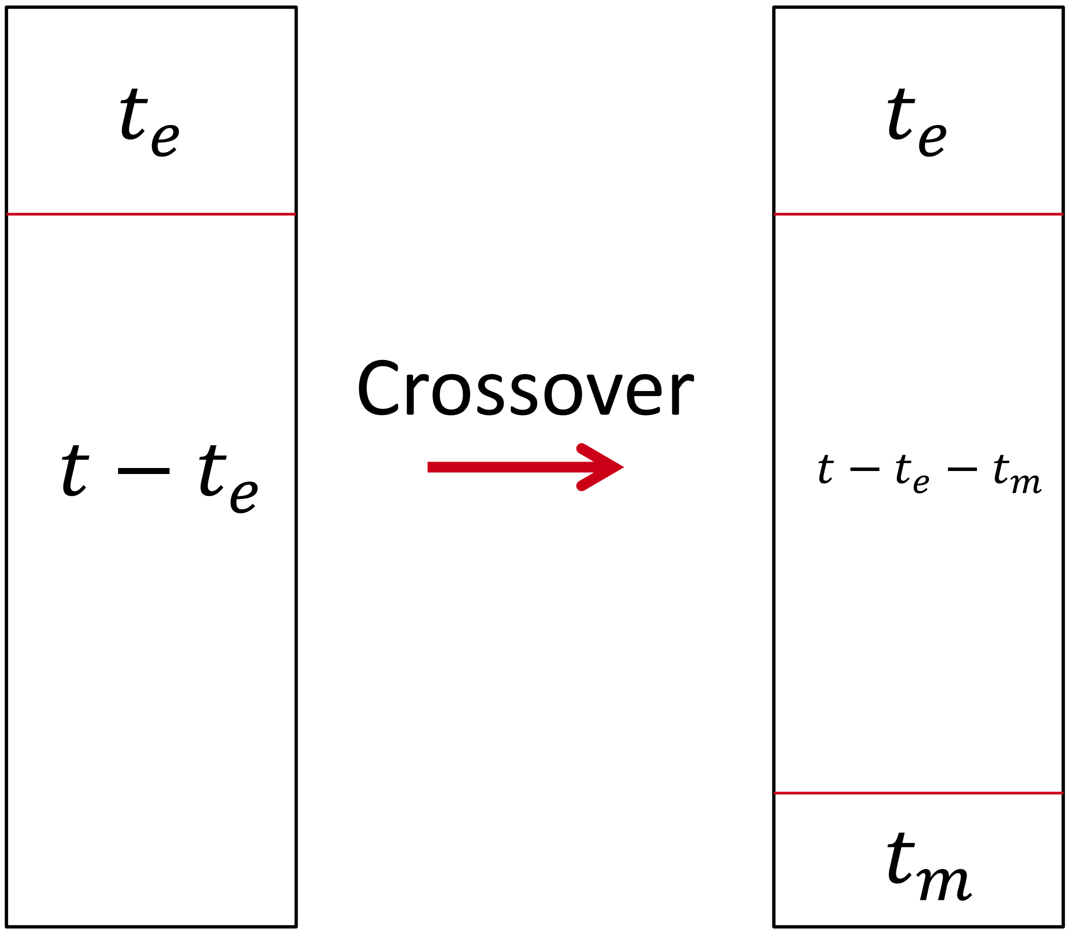
t_e
$t - t_e$
t_e

Methodology
BRKGA

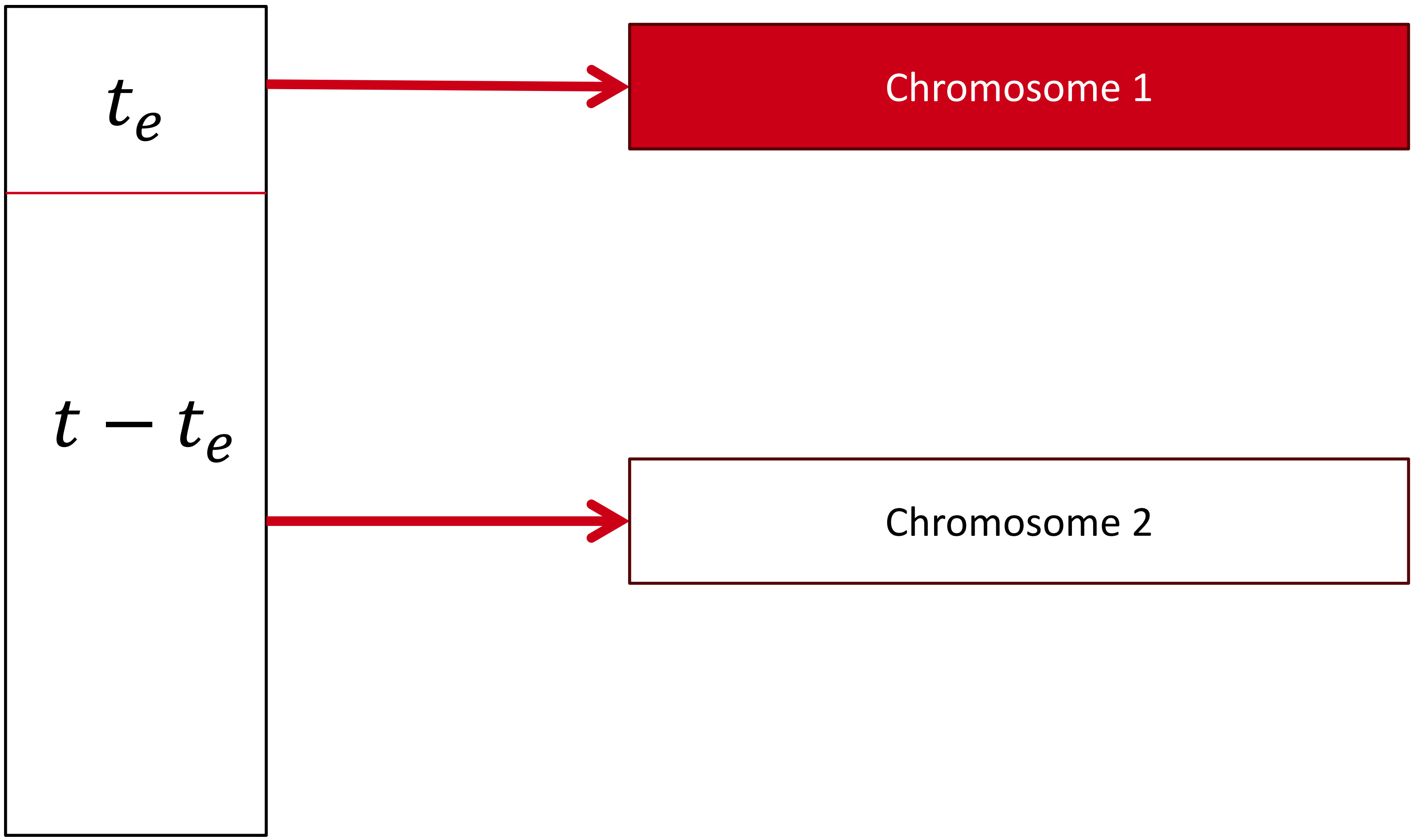


Methodology

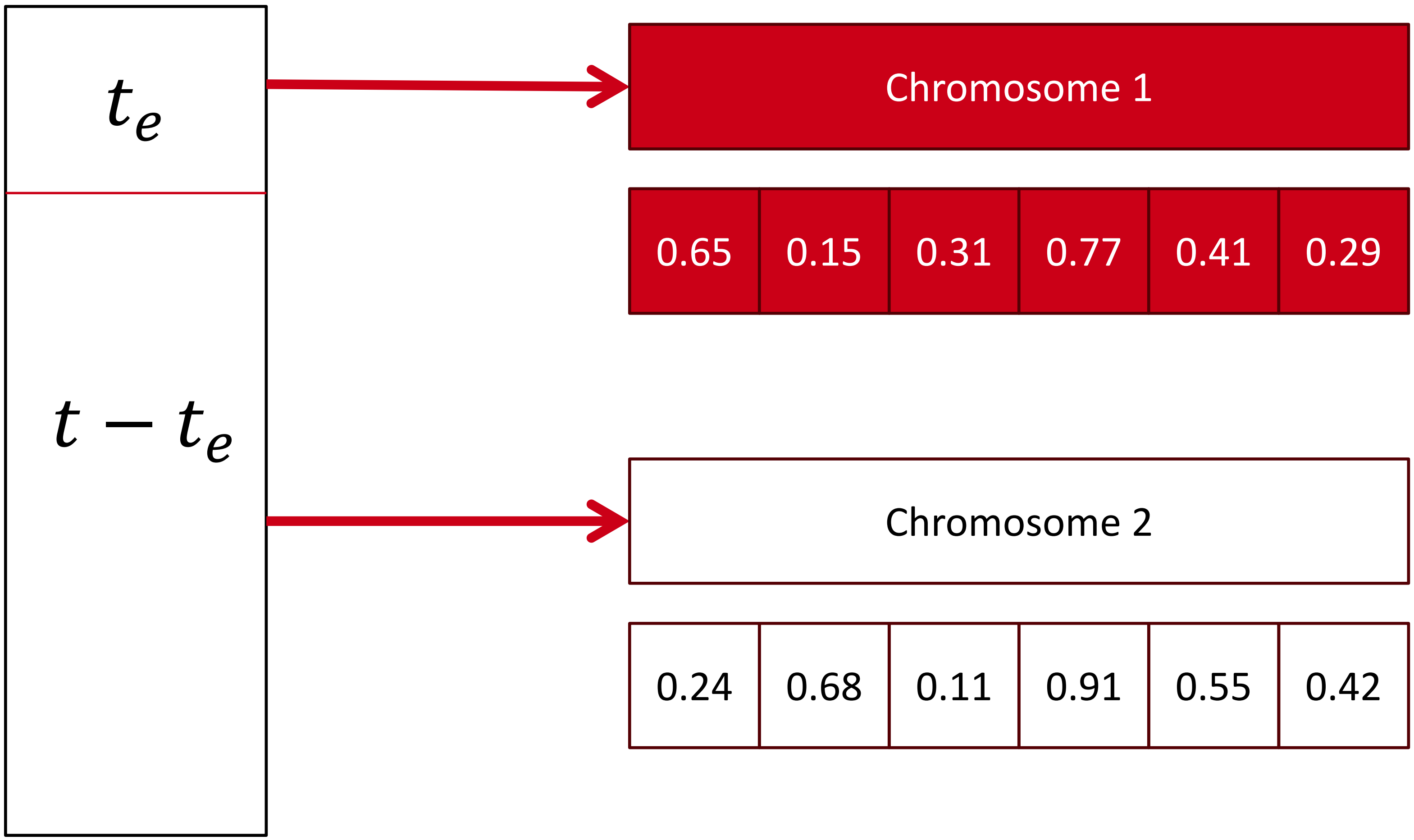
BRKGA



Methodology
BRKGA



Methodology
BRKGA



Methodology

BRKGA

Chromosome 1

0.65	0.15	0.31	0.77	0.41	0.29

Chromosome 2

0.24	0.68	0.11	0.91	0.55	0.42

$$\rho_e = 0.60$$

Random Choice

0.74	0.12	0.60	0.94	0.54	0.84
$\geq$	$<$	$\geq$	$\geq$	$<$	$\geq$

Offspring Chromosome

0.65	0.68	0.31	0.77	0.55	0.29

Methodology

Decoder

- It is necessary to define a method to **convert** the **allele** values **into** elements of the **solution**.
- Each **position** in the chromosome represents the **node** with the **same id**.

Methodology
Decoder

Node id	0	1	2	3	4	5
	0.65	0.68	0.31	0.77	0.55	0.29

Methodology

Decoder

Node id	0	1	2	3	4	5
	0.65	0.68	0.31	0.77	0.55	0.29

↓

Node id	3	1	0	4	2	5
	0.77	0.68	0.65	0.55	0.31	0.29

Methodology

Decoder

Node id	0	1	2	3	4	5
	0.65	0.68	0.31	0.77	0.55	0.29

Node id	3	1	0	4	2	5
	0.77	0.68	0.65	0.55	0.31	0.29

3	1	0	4	2	5
0.77	0.68	0.65	0.55	0.31	0.29

$$Z = \{0, 1, 3\}$$

Methodology
Local Search Method

- Once a diverse population is obtained, a **local improvement** phase is executed.
- This process is **applied to** the solutions belonging to the **elite set**.
- The **algorithm returns** the **best solution found after** the application of the **local search** method.

Methodology
Local Search Method

- Let us first define the **movement** that generates the explored **neighborhood** as:

$$Swap(Z, a, b) \leftarrow Z \cup \{b\} \setminus \{a\}$$

where $a \in Z$ and $b \notin Z$, it is, a represents an open facility and b a demand point in the solution Z.

Methodology
Local Search Method

- The exploration of this neighborhood is quite **time-consuming**.
- In this work **we reduce the number of facilities and clients** that are **swapped**.
  - The **facilities** that are closed are **only those that represent the αth closest facility** to some client
  - For each facility, **only those clients** whose **distance** to the facility is **less than** that of its αth closest client are candidates to participate in the movement.
- This **neighborhood is explored** following a classical **first improvement** strategy.

Experiments and results

- The experiments performed in this work are devoted to test the actual state of the proposal in terms of solution quality and computing time.
- As a **preliminary experimentation**, a representative subset of 39/111 instances has been tested.

Experiments and results

- Instances **derived from** the **TSP-Lib**.
  - **37 instances** solved with **three different** α values: 1, 2, 3.
  - Instances proposed in Callaghan et al. (2019)
  - $48 \leq n \leq 1323$, where n is the number of points.
  - $10 \leq p \leq 100$.
- Java 17, AMD Ryzen 5 3600 (2.2 GHz) CPU with 16GB RAM.

Experiments and results

- The **parameters** of the algorithm have been **set automatically** using the **irace package**.
  - Population size (t): fixed to 100.
  - % of elite chromosomes (t_e): $0.10t \leq t_e \leq 0.25t$.
  - % of mutant chromosomes (t_m): $0.10t \leq t_m \leq 0.30t$.
  - Allele inheritance probability (ρ_e): $0.5 \leq \rho_e \leq 0.8$.
- The proposal is compared against the best method found in literature, a GRASP with strategic oscillation approach (Sánchez-Oro et al., 2022).

Experiments and results

$t_e = 0.17, t_m = 0.11, \rho_e = 0.79$

α	Algorithm	Avg.	Time (s)	#Best	Dev (%)
1	BRKGA	698.52	720.92	4	9.34
	GRASP + SO	**604.93**	**301.29**	**13**	**0.00**
2	BRKGA	1043.71	615.60	5	7.07
	GRASP + SO	**995.42**	**382.08**	**10**	**3.85**
3	BRKGA	1314.27	530.41	6	5.10
	GRASP + SO	**1211.17**	**414.33**	**7**	**2.80**

Conclusions and future work

- A novel approach based on BRKGA is proposed for solving the α-neighbor p-Center problem.
- A decoder and a local search method have been proposed.
- The experimental phase shows that the algorithm provides promising quality and computational time performance.
  - However, it still needs to be improved

Conclusions and future work

- New decoders and improvement methods can be proposed.
- A parallel version of the algorithm could provide better results in less computing time.
- Test the approach against instances widely used in facility location problems.

A Biased Random Key Genetic Algorithm for solving the α-neighbor p-center problem

Sergio Pérez-Peló (sergio.perez.pelo@urjc.es)

Jesús Sánchez-Oro (jesus.sanchezoro@urjc.es)

Abraham Duarte (abraham.duarte@urjc.es)

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

AGENCIA ESTATAL DE INVESTIGACIÓN

Universidad Rey Juan Carlos