Large-scale multi-objective optimisation for sustainable waste management using Evolutionary Algorithms

A meta-heuristic approach to sustainable waste management

Felix Boelter

Supervisor: Prof. Luca Gambardella

Co-Supervisor: Umberto Jr. Mele

12 September 2022

The situation

Ghazipur Landfill, New Delhi

Source: Money Sharma/AFP/Getty Images

Source: Mohd Zakir/Hindustan Times/Getty Images

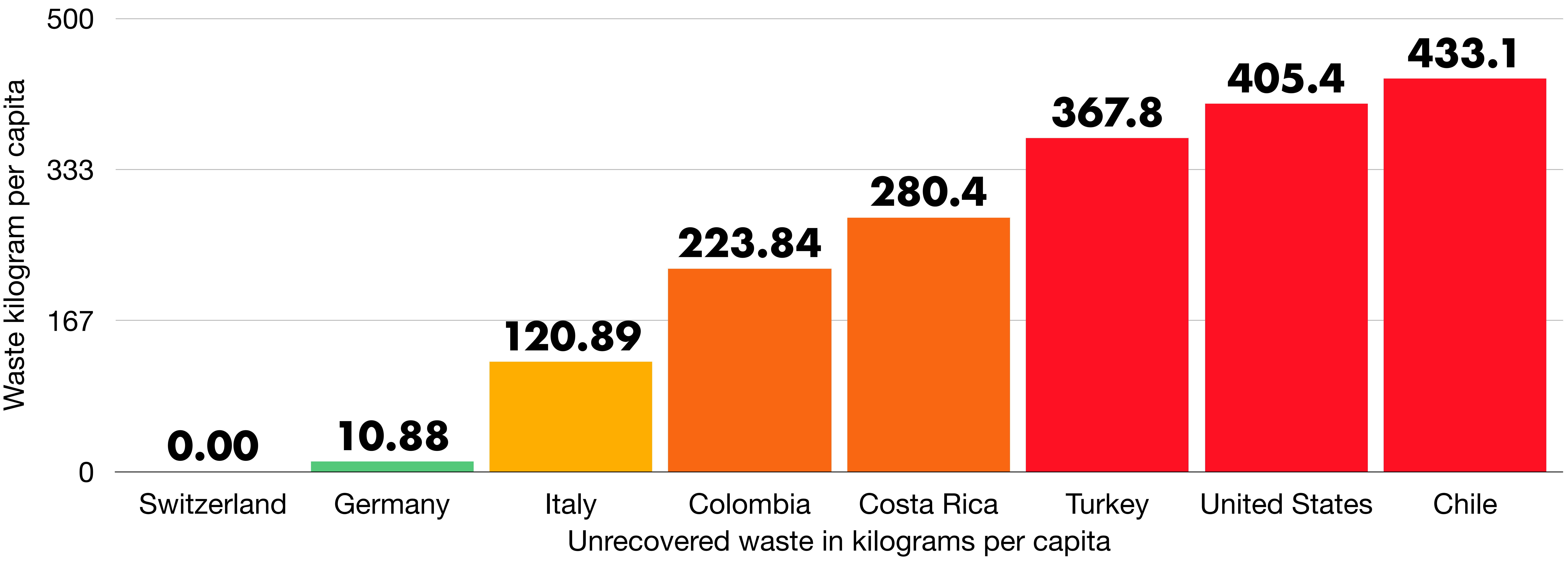
Generation of municipal waste per capita

Source: "OECD (2019), "Waste: Municipal waste", OECD Environment Statistics (database)."

What do the countries do with the waste?

Source: "OECD (2019), "Waste: Municipal waste", OECD Environment Statistics (database)."

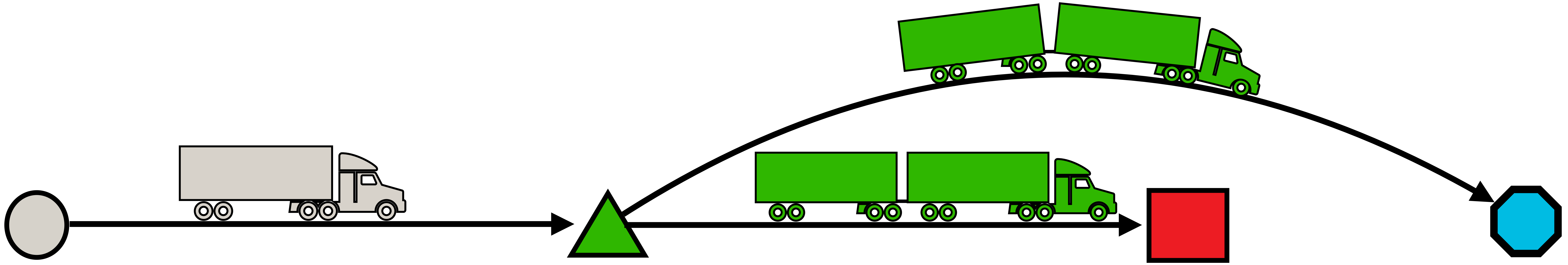
Municipal unrecovered waste per capita



Source: "OECD (2019), "Waste: Municipal waste", OECD Environment Statistics (database)."

How do we currently solve this problem?

Waste management flow chart



Collection centre $i \in I$

Sorting facility $j \in J$

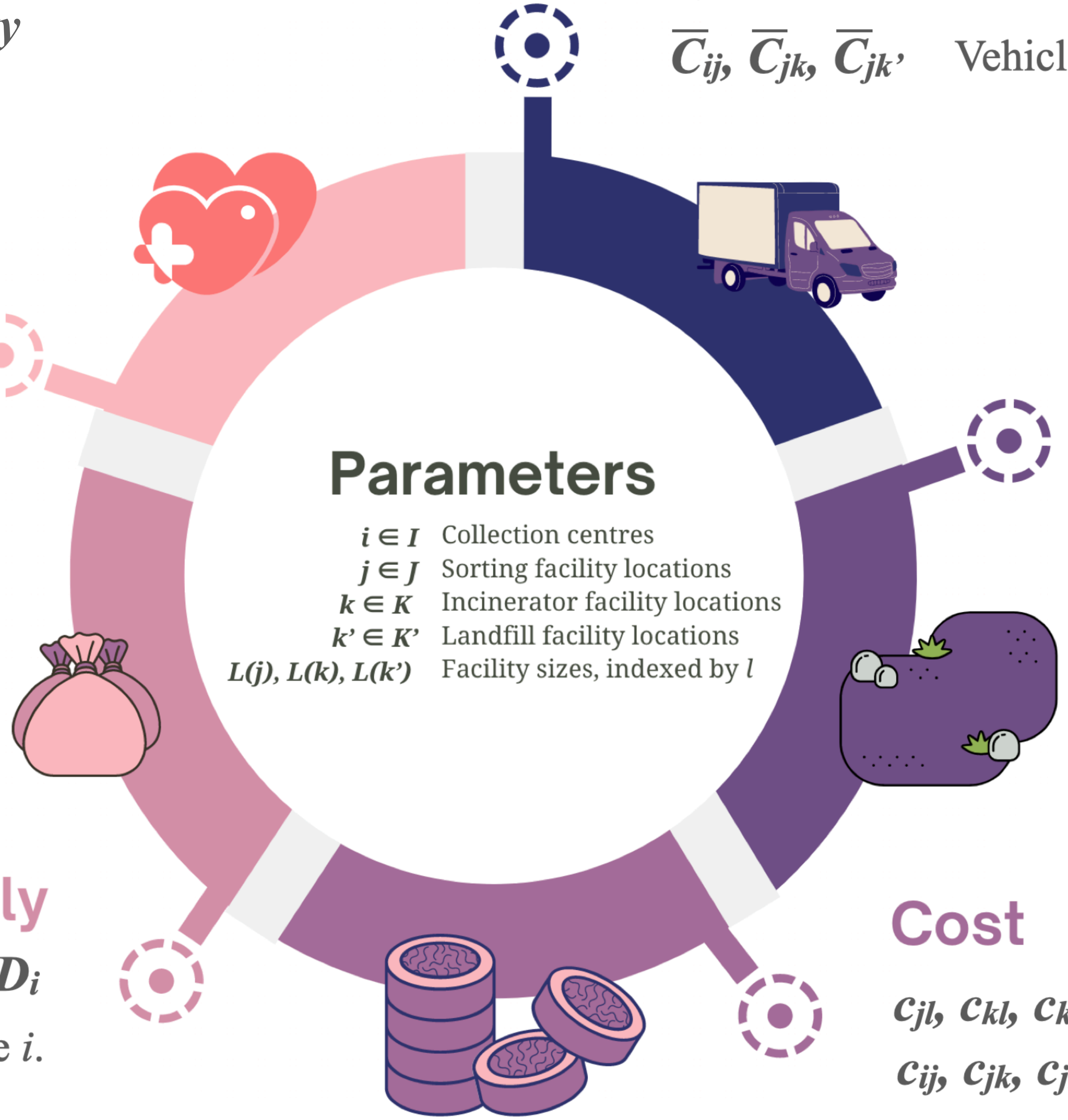
Incinerator facility $k \in K$

Landfill facility $k' \in K'$

Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

MILP Model Parameters

DALY = Years of life lost due to early death + years lost due to disease or disability



Capacity

$C_{jl}, C_{kl}, C_{k'l}$ Storage capacities of facilities of size l.

$\overline{C}_{ij}, \overline{C}_{jk}, \overline{C}_{jk'}$ Vehicle capacities on a single trip for all edges.

Health impact

$p_{ij}, p_{jk}, p_{jk'}$

Number of people living near edges.

$p_{jl}, p_{kl}, p_{k'l}$

Number of people living near facilities of size l.

$d_{ij}, d_{jk}, d_{jk'}$

DALYs per person for all edges.

$d_{jl}, d_{kl}, d_{k'l}$

DALYs per person for all facilities of size l.

Land usage

$s_{jl}, s_{kl}, s_{k'l}$

Land-use stress ratios for facilities of size l.

Waste Supply

D_i

Amount of solid waste at collection centre i.

Cost

$c_{jl}, c_{kl}, c_{k'l}$ Opening costs for facilities of size l.

$c_{ij}, c_{jk}, c_{jk'}$ Transportation costs for all edges.

$o_j, o_k, o_{k'}$ Operation costs to manage the flow of waste.

Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Decision Variables

Binary, y

The location variables equal 1 when a facility is open of size l.

Integer, x

The number of vehicle trips for all edges.

Continuous, f

The amount of solid waste transported for all edges.

Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Minimisation Objective Functions

- Cost objective function
- Land-use objective function
- Health impact objective function

Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

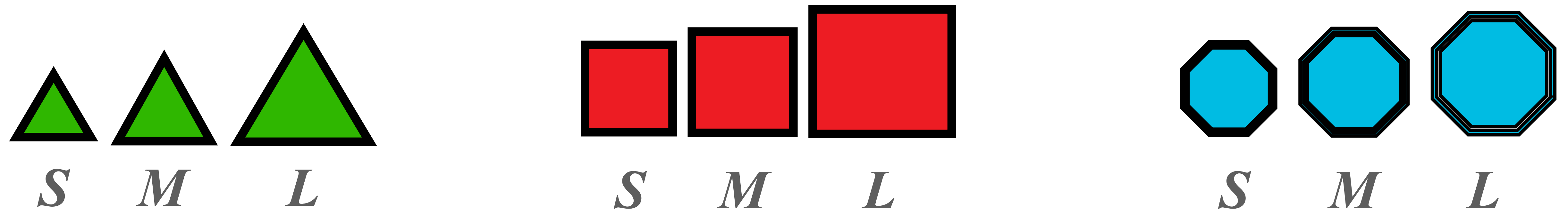
Minimisation Objective Functions

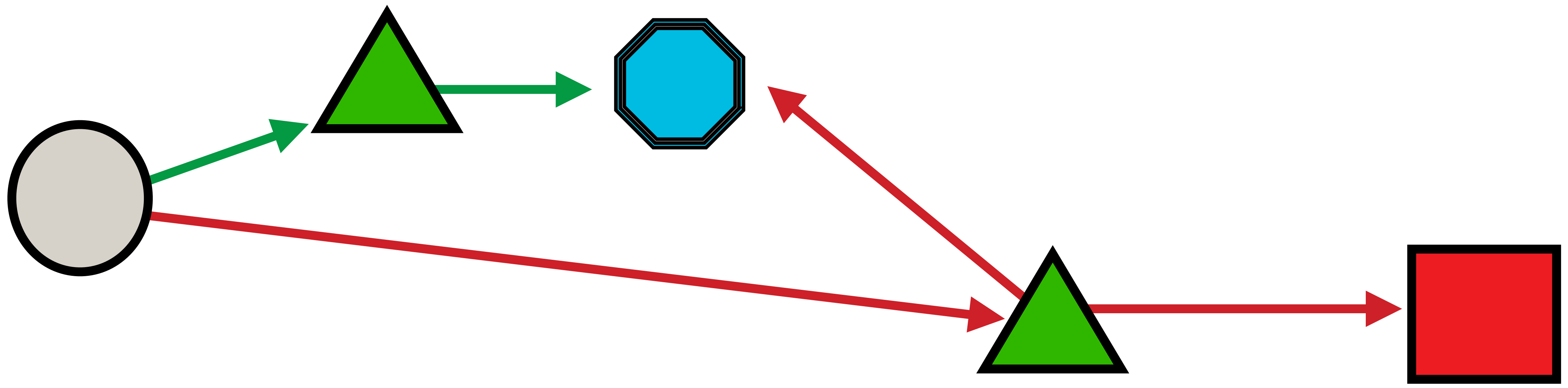
- Cost objective function
- Land-use objective function
- Health impact objective function

Collection centre *Sorting facility* *Incinerator facility* *Landfill facility*

$$F_c = \sum_{j \in J} \sum_{l \in L(j)} c_{jl} y_{jl} + \sum_{k \in K} \sum_{l \in L(k)} c_{kl} y_{kl} + \sum_{k' \in K'} \sum_{l \in L(k')} c_{k'l} y_{k'l}$$

$$+ \sum_{i \in I} \sum_{j \in J} (t_{ij} + o_j) x_{ij} + \sum_{j \in J} \sum_{k \in K} (t_{jk} + o_k) x_{jk} + \sum_{j \in J} \sum_{k' \in K'} (t_{jk'} + o_{k'}) x_{jk'}$$



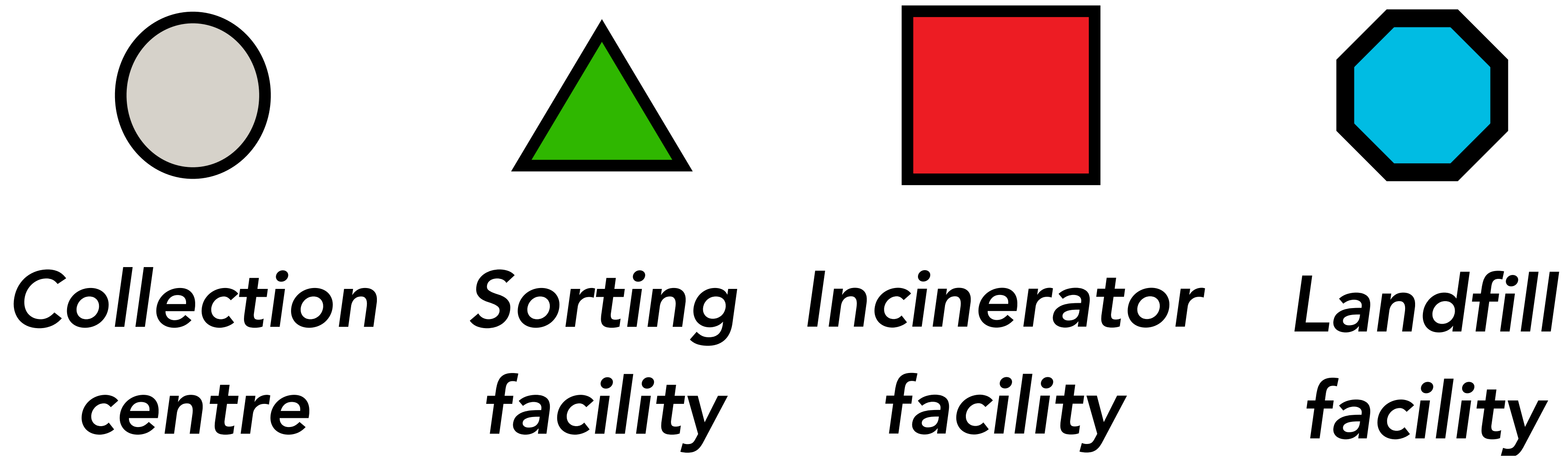


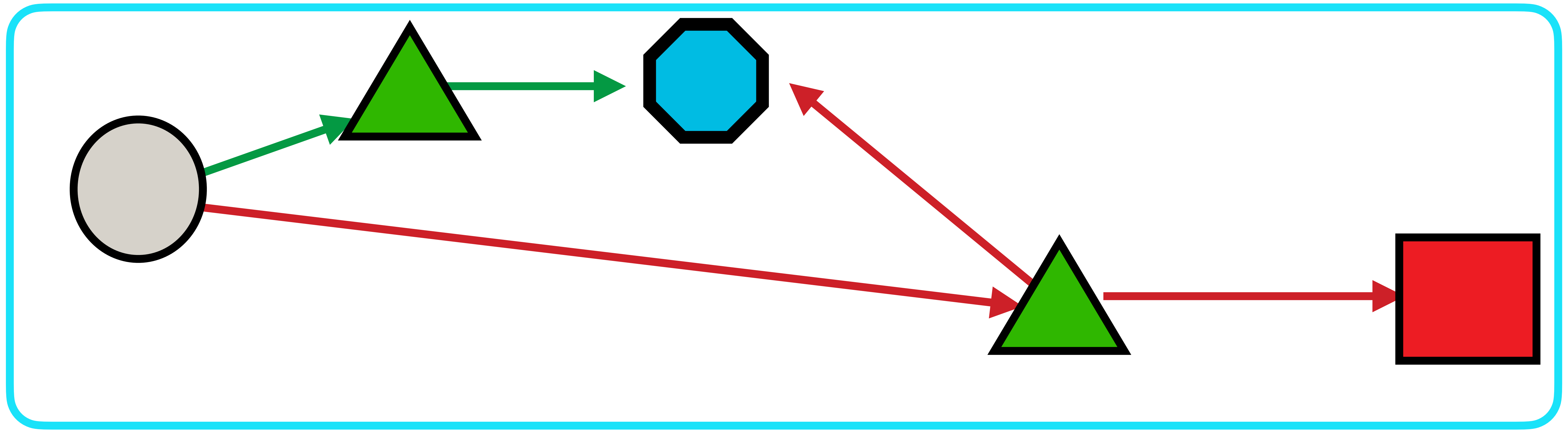
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

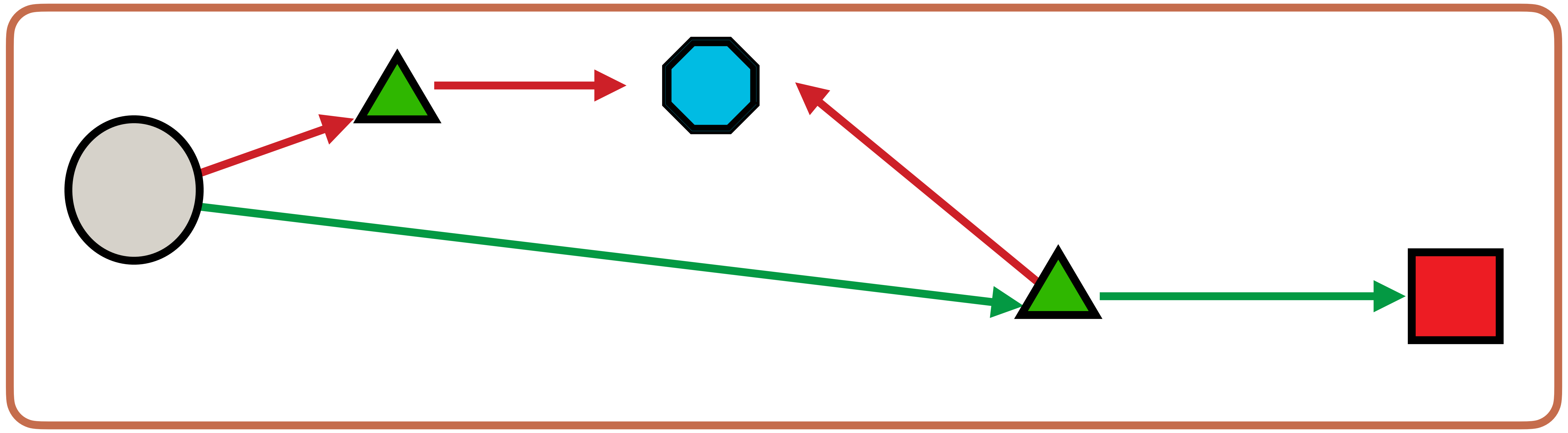
Minimisation Objective Functions

$$F_u = \boxed{\sum_{j \in J} \sum_{l \in L(j)} s_{jl} y_{jl} + \sum_{k \in K} \sum_{l \in L(k)} s_{kl} y_{kl} + \sum_{k' \in K'} \sum_{l \in L(k')} s_{k'l} y_{k'l}}$$

- Cost objective function
- **Land-use objective function**
- Health impact objective function





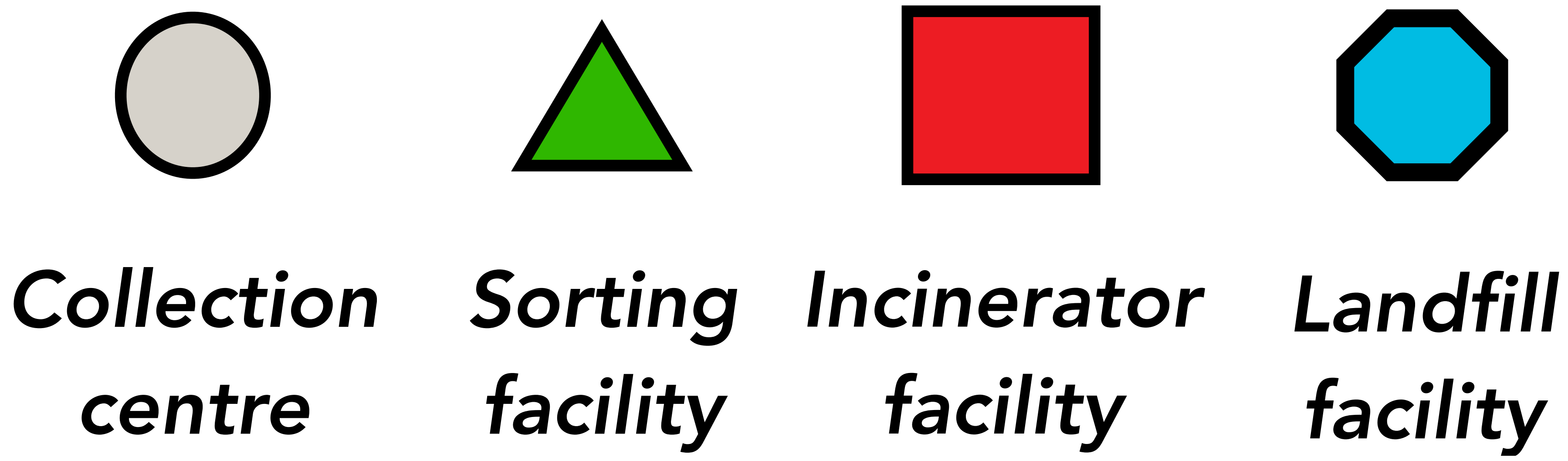


Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Minimisation Objective Functions

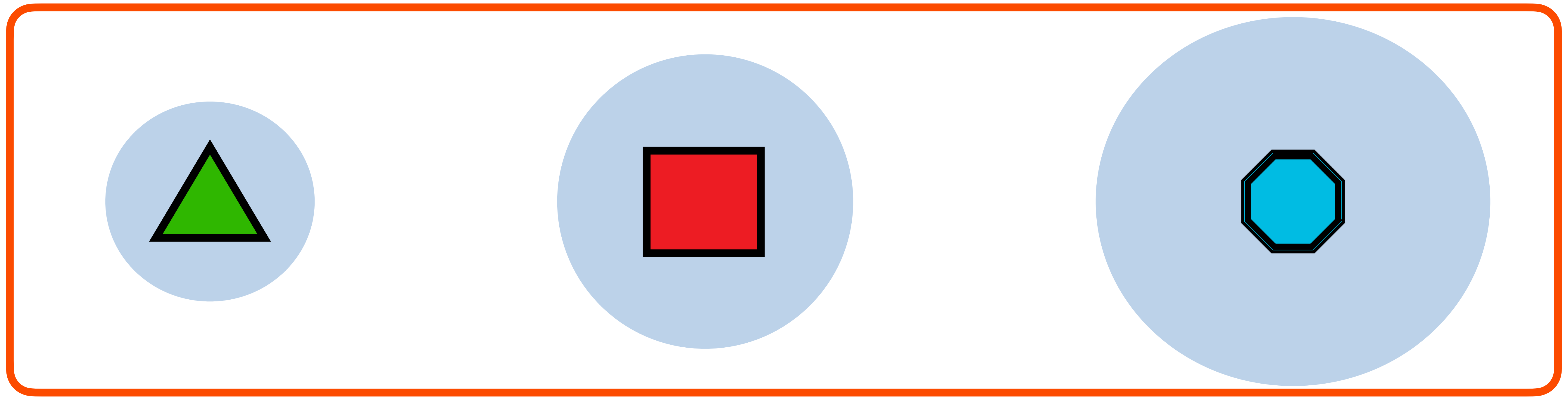
DALY = Years of life lost due to early death + years lost due to disease or disability

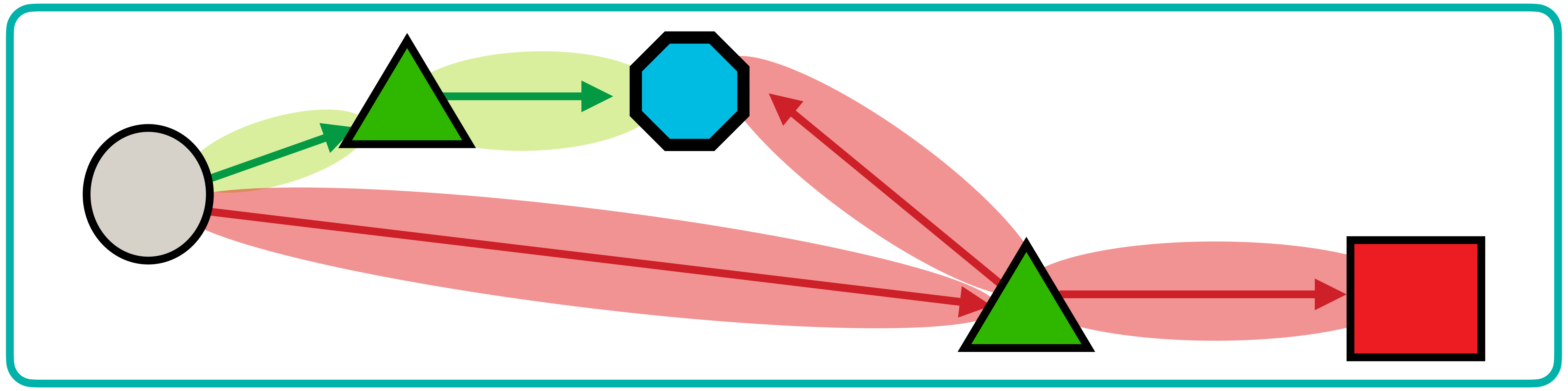
- Cost objective function
- Land-use objective function
- **Health impact objective function**



$$F_h = \sum_{j\in J}\sum_{l\in L(j)} p_{jl}d_{jl}y_{jl} + \sum_{k\in K}\sum_{l\in L(k)} p_{kl}d_{kl}y_{kl} + \sum_{k'\in K'}\sum_{l\in L(k')} p_{k'l}d_{k'l}y_{k'l}$$

$$+ \sum_{i\in I}\sum_{j\in J} p_{ij}d_{ij}x_{ij} + \sum_{j\in J}\sum_{k\in K} p_{jk}d_{jk}x_{jk} + \sum_{j\in J}\sum_{k'\in K'} p_{jk'}d_{jk'}x_{jk'}$$





Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Constraints

- Flow constraints
- Facility capacity constraints
- Vehicle capacity constraints
- One facility size constraints

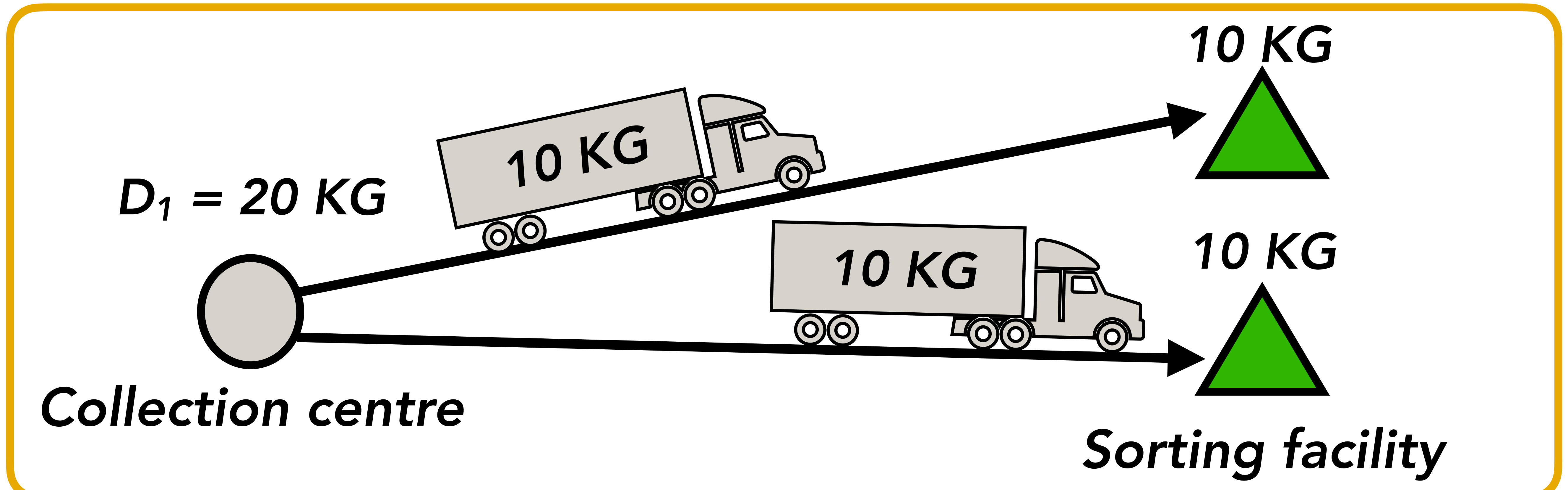
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

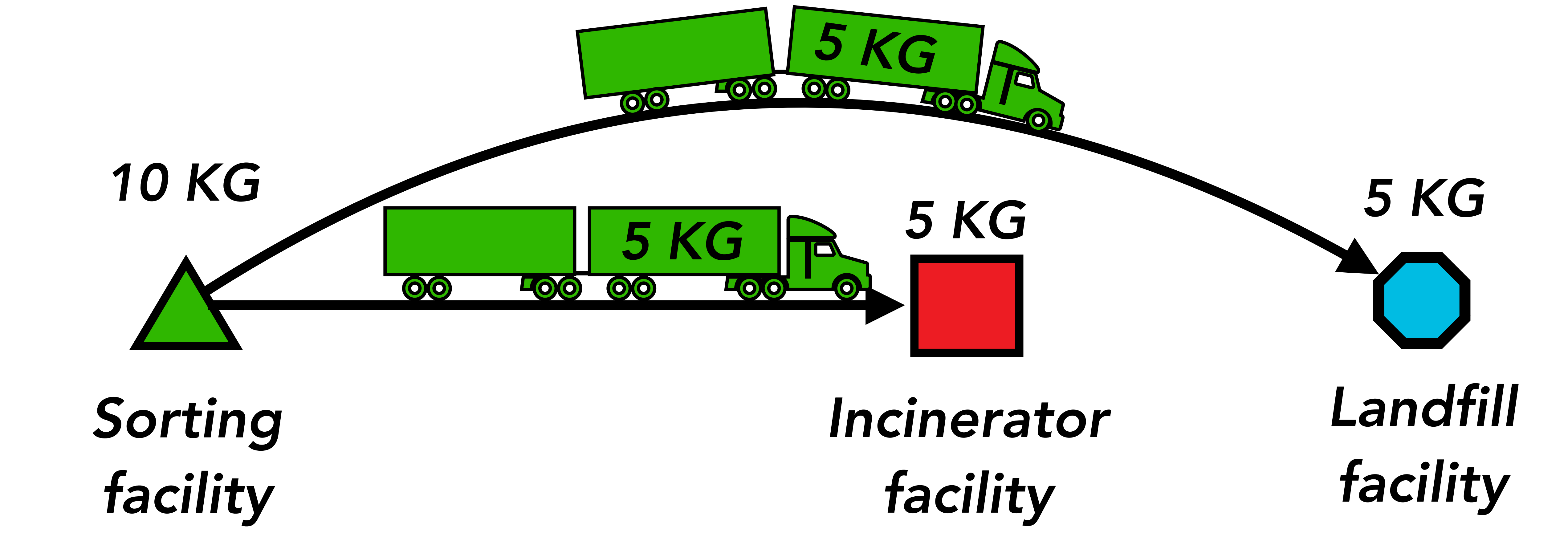
Constraints

$$\sum_{j \in J} f_{ij} = D_i, \quad \forall i \in I$$

$$\sum_{i \in I} f_{ij} = \sum_{k \in K \cup K'} f_{jk}, \quad \forall j \in J$$

- Flow constraints
- Facility capacity constraints
- Vehicle capacity constraints
- One facility size constraints





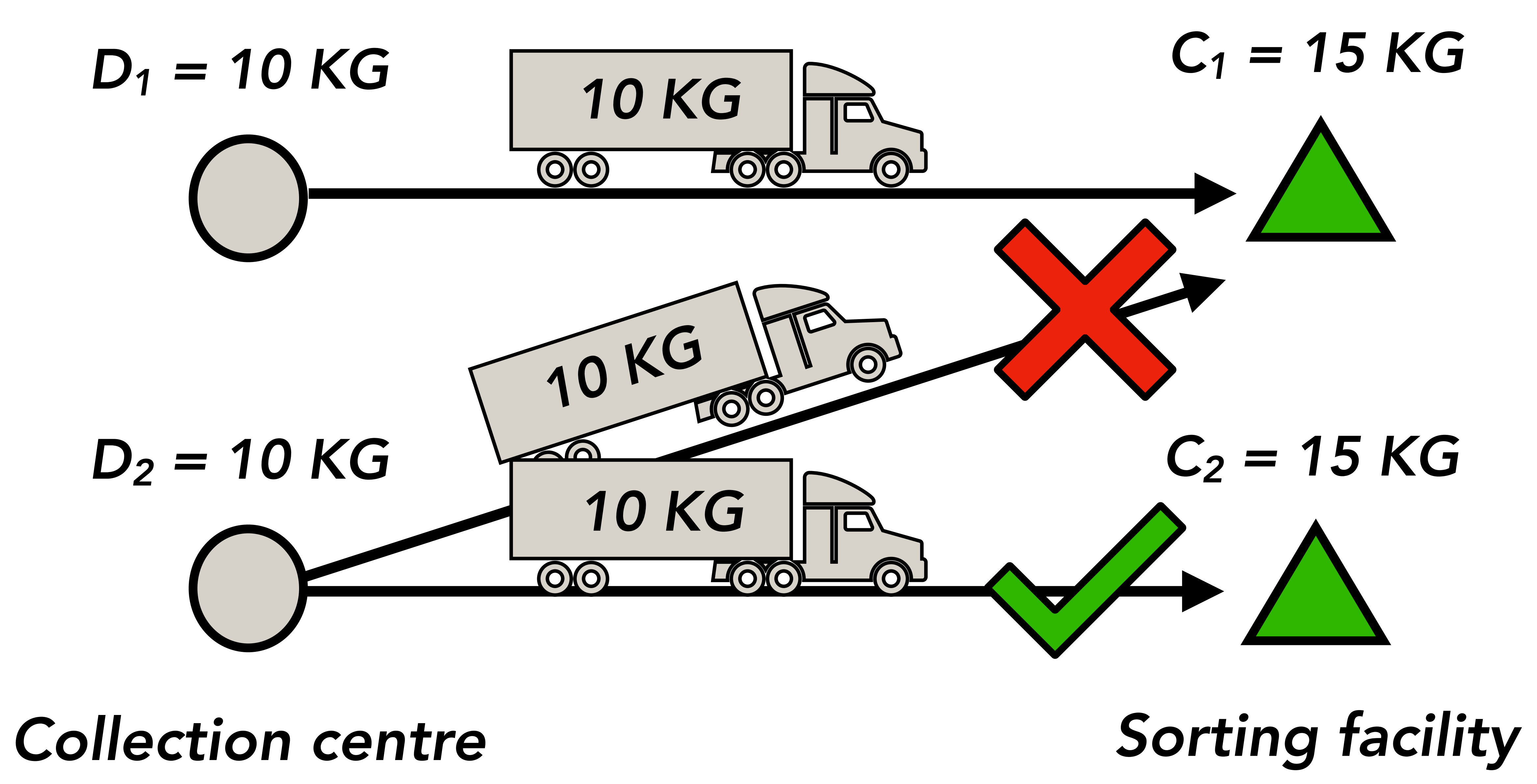
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Constraints

- Flow constraints
- **Facility capacity constraints**
- Vehicle capacity constraints
- One facility size constraints

$$\sum_{j\in J} f_{jk} \leq \sum_{l\in L(k)} C_{kl}y_{kl}, \quad \forall k \in K \qquad \sum_{j\in J} f_{jk'} \leq \sum_{l\in L(k')} C_{k'l'}y_{k'l'}, \quad \forall k' \in K'$$

$$\sum_{i\in I} f_{ij} \leq \sum_{l\in L(j)} C_{jl}y_{jl}, \quad \forall j \in J$$



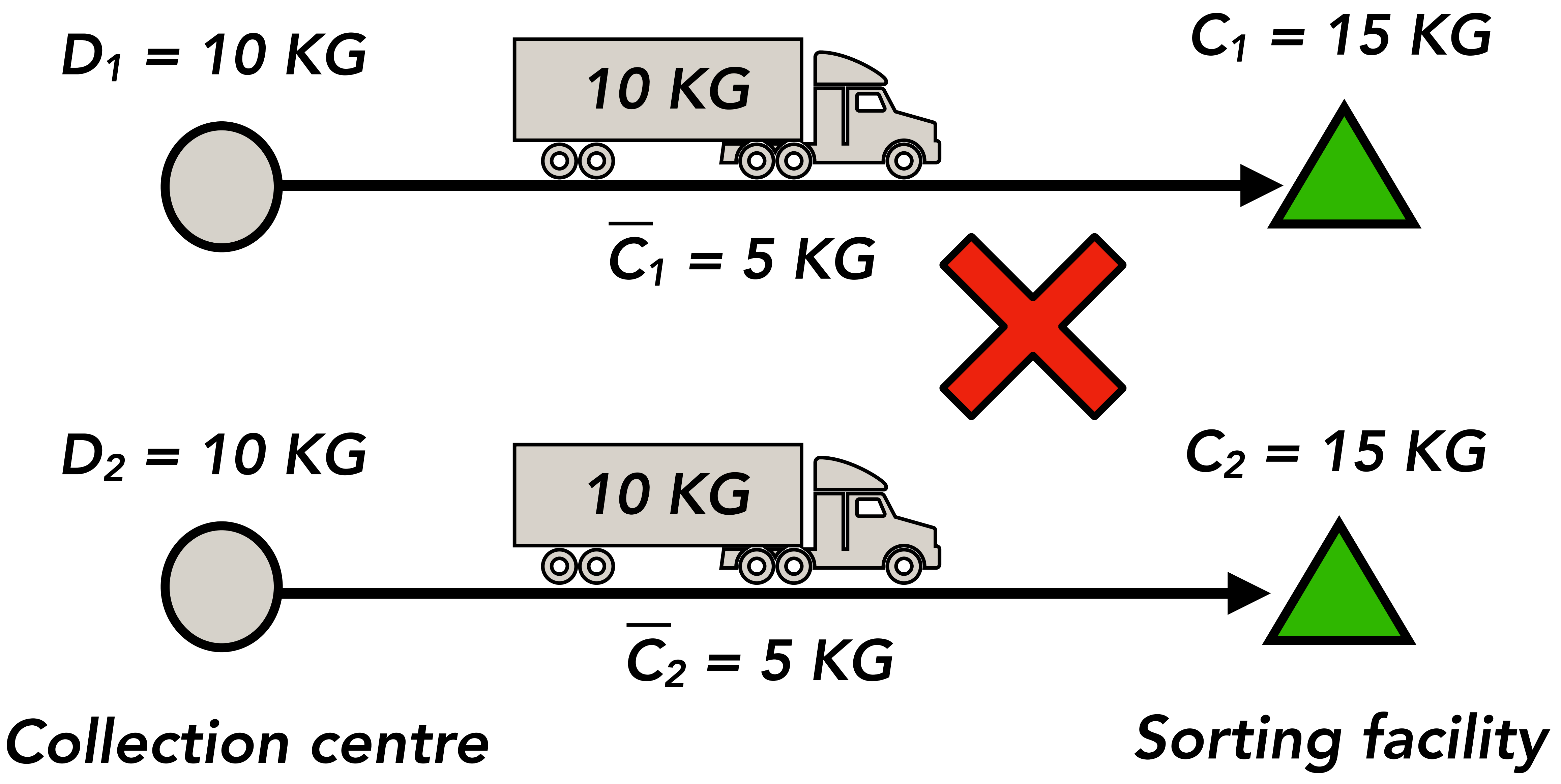
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Constraints

- Flow constraints
- Facility capacity constraints
- **Vehicle capacity constraints**
- One facility size constraints

$$f_{jk} \leq \overline{C}_{jk} x_{jk}, \quad \forall j \in J, k \in K \qquad f_{jk'} \leq \overline{C}_{jk'} x_{jk'}, \quad \forall j \in J, k' \in K'$$

$$f_{ij} \leq \overline{C}_{ij} x_{ij}, \quad \forall i \in I, j \in J$$



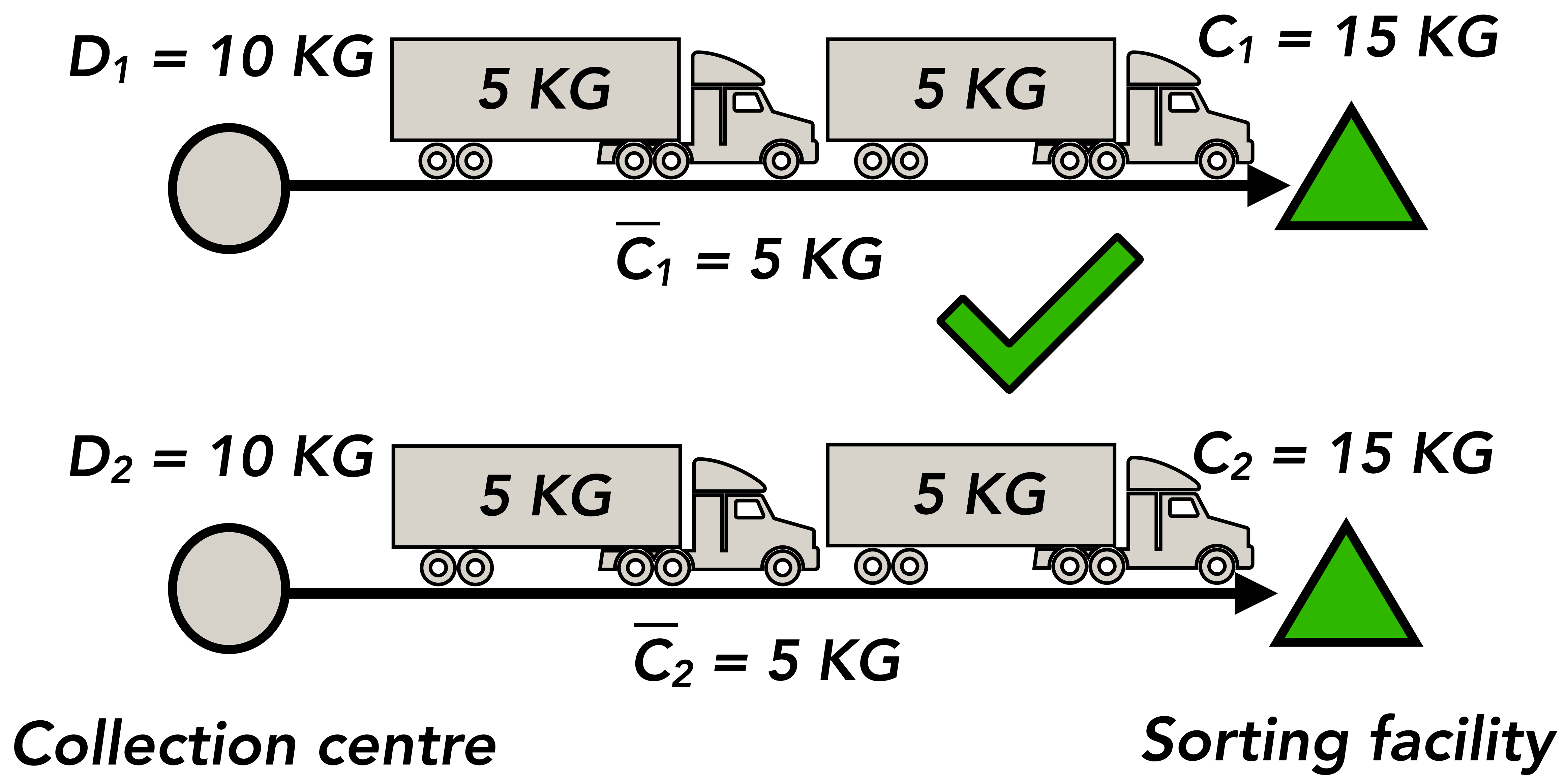
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Constraints

- Flow constraints
- Facility capacity constraints
- **Vehicle capacity constraints**
- One facility size constraints

$$f_{jk} \leq \overline{C}_{jk} x_{jk}, \quad \forall j \in J, k \in K \qquad f_{jk'} \leq \overline{C}_{jk'} x_{jk'}, \quad \forall j \in J, k' \in K'$$

$$f_{ij} \leq \overline{C}_{ij} x_{ij}, \quad \forall i \in I, j \in J$$



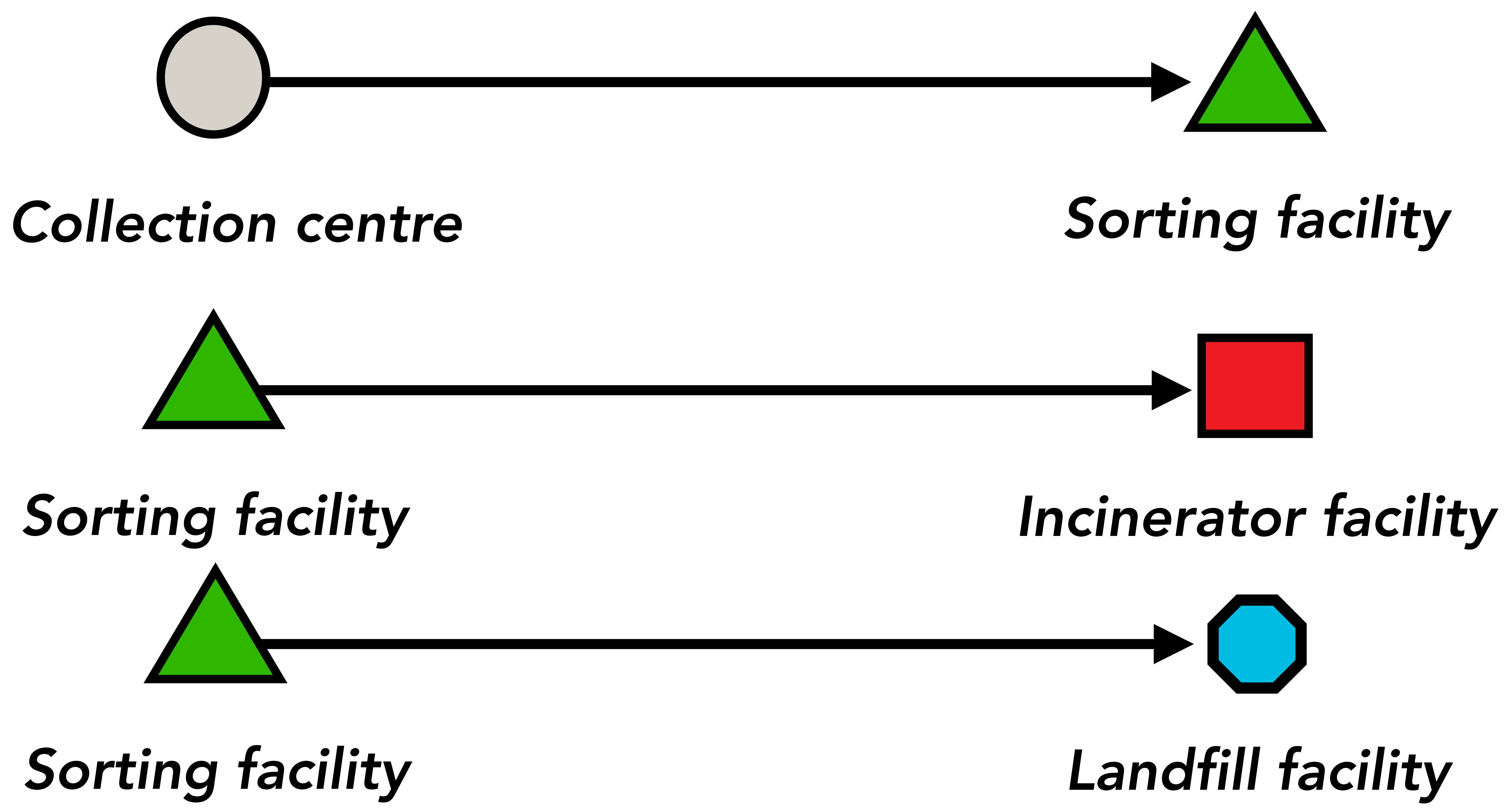
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Constraints

- Flow constraints
- Facility capacity constraints
- Vehicle capacity constraints
- **One facility size constraints**

$$\sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J$$



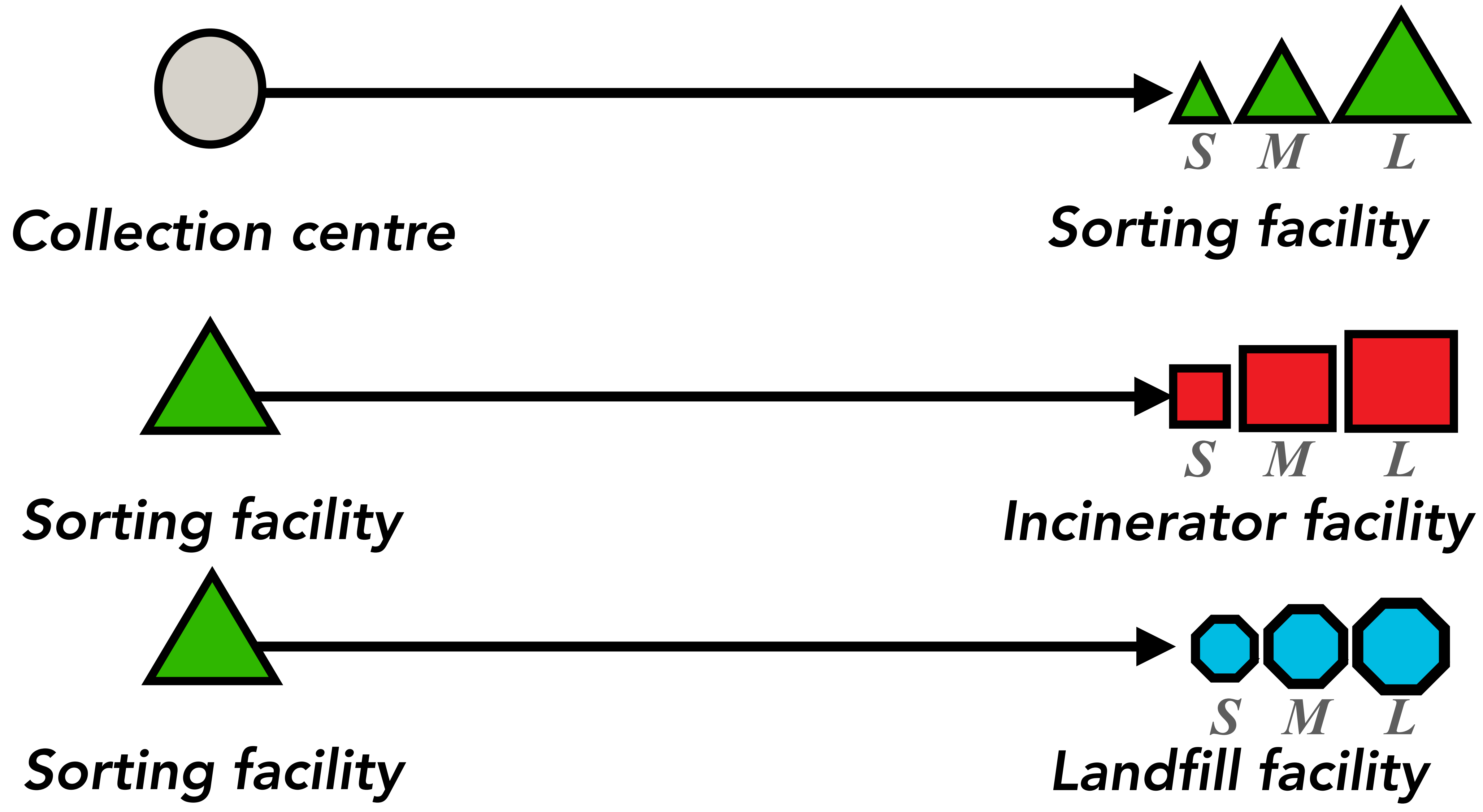
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Constraints

- Flow constraints
- Facility capacity constraints
- Vehicle capacity constraints
- **One facility size constraints**

$$\sum_{l \in L(k)} y_{kl} \le 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \le 1, \quad \forall k' \in K'$$

$$\sum_{l \in L(j)} y_{jl} \le 1, \quad \forall j \in J$$



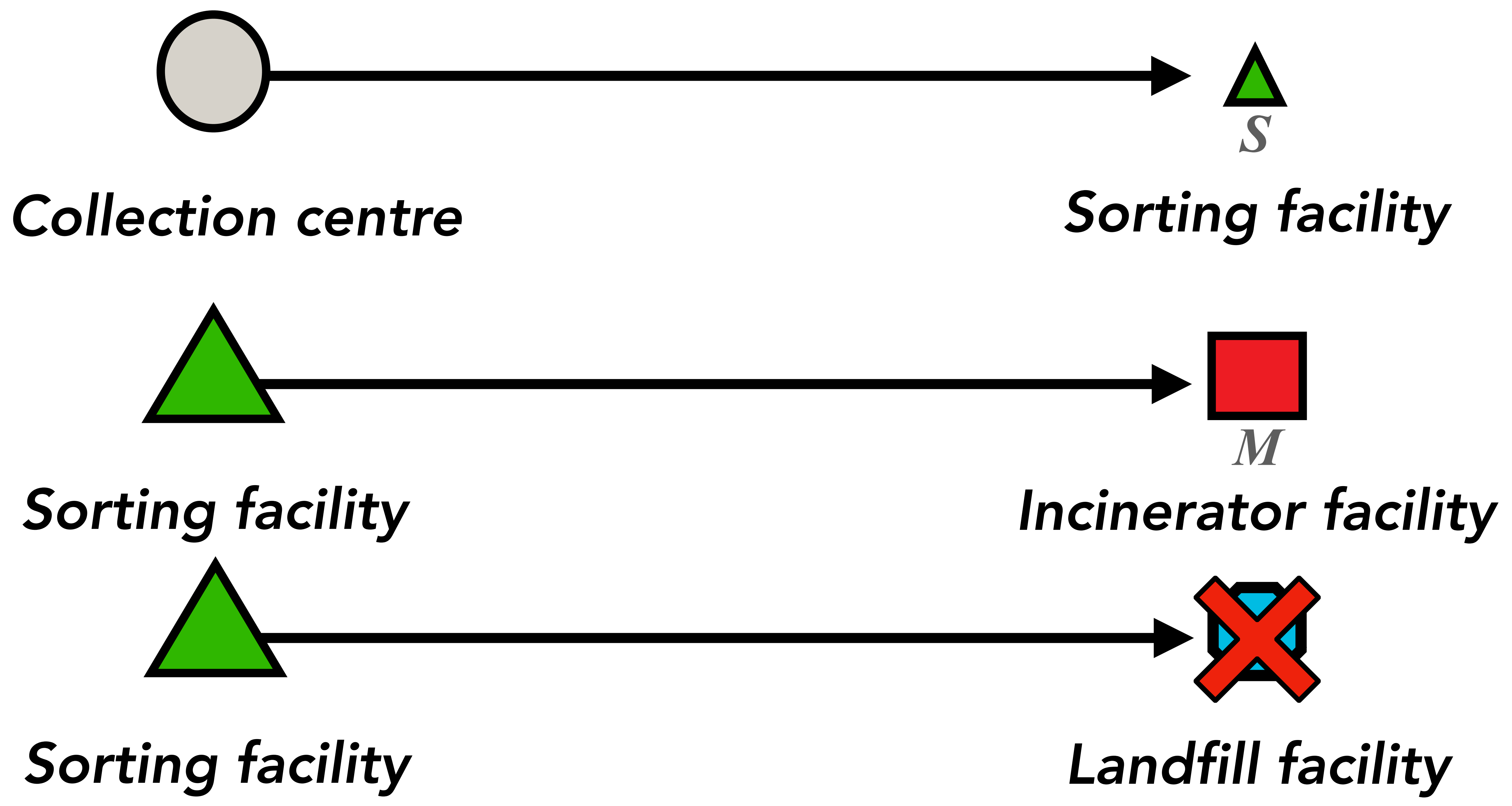
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Constraints

- Flow constraints
- Facility capacity constraints
- Vehicle capacity constraints
- **One facility size constraints**

$$\sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J$$

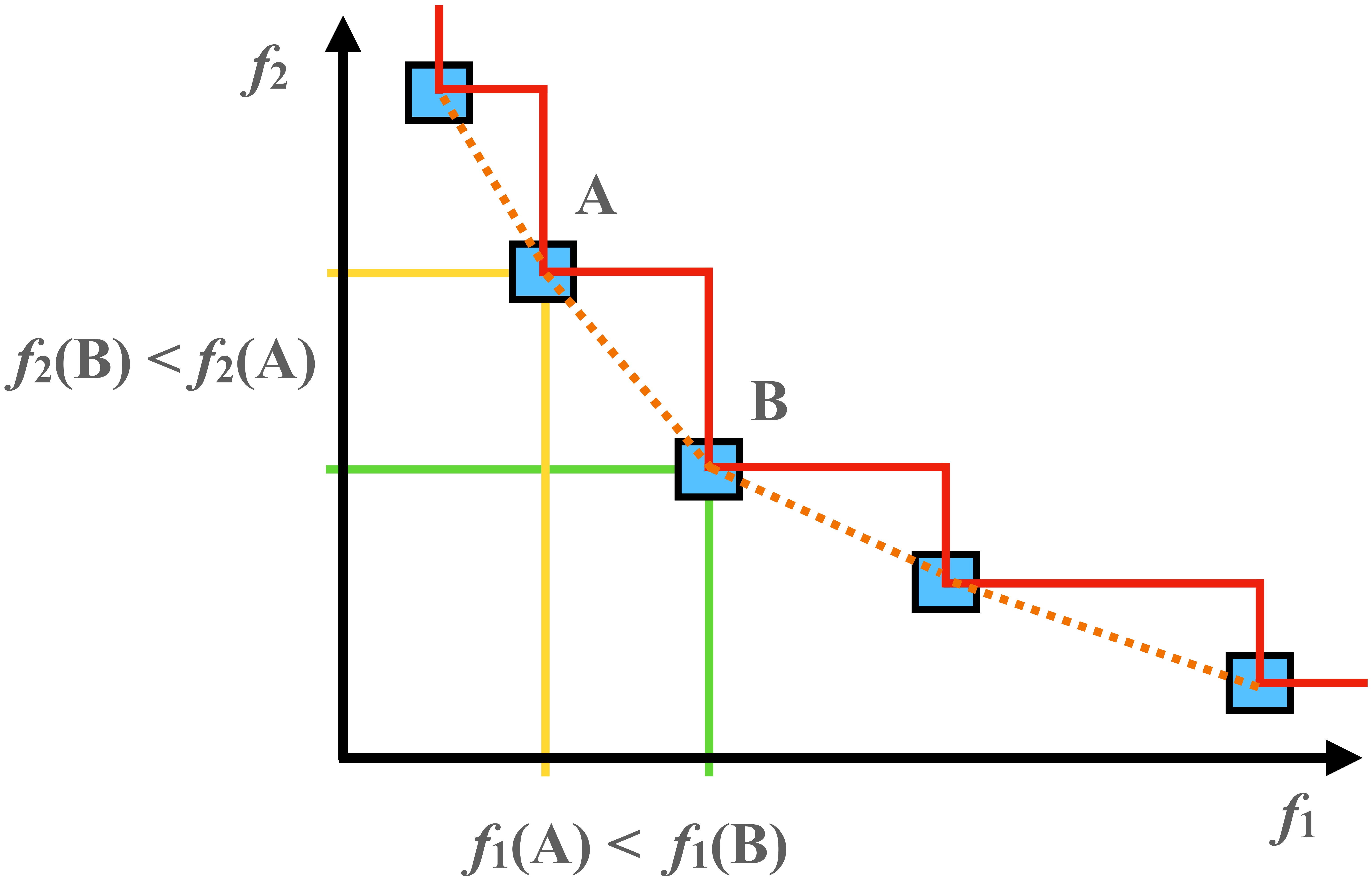


Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Multi-objective formulation

Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Multi-objective problem



Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Multi-objective formulation

Binary, y

The location variables equal 1 when a facility is open of size l.

Integer, x

The number of vehicle trips for all edges.

Continuous, f

The amount of solid waste transported for all edges.

Continuous, z

Variable to minimise for the multi-objective formulation.

Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Multi-objective formulation

Minimise single-objectives

Optimise each objective function subject to constraints and obtain each ideal solution.

Calculate deviations

$$\sigma_c = \frac{F_c - F_c^{ideal}}{F_c^{max} - F_c^{ideal}}$$

$$\sigma_u = \frac{F_u - F_u^{ideal}}{F_u^{max} - F_u^{ideal}} \qquad \sigma_h = \frac{F_h - F_h^{ideal}}{F_h^{max} - F_h^{ideal}}$$

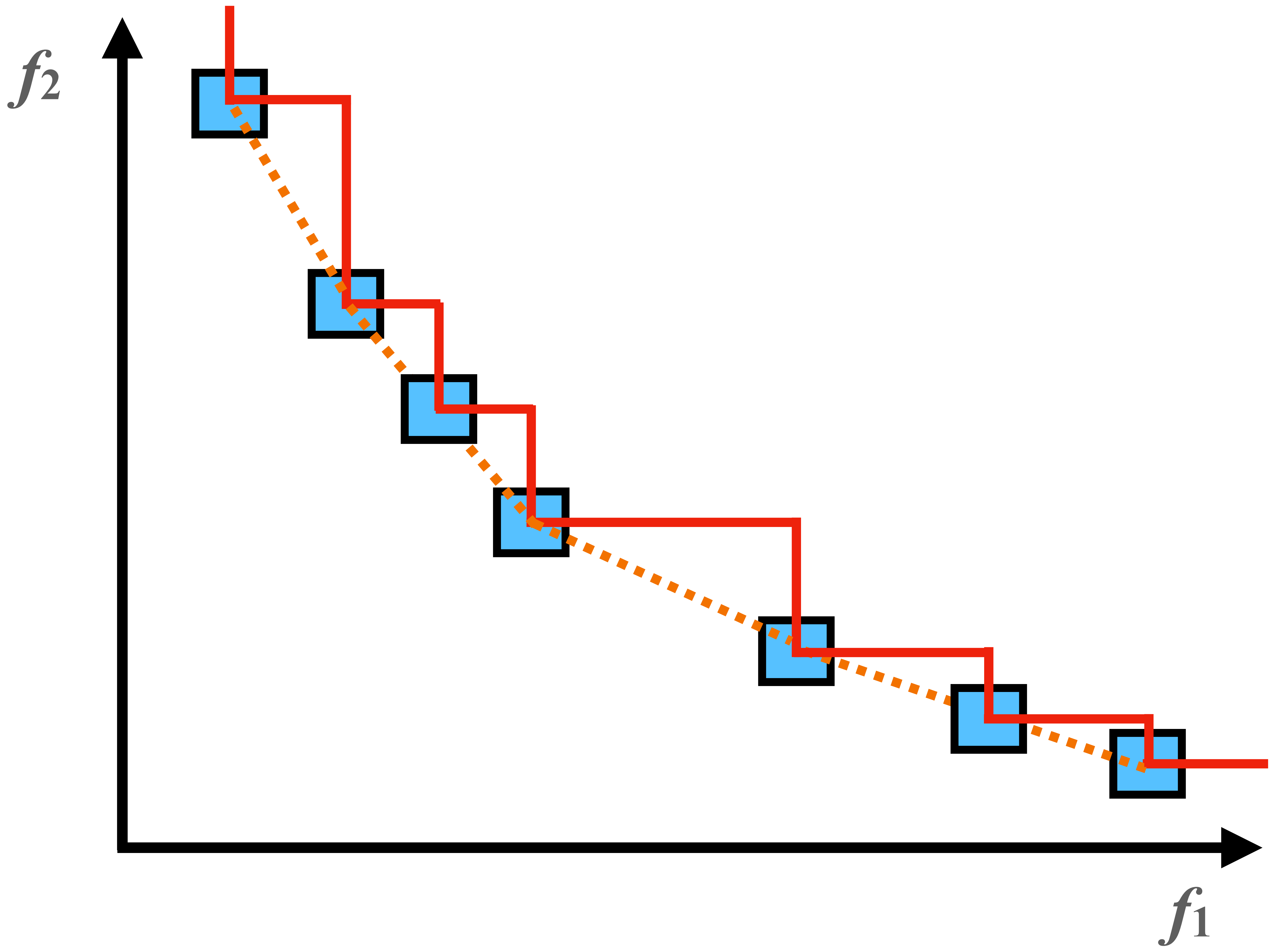
Minimise multi-objective in a single-objective manner

$$\begin{aligned} min \quad & z \\ s.t. \quad & \sigma_c \leq z \\ & \sigma_u \leq z \\ & \sigma_h \leq z \\ & Other\ constraints \end{aligned}$$

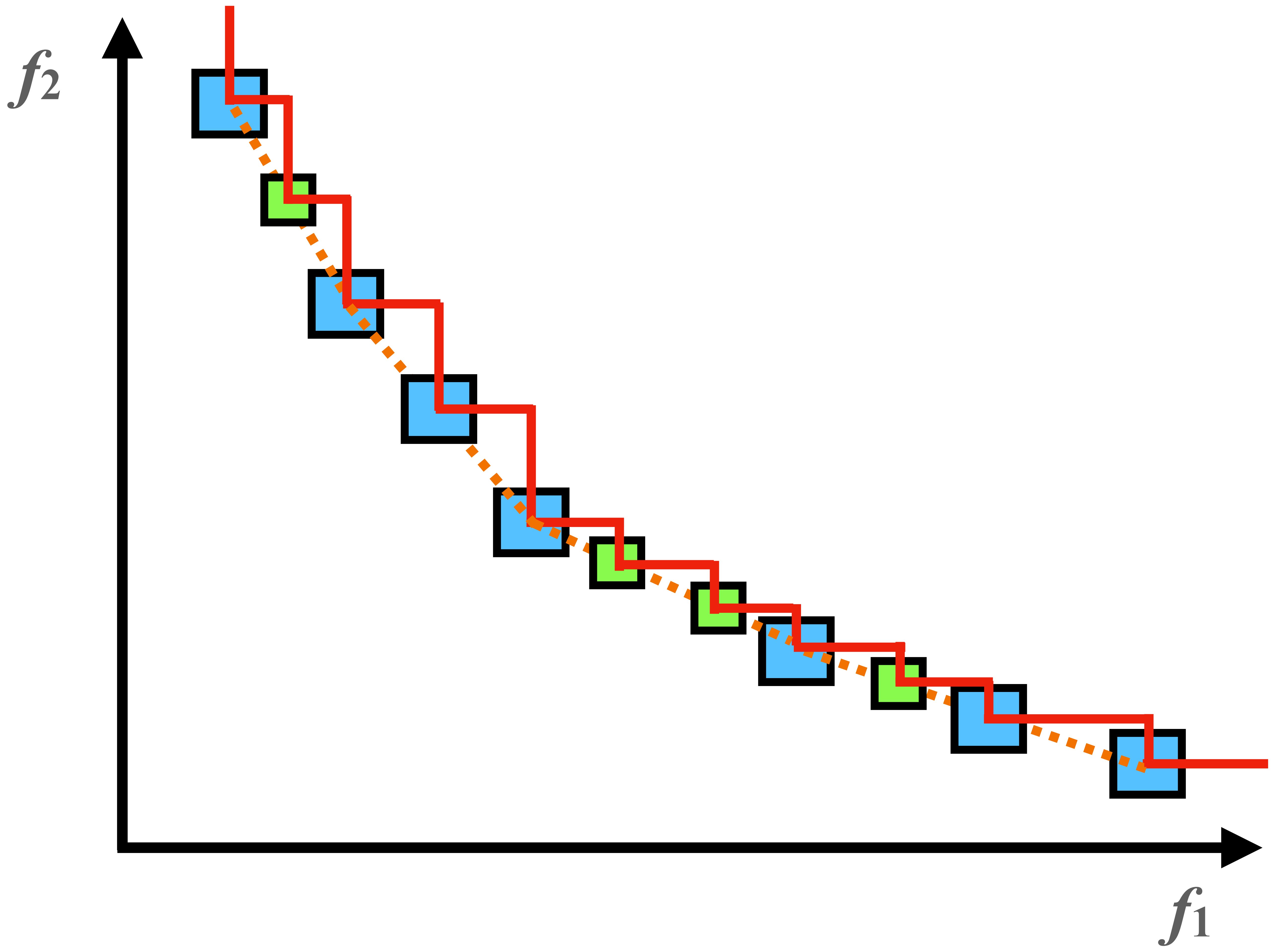
Source: Olapiriyakul et al. Multi-objective Optimization Model for Sustainable Waste Management Network Design. May 2019.

Limitations of the multi-objective approach

Limitations of the multi-objective approach



Limitations of the multi-objective approach



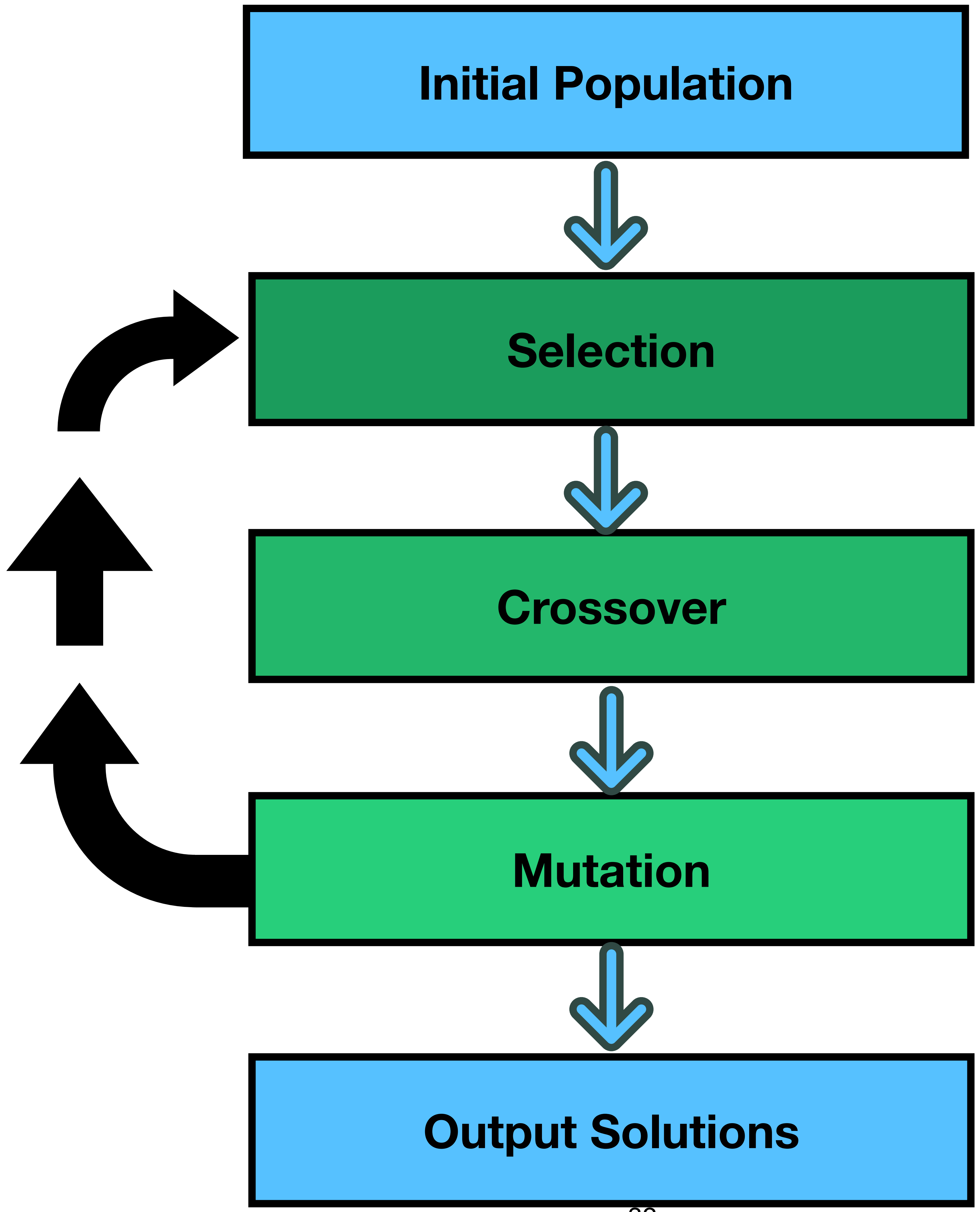
Limitations of the multi-objective approach

Evolutionary Algorithms

Evolutionary Algorithms : Claims

- An improvement in time to solve the multi-objective waste management MILP problem on a larger scale.
- We are still obtaining solutions close to the baseline model.
- We are creating trade-offs between the three objectives by receiving a Pareto-set of diverse solutions.

Evolutionary Algorithms



Evolutionary Algorithms : Crossover for y

	S	M	L	S	M	L
Parent 1:	**0**	**1**	**0**	**0**	**1**	**0**
Parent 2:	**1**	**0**	**0**	**0**	**0**	**1**

Evolutionary Algorithms : Crossover for y

Crossover point

	S	M	L	S	M	L
Parent 1:	0	1	0	0	1	0
Parent 2:	1	0	0	0	0	1

Evolutionary Algorithms : Crossover for y

	S	M	L	S	M	L
Parent 1:	0	1	0	0	1	0
Parent 2:	1	0	0	0	0	1
Offspring 1:	0	0	0	0	0	1
Offspring 2:	1	1	0	0	1	0

Evolutionary Algorithms: Adapted Crossover for y

Offspring 2:

S	M	L	S	M	L
1	1	0	0	1	0

Adapted Offspring 2:

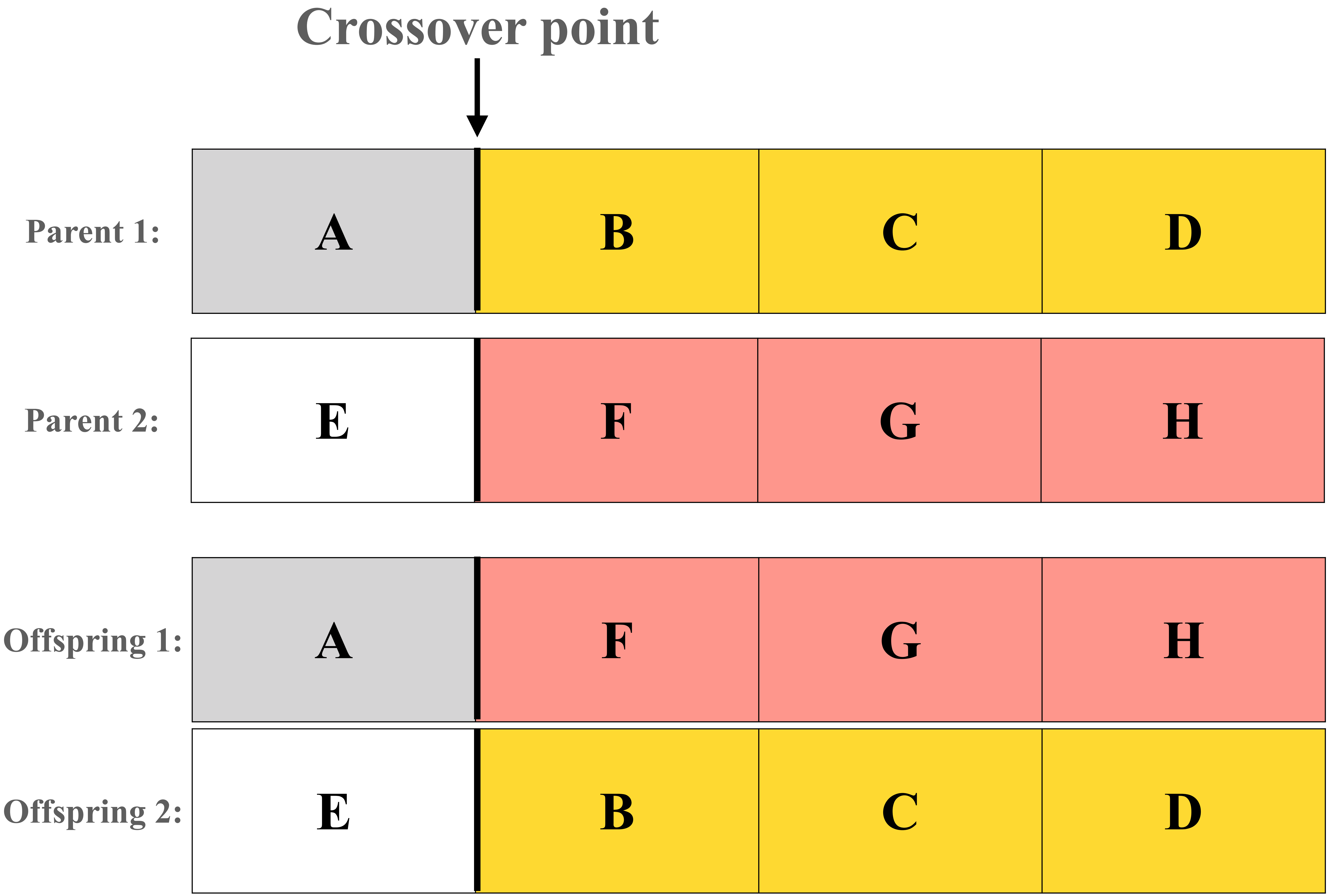
S	M	L	S	M	L
0	1	0	0	1	0

Istituto Dalle Molle di studi sull'intelligenza artificiale
USI/SUPSI IDSIA
Università della Svizzera italiana

Evolutionary Algorithms : Crossover for f

Parent 1:	**A**	**B**	**C**	**D**
Parent 2:	**E**	**F**	**G**	**H**

Evolutionary Algorithms: Crossover for f



Evolutionary Algorithms: Adapted Crossover for f

Normalise using supply

Adapted Offspring 1:	**a**	**f**	**g**	**h**
Adapted Offspring 2:	**e**	**b**	**c**	**d**

Evolutionary Algorithms : Adapted Crossover for f

Divide by sum of offspring

Adapted Offspring 1:	**i**	**j**	**k**	**l**
Adapted Offspring 2:	**m**	**n**	**o**	**p**

Evolutionary Algorithms: Adapted Crossover for f

De-normalise using supply

Adapted Offspring 1:	**I**	**J**	**K**	**L**
Adapted Offspring 2:	**M**	**N**	**O**	**P**

Evolutionary Algorithms: Mutation for y

Individual 1:

S	M	L
0	1	0
0	1	0

Offspring 1:

S	M	L
0	1	1
0	1	0

Evolutionary Algorithms: Mutation for y

	S	M	L
	0	0	1
Adapted Offspring 1:	0	1	0

Evolutionary Algorithms : Mutation for f

Individual 1:	**A**	**B**	**C**	**D**
Offspring 1:	**A**	**E**	**C**	**D**

Evolutionary Algorithms : Mutation for f

Normalise using supply

Offspring 1:

A	E	C	D

→

Adapted Offspring 1:

a	e	c	d

Evolutionary Algorithms : Mutation for f

Divide by sum of offspring

Adapted Offspring 1:

a	e	c	d

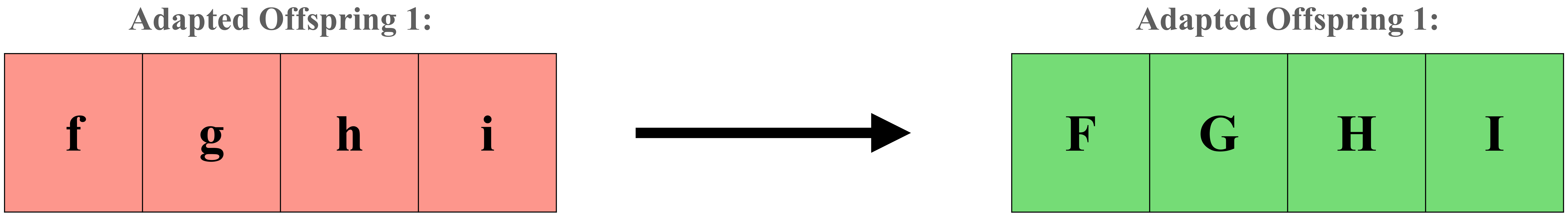
→

Adapted Offspring 1:

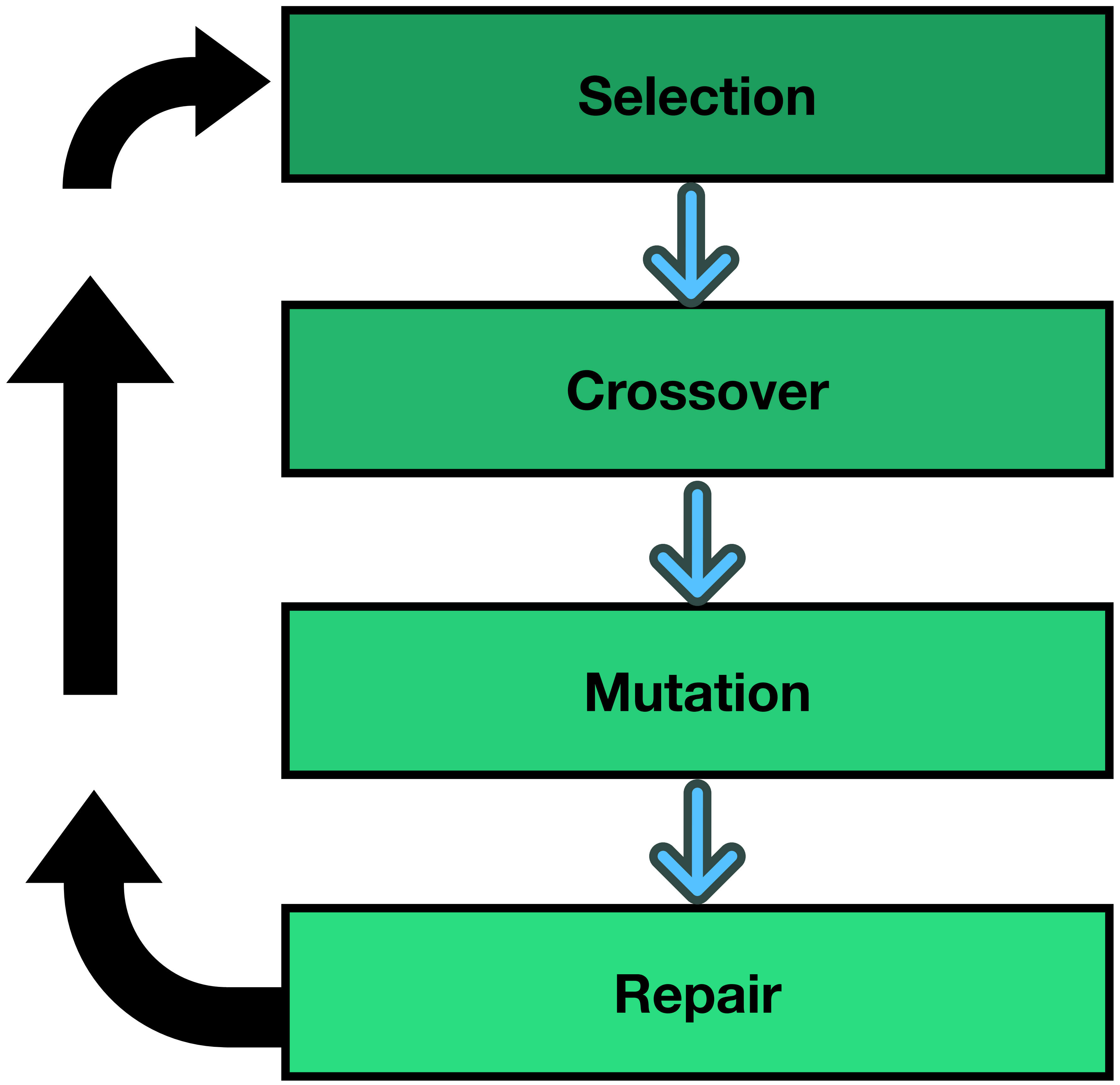
f	g	h	i

Evolutionary Algorithms: Mutation for f

De-normalise using supply

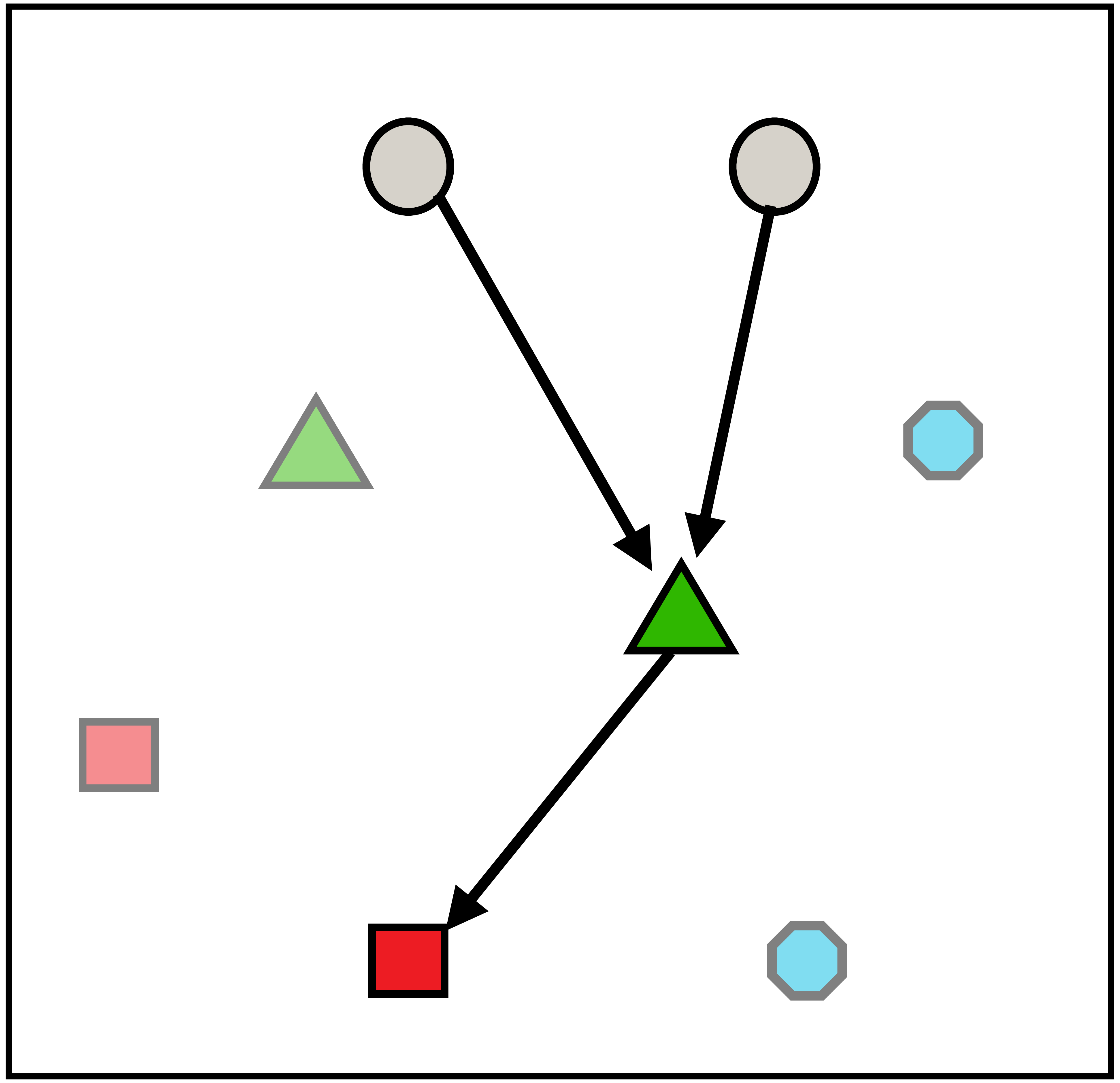


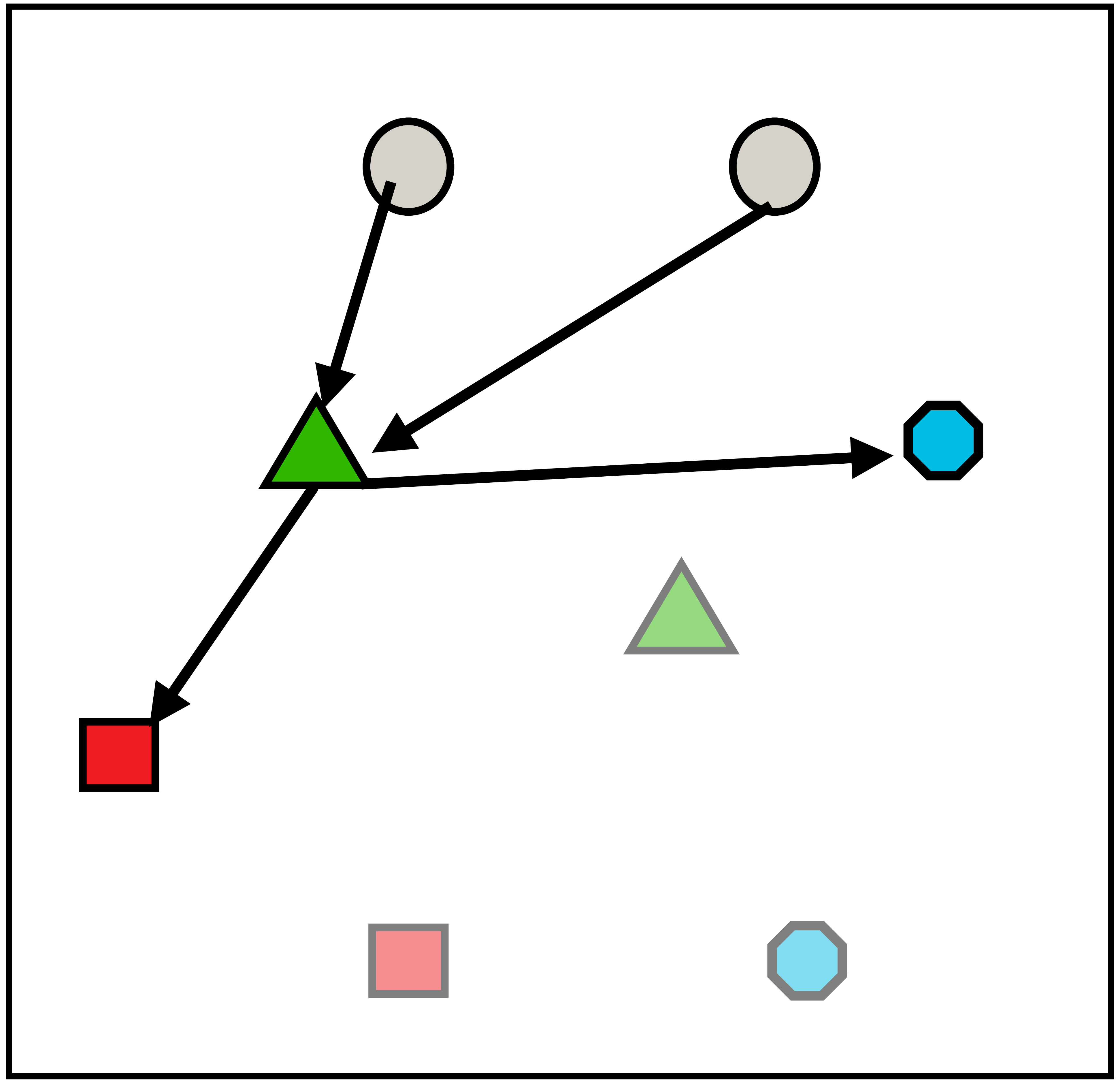
Evolutionary Algorithms : Repair operator



Evolutionary Algorithms : Repair operator

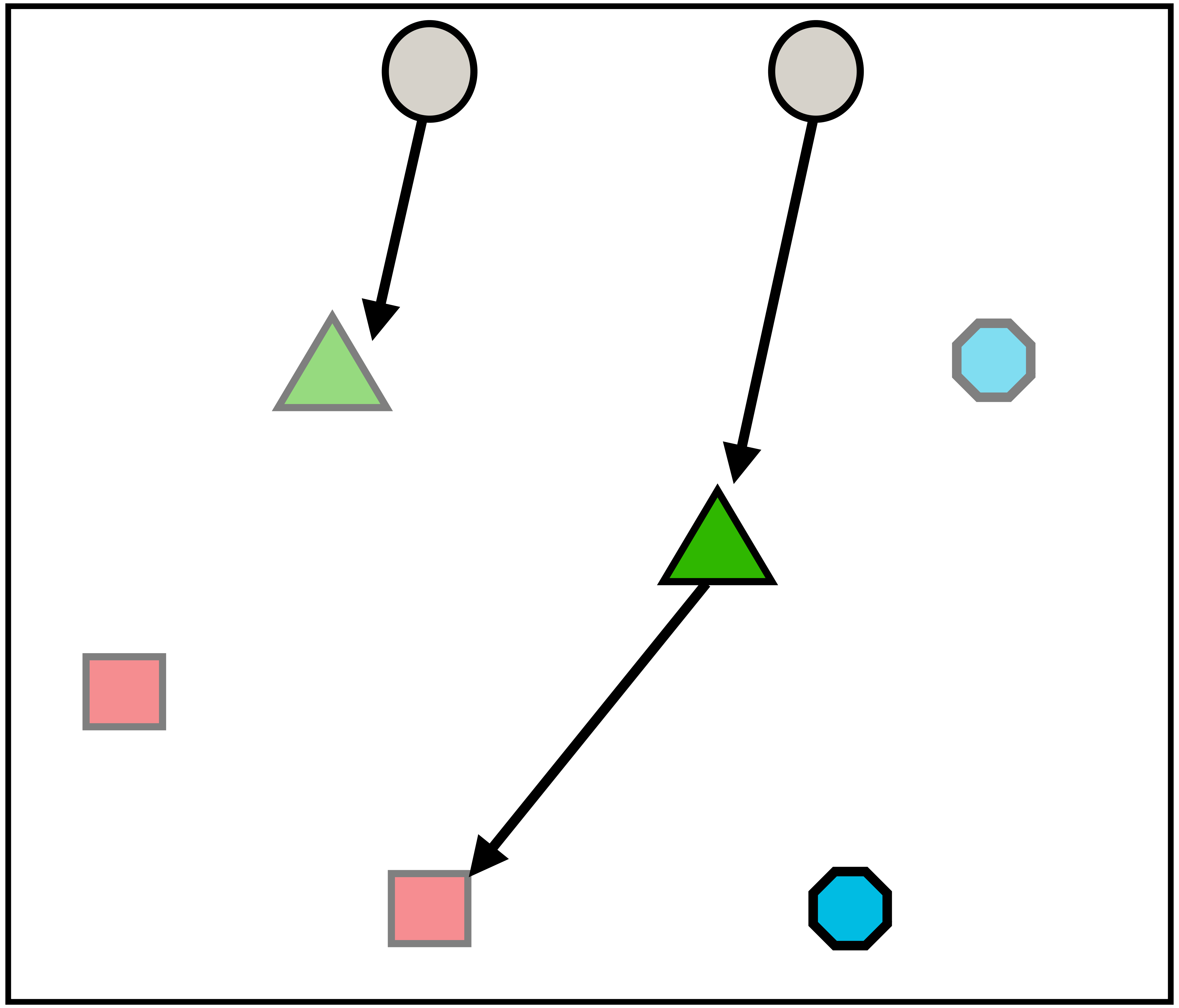
Parent graphs used for crossover and mutation





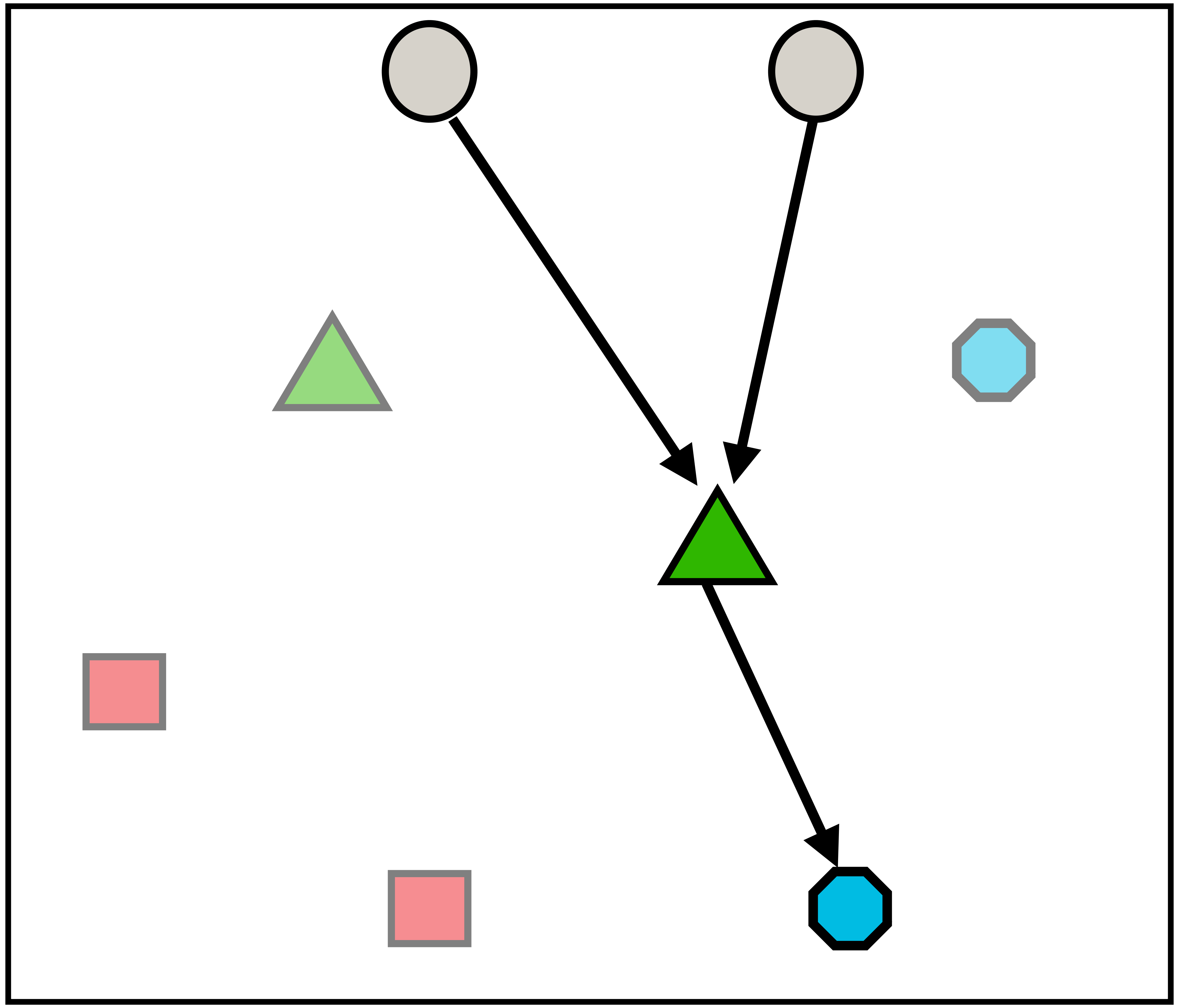
Evolutionary Algorithms : Repair operator

Graph after crossover and mutation



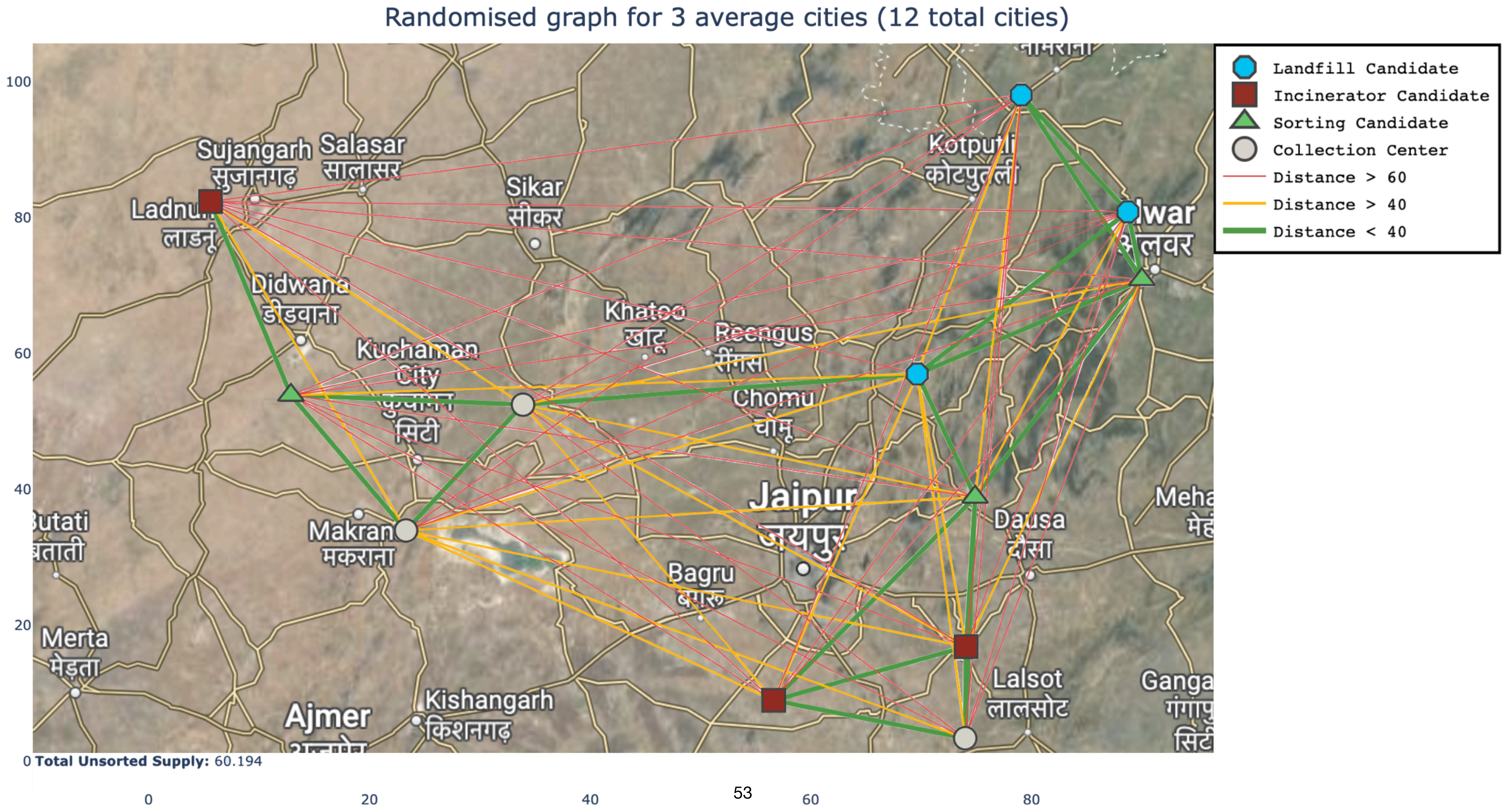
Evolutionary Algorithms: Repair operator

Graph after repair



Empirical Study

Empirical Study : Graph generation



Empirical Study : Optimisation frameworks

Baseline

CPLEX

Evolutionary Algorithms

Source: J. Blank and K. Deb. pymoo: multi-objective optimization in python. 2020.

Empirical Study : Algorithms

Baseline

Branch and cut

Multi-objective Evolution

NSGA-II

NSGA-III

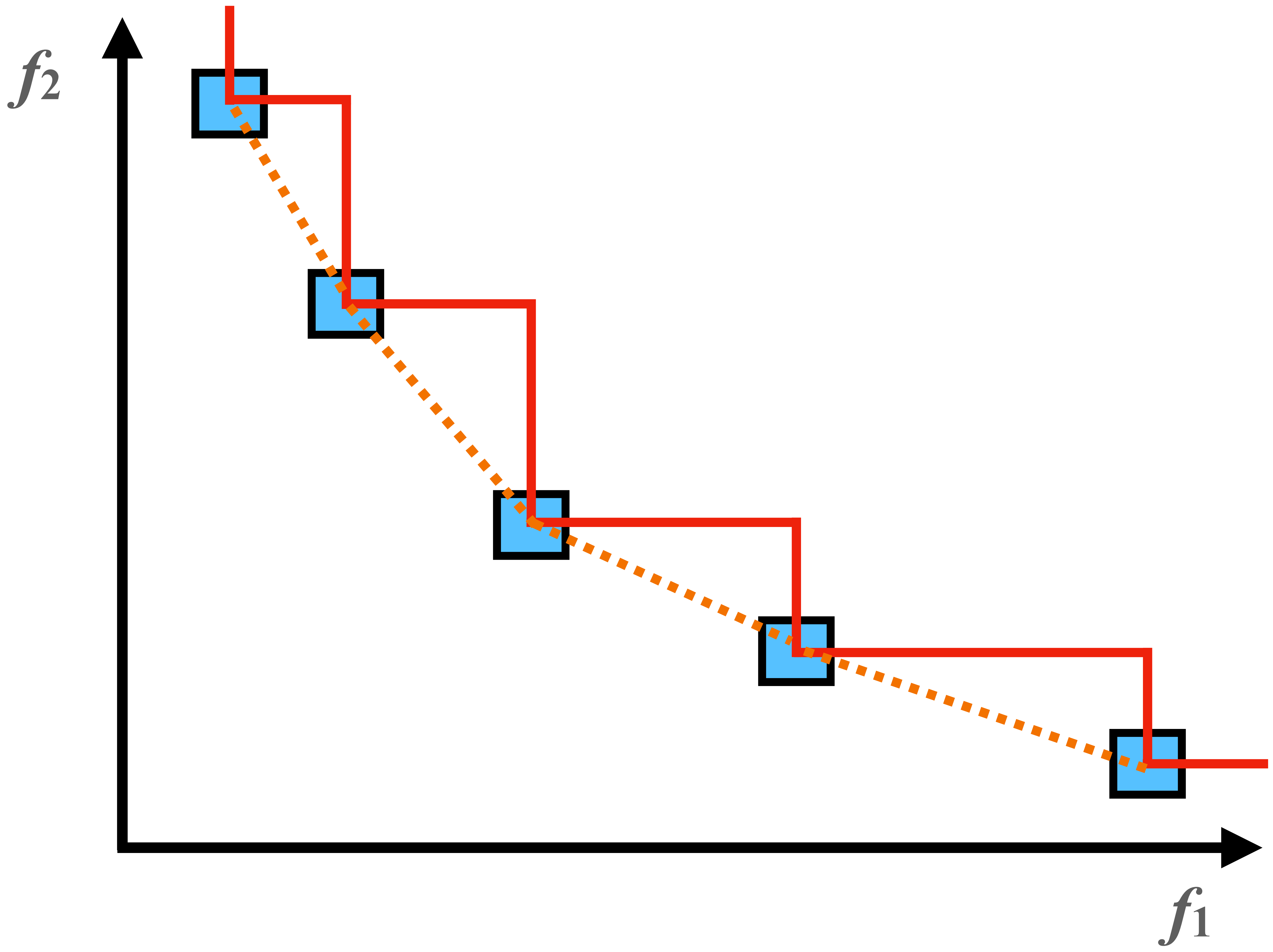
U-NSGA-III

C-TAEA

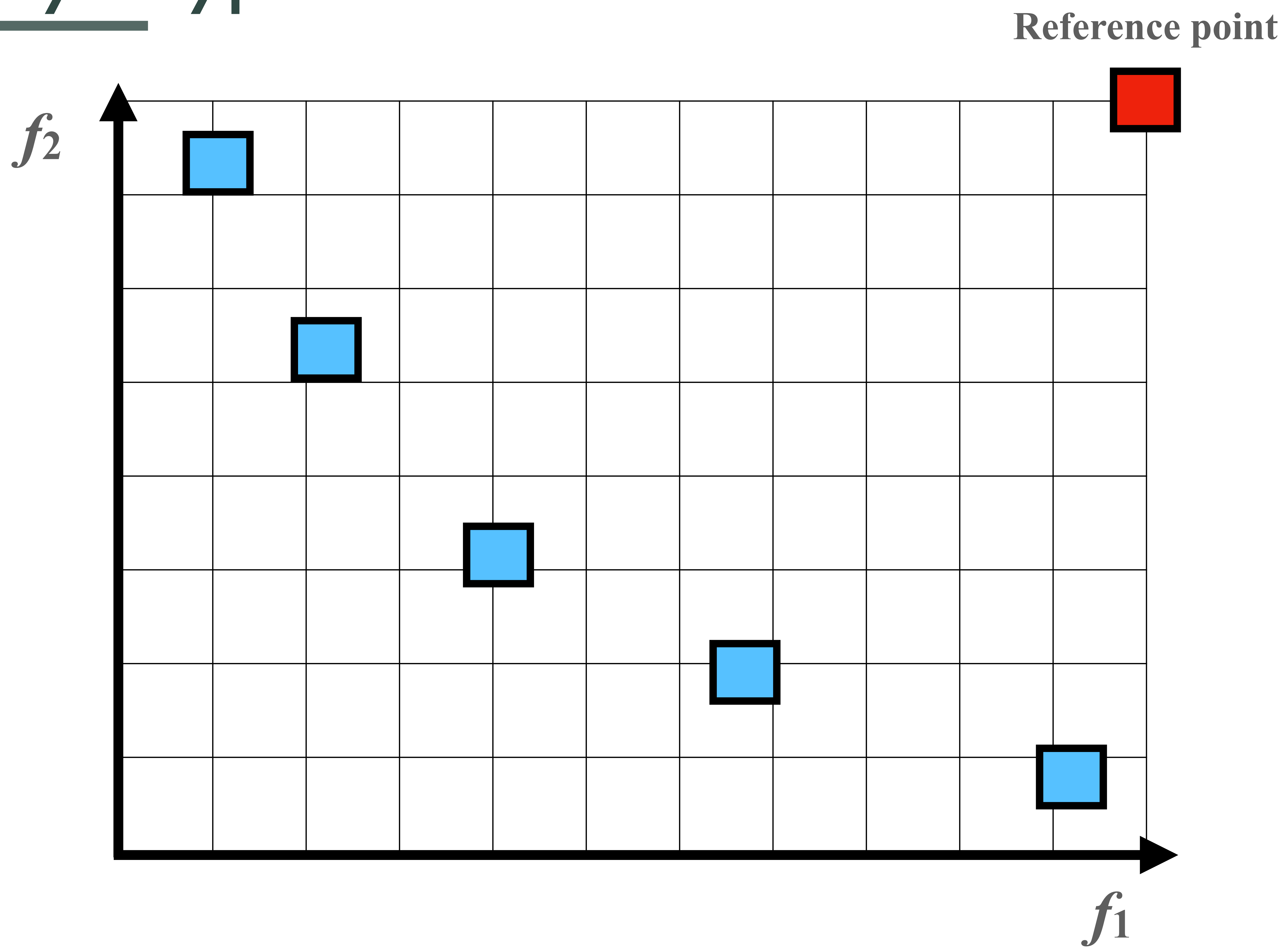
AGE-MOEA

Sources: [1], [2], [3], [4], [5], [6]

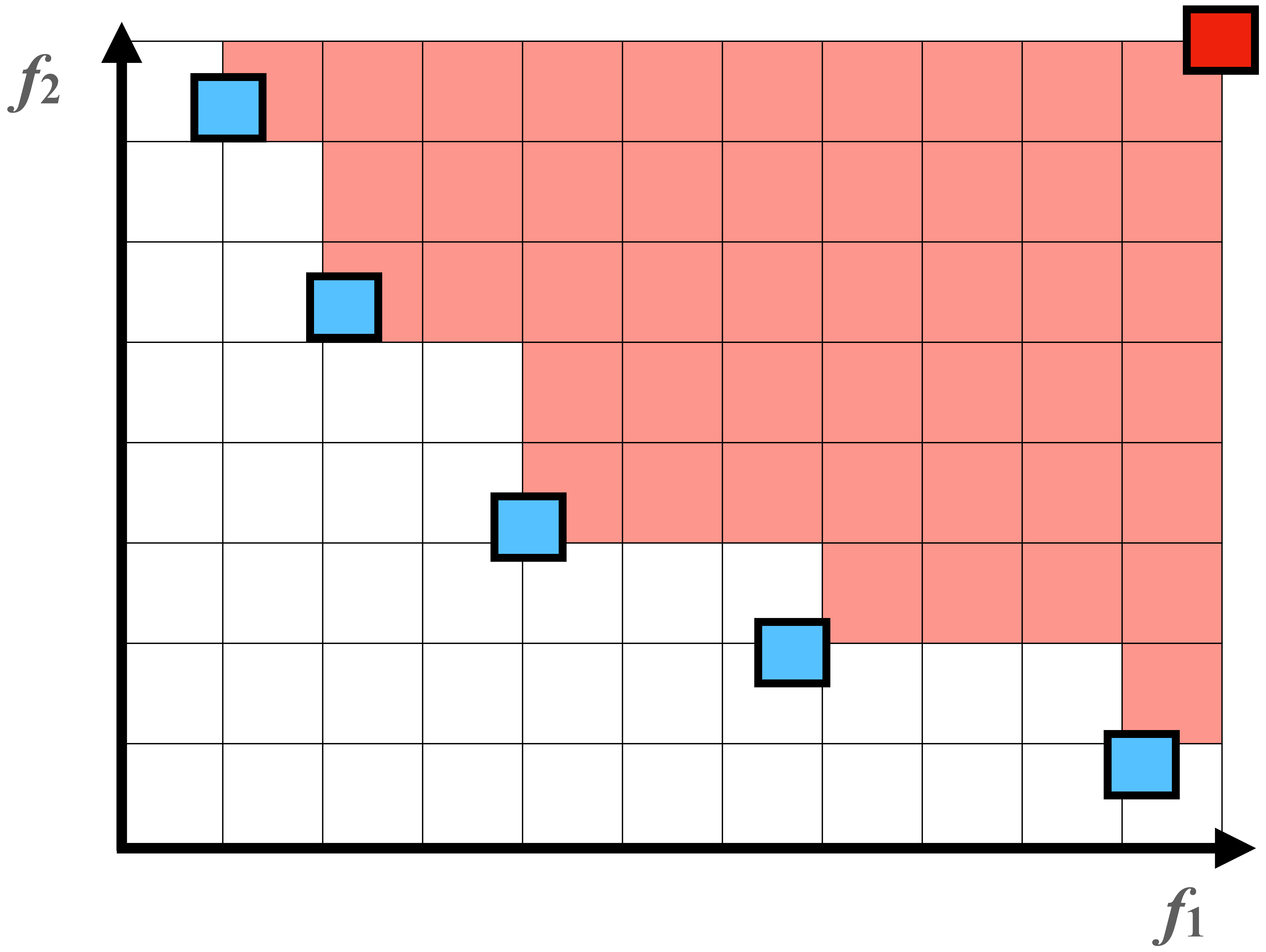
Empirical Study : Hypervolume metric



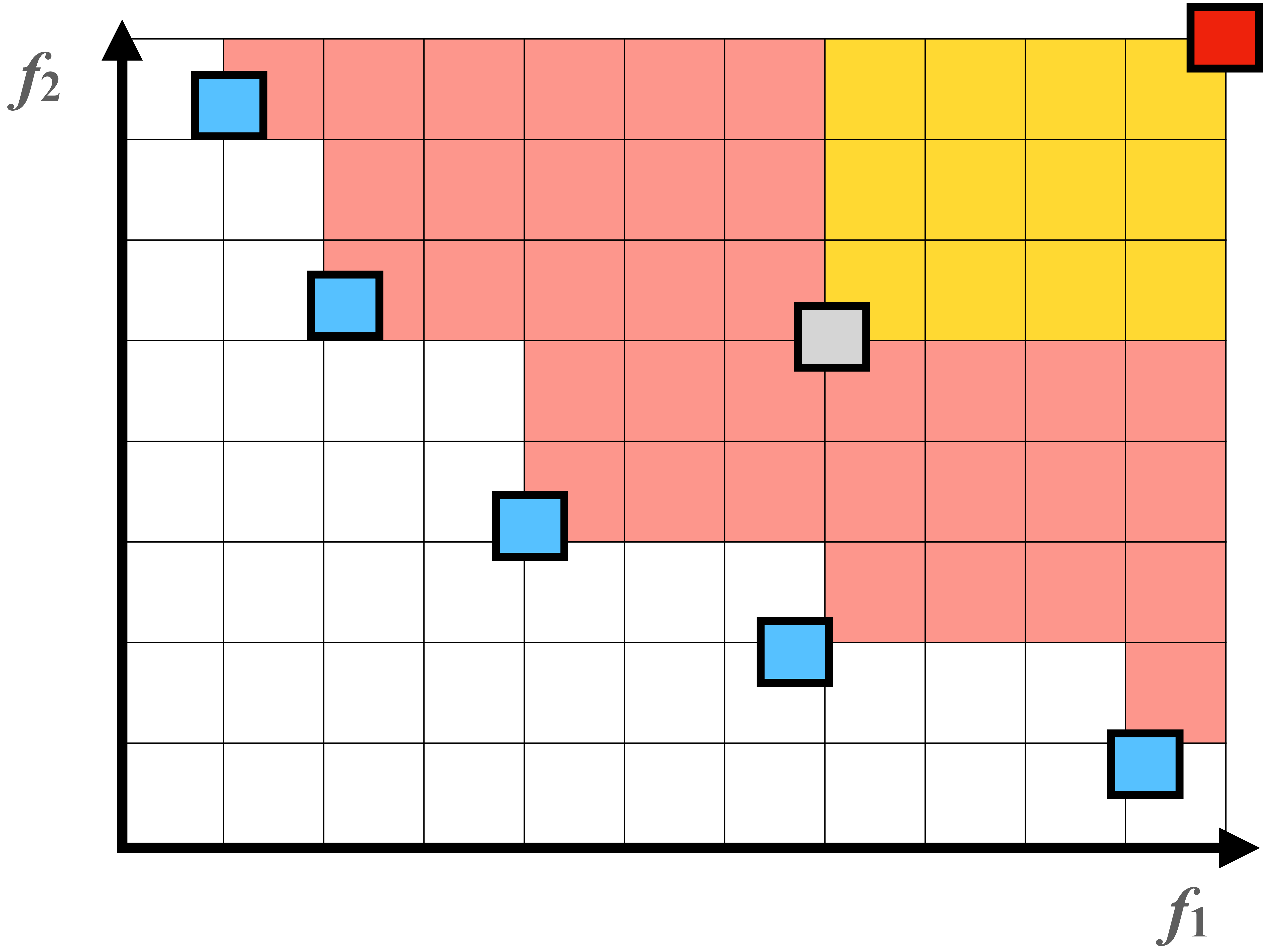
Empirical Study : Hypervolume metric



Empirical Study : Hypervolume metric



Empirical Study : Hypervolume metric



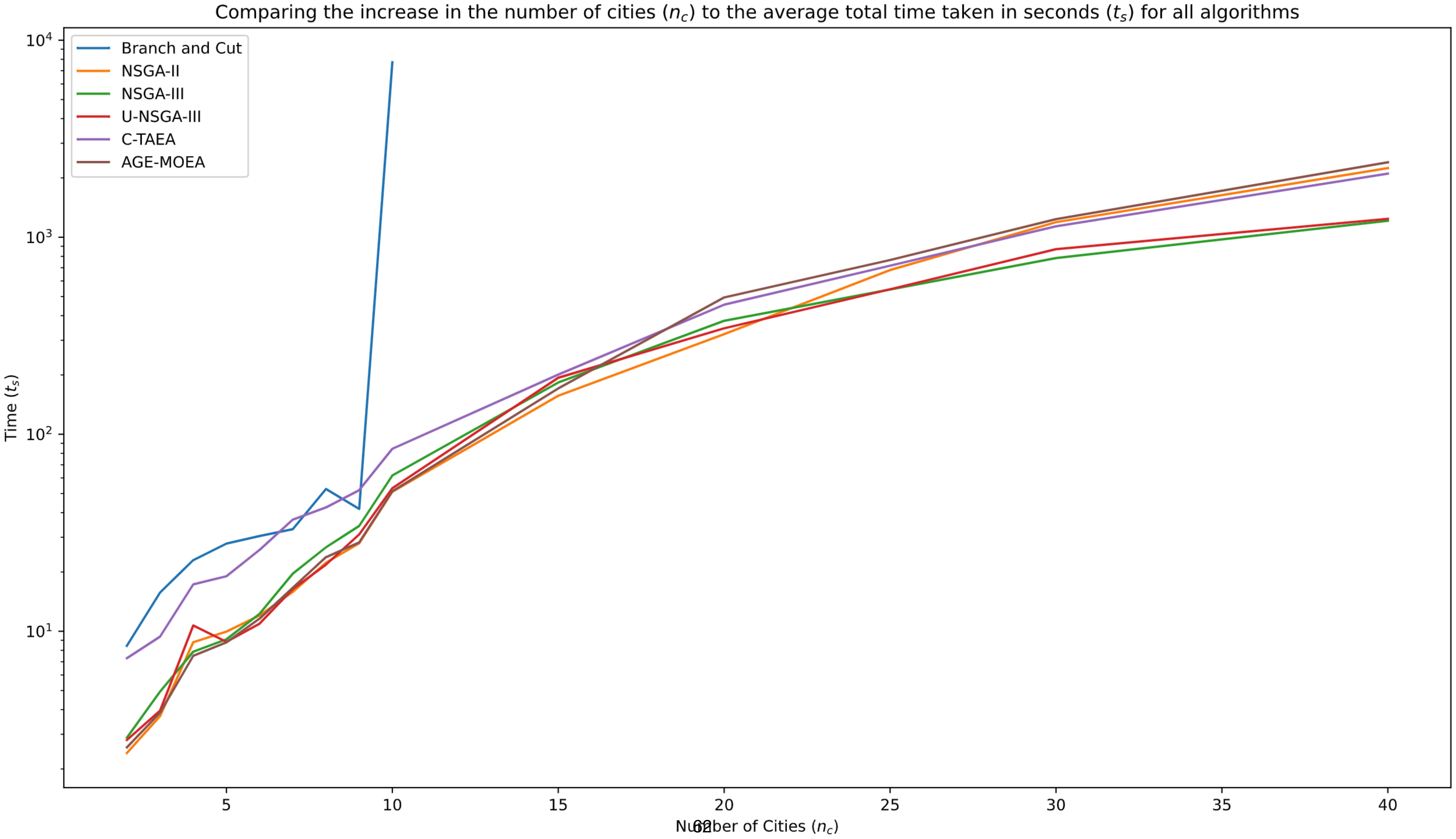
Empirical Study : Evolutionary parameters

Problem size	Population size	Reference points	Crossover probability	Mutation probability
Small scale (<= 40 cities)	200	200	80%	1%
Large scale (> 40 cities)	200	200	80%	0.5%

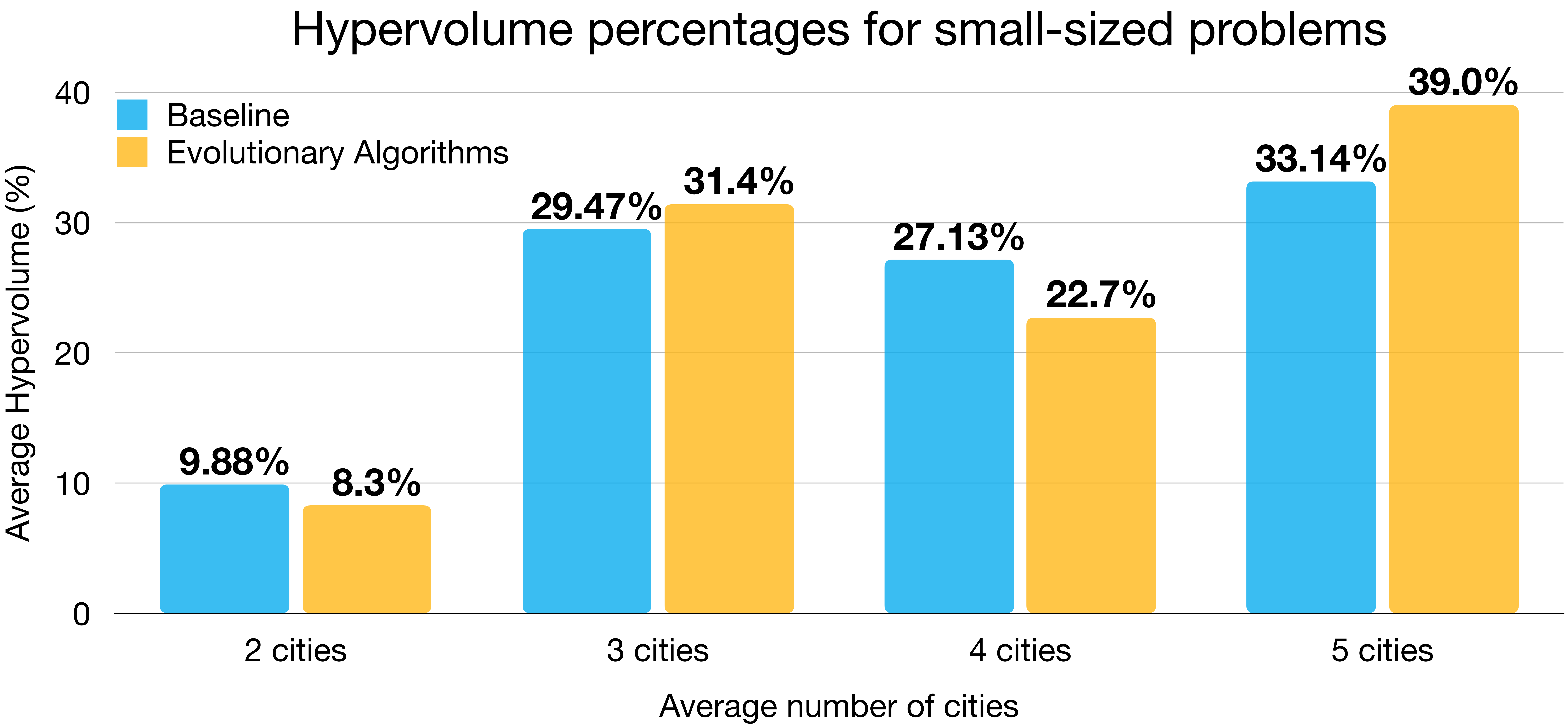
Empirical Study : Experimental setup

Number of average cities	Algorithms	Number of generated graphs	Seeds	Total number of experiments
2	CPLEX and Evolution	10	3	180
…	CPLEX and Evolution	10	3	180
10	CPLEX and Evolution	10	3	180
15	Evolution	10	3	150
20	Evolution	10	3	150
25	Evolution	10	3	150
30	Evolution	10	3	150
40	Evolution	10	3	150
		Total number of graphs: 420		**Total: 2370**

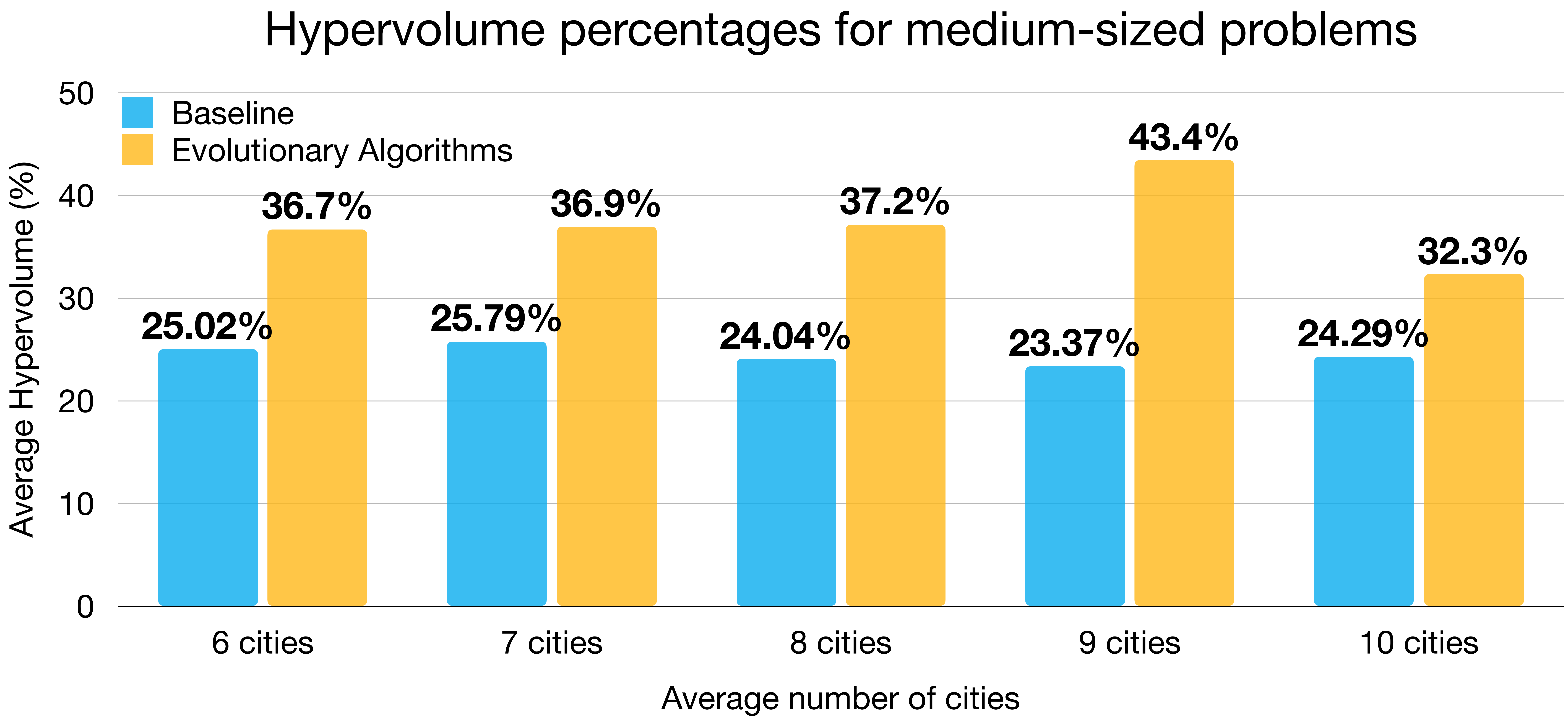
Empirical Study : Claim of time



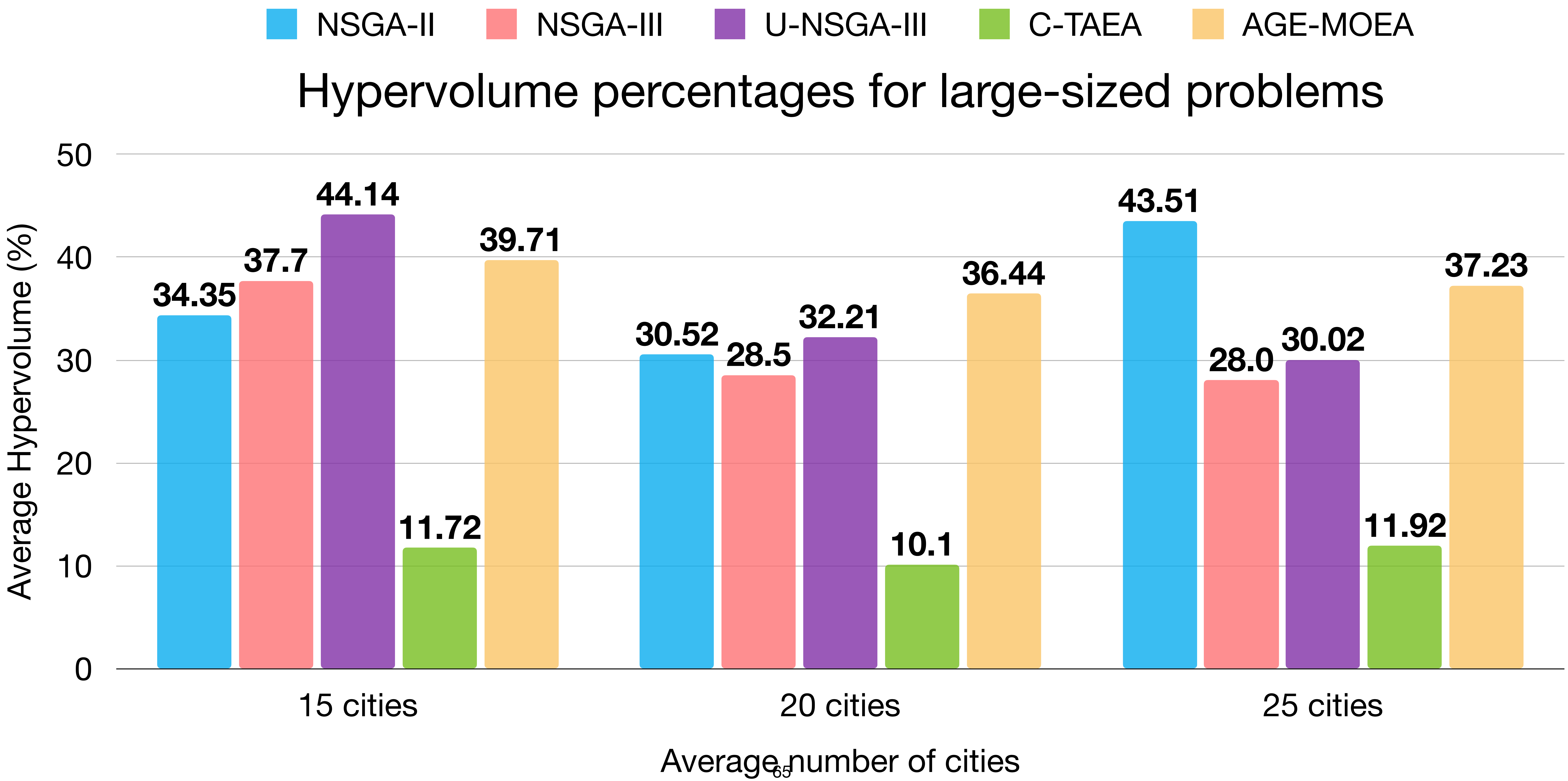
Empirical Study : Claim of solution quality



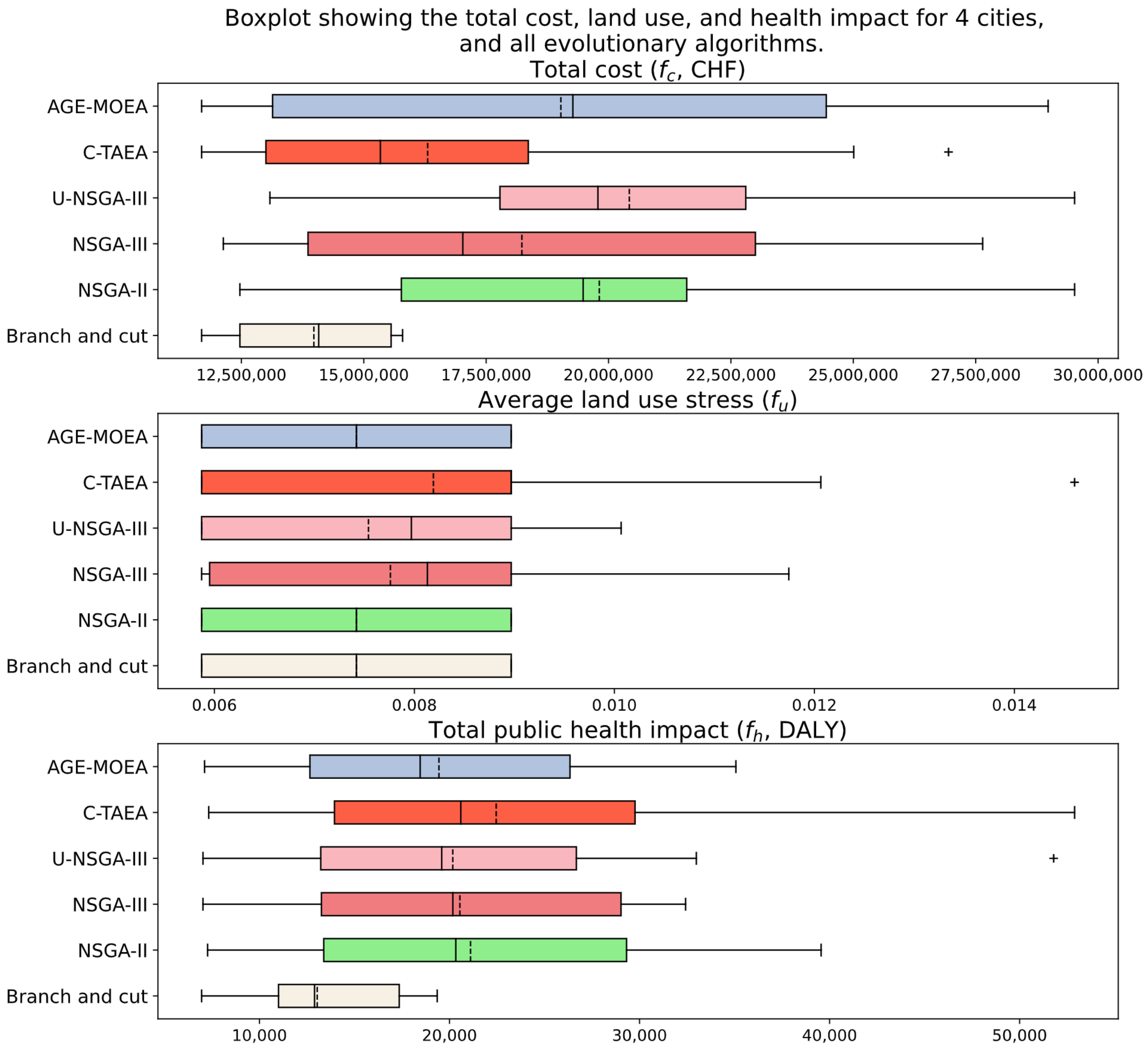
Empirical Study : Claim of solution quality

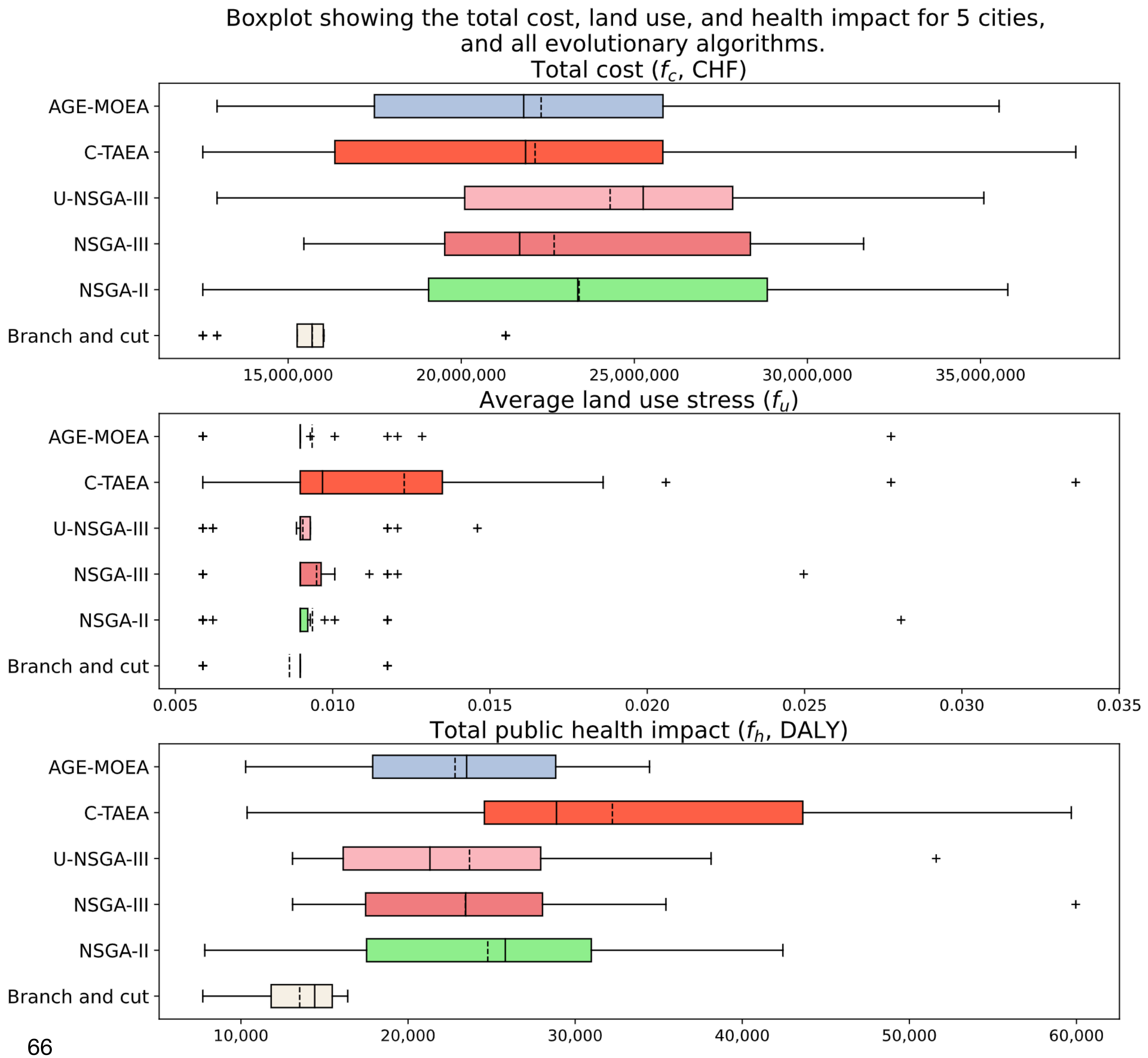


Empirical Study : Claim of solution quality



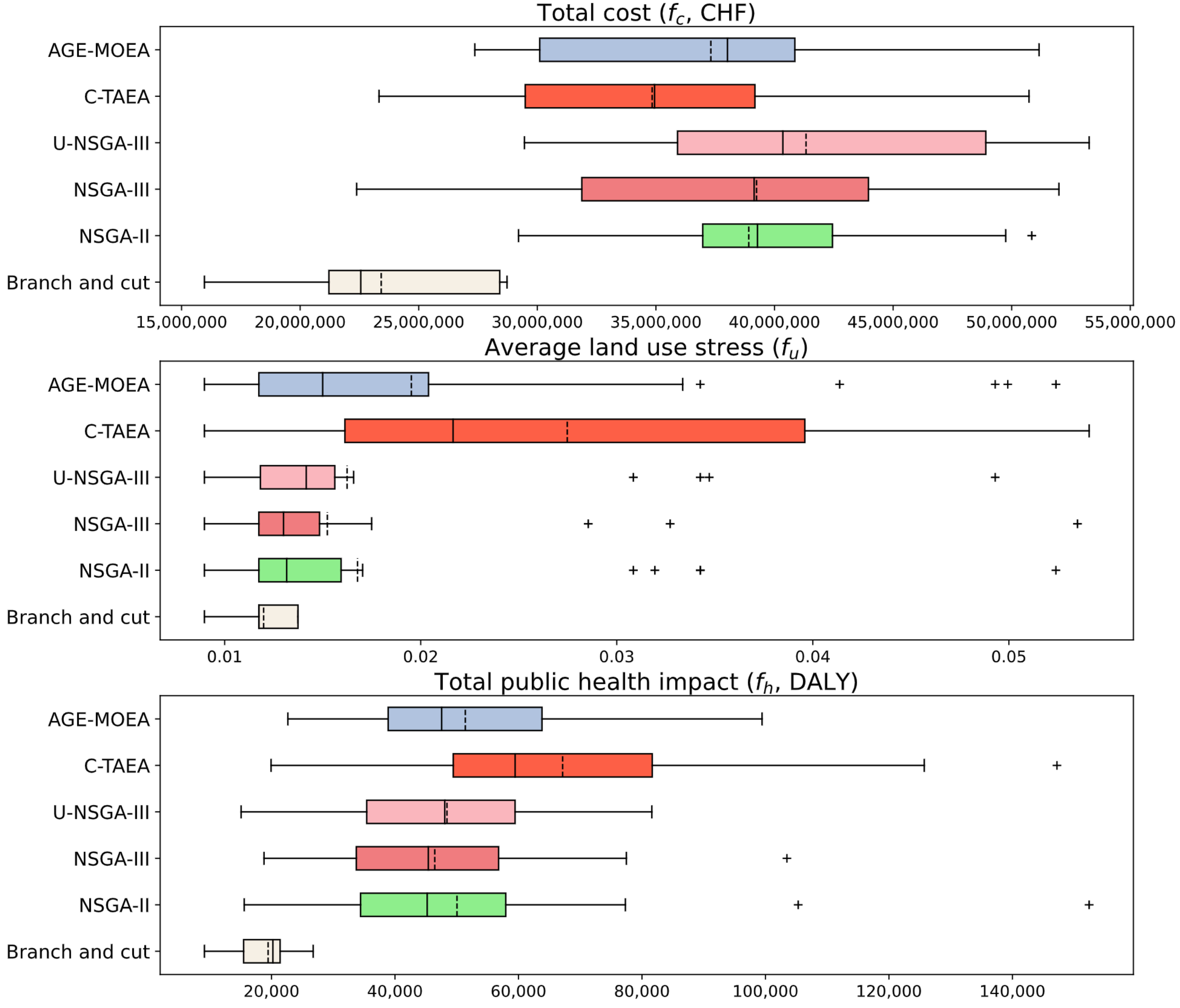
Empirical Study : Claim of solution diversity



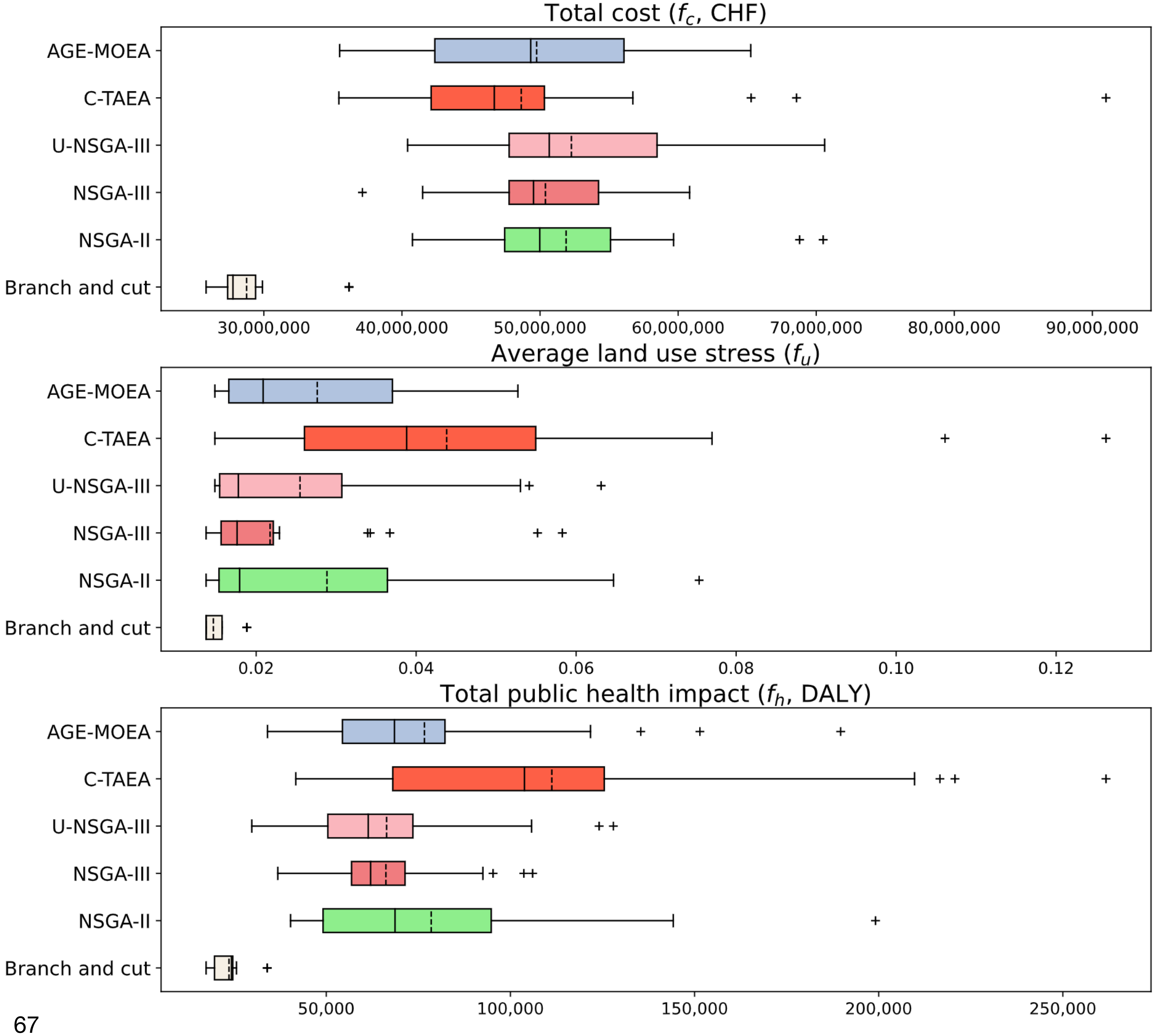


Empirical Study : Claim of solution diversity

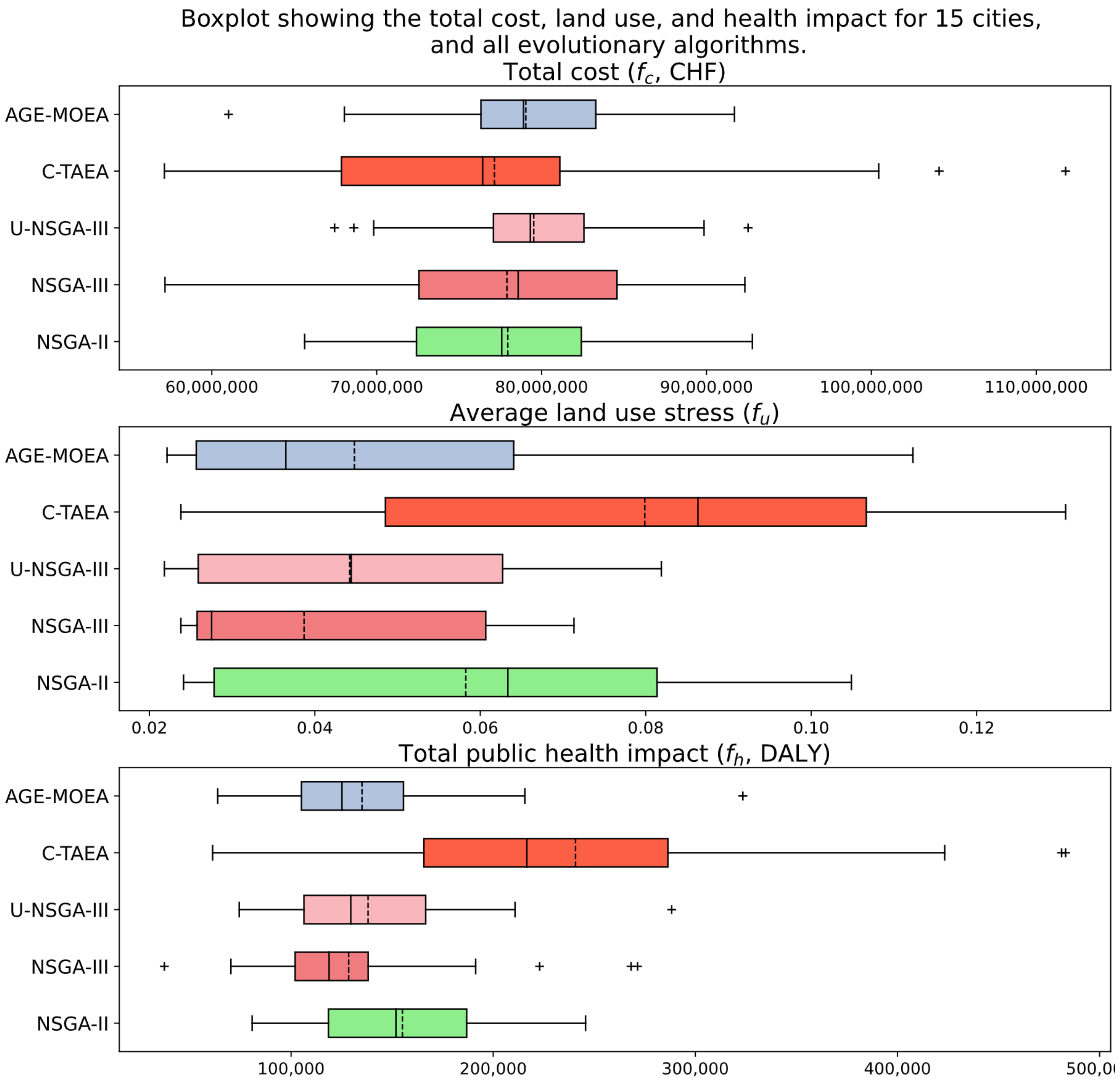
Boxplot showing the total cost, land use, and health impact for 8 cities, and all evolutionary algorithms.

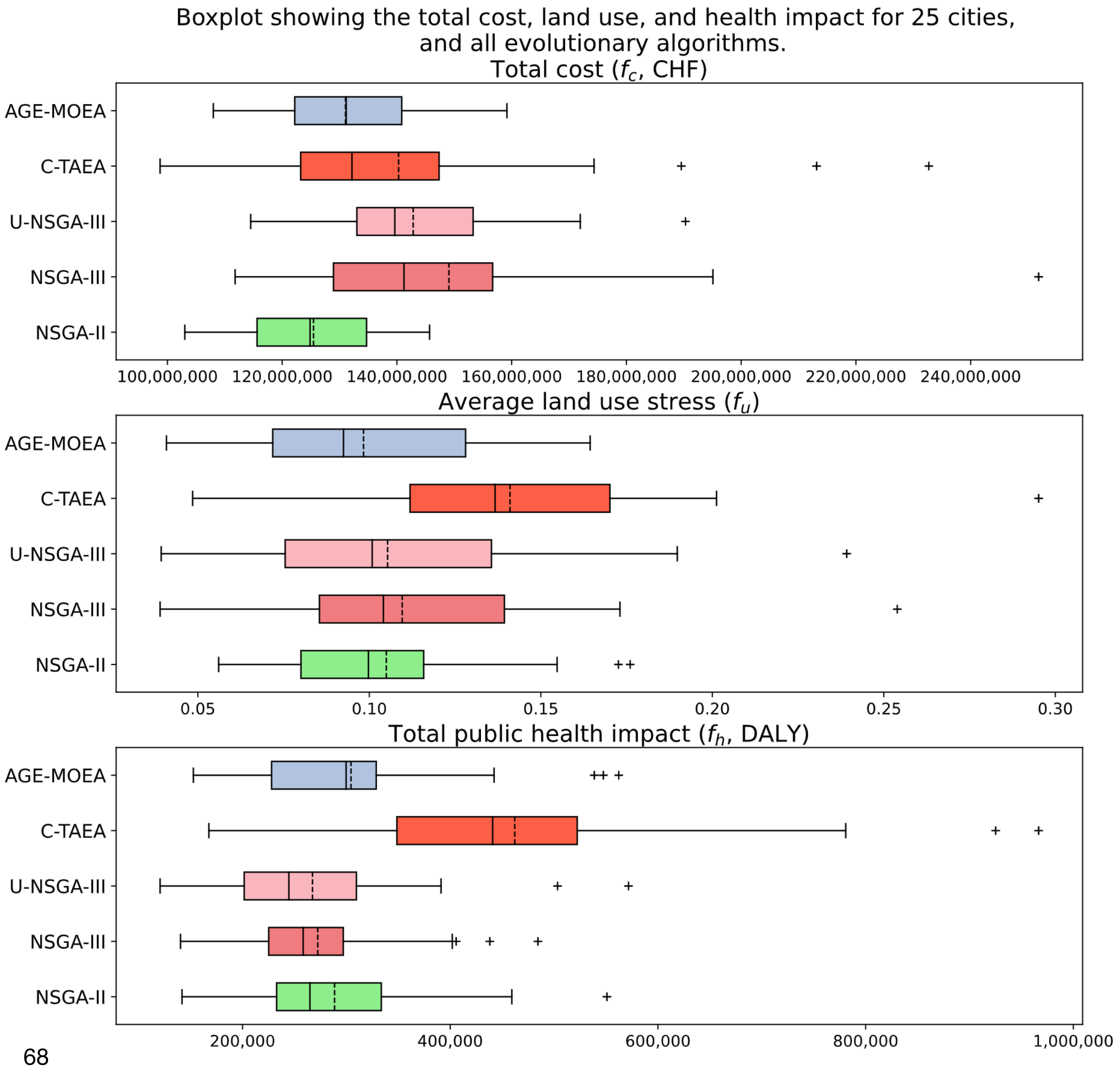


Boxplot showing the total cost, land use, and health impact for 10 cities, and all evolutionary algorithms.



Empirical Study : Claim of solution diversity





Conclusion

Conclusion: Advantages

- We extended the current state-of-the-art by incorporating Evolutionary Algorithms.
- We obtained better trade-off solutions compared to the current state-of-the-art.

Conclusion: Limitations

- Feasible solutions are not guaranteed for all algorithms in huge problems.

- Parameter tuning of the algorithms.

Conclusion: Future work

- A deeper look into the sorting facilities by incorporating the placement of sorting systems.

- Evolutionary algorithms that try to optimise multi-objective large-scale problems.

Thank you for your attention

Bibliography

1. Gomory, Ralph. 1960. *AN ALGORITHM FOR THE MIXED INTEGER PROBLEM*. RAND CORP SANTA MONICA CA. https://apps.dtic.mil/sti/citations/AD0616505 (August 14, 2022).

2. Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6(2): 182–97.

3. Deb, Kalyanmoy, and Himanshu Jain. 2014. "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints." *IEEE Transactions on Evolutionary Computation* 18(4): 577–601.

4. Seada, Haitham, and Kalyanmoy Deb. 2016. "A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives." *IEEE Transactions on Evolutionary Computation* 20(3): 358–69.

5. Li, Ke, Renzhi Chen, Guangtao Fu, and Xin Yao. 2019. "Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization." *IEEE Transactions on Evolutionary Computation* 23(2): 303–15.

6. Panichella, Annibale. 2019. "An Adaptive Evolutionary Algorithm Based on Non-Euclidean Geometry for Many-Objective Optimization." In *Proceedings of the Genetic and Evolutionary Computation Conference*, Prague Czech Republic: ACM, 595–603. https://dl.acm.org/doi/10.1145/3321707.3321839 (August 20, 2022).

Appendix

Evolutionary Algorithms: The binary decision variable, y

Evolutionary Algorithms: The binary decision variable, y

The Constraints

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J \qquad \sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

Evolutionary Algorithms: The binary decision variable, y

The Crossover

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J \qquad \sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

Population after single-point crossover

Small	Medium	Large
0	1	0
1	1	0
0	1	1
1	0	1

Evolutionary Algorithms: The binary decision variable, y

The Crossover

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J \qquad \sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

Find the individuals that sum to two/three

Small	Medium	Large
0	1	0
1	1	0
0	1	1
1	0	1

Istituto Dalle Molle di studi sull'intelligenza artificiale IDSIA
Università della Svizzera italiana

Evolutionary Algorithms: The binary decision variable, y

The Crossover

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J$$

$$\sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K$$

$$\sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

Select two/three random columns to set to zero for each row.

Small	Medium	Large
0	1	0
0	0	0
0	0	1

Evolutionary Algorithms: The binary decision variable, y

The Mutation

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J \qquad \sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

Small	Medium	Large
0	1	0
0	1	0
0	0	0
0	0	1

Evolutionary Algorithms: The binary decision variable, y

The Mutation

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J \qquad \sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K \qquad \sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

Apply the binary mutation operator

Small	Medium	Large
1	1	0
0	1	1
0	1	0
0	0	0

Evolutionary Algorithms: The binary decision variable, y

The Mutation

$$\sum_{l \in L(j)} y_{jl} \leq 1, \quad \forall j \in J$$

$$\sum_{l \in L(k)} y_{kl} \leq 1, \quad \forall k \in K$$

$$\sum_{l \in L(k')} y_{k'l} \leq 1, \quad \forall k' \in K'$$

Set the other values in the mutated one rows to zero.

Small	Medium	Large
1	0	0
0	0	1
0	1	0
0	0	0

Evolutionary Algorithms: The continuous decision variable, f

The Constraints

$$\sum_{j \in J} f_{ij} = D_i, \quad \forall i \in I \qquad\qquad \sum_{i \in I} f_{ij} = \sum_{k \in K \cup K'} f_{jk}, \quad \forall j \in J$$

Evolutionary Algorithms: The continuous decision variable, f

The mutation/crossover procedure

$$\sum_{j \in J} f_{ij} = D_i, \quad \forall i \in I \qquad\qquad \sum_{i \in I} f_{ij} = \sum_{k \in K \cup K'} f_{jk}, \quad \forall j \in J$$

$\frac{f}{Supply}$ — Normalise using the supply

$\frac{f_{ij}}{\sum f_i}$ — Make each row sum to one

$f \cdot Supply$ — Denormalise using the supply

Evolutionary Algorithms: The continuous decision variable, f

The mutation/crossover procedure

$$\sum_{j\in J} f_{ij} = D_i, \quad \forall i \in I \qquad\qquad \sum_{i\in I} f_{ij} = \sum_{k\in K\cup K'} f_{jk}, \quad \forall j \in J$$

Links ij

- $\frac{f}{Supply}$ — Normalise using the supply
- $\frac{f_{ij}}{\sum f_j}$ — Make each row sum to one
- $f \cdot Supply$ — Denormalise using the supply

→

Links jk & jk'

- $\sum f_i$ — Obtain new supplies from sorting facilities
- $\frac{f}{Supply}$ — Normalise using the supply
- $\frac{f_{jk}}{\sum f_k}$ — Make each row sum to one
- $f \cdot Supply$ — Denormalise using the supply

Evolutionary Algorithms: The continuous decision variable, f

The mutation/crossover procedure

$$\sum_{j \in J} f_{ij} = D_i, \quad \forall i \in I \qquad \sum_{i \in I} f_{ij} = \sum_{k \in K \cup K'} f_{jk}, \quad \forall j \in J$$

Population before normalisation

Supplies		j_1	j_2	j_3
$D_1 = 40$	i_1	10	20	30
$D_2 = 30$	i_2	20	30	40
$D_3 = 20$	i_3	0	0	10

Evolutionary Algorithms: The continuous decision variable, f

The mutation/crossover procedure

$$\sum_{j \in J} f_{ij} = D_i, \quad \forall i \in I \qquad \sum_{i \in I} f_{ij} = \sum_{k \in K \cup K'} f_{jk}, \quad \forall j \in J$$

Population after normalisation

Supplies
$D_1 = 40$
$D_2 = 30$
$D_3 = 20$

	j_1	j_2	j_3
i_1	$\frac{10}{40}$	$\frac{20}{40}$	$\frac{30}{40}$
i_2	$\frac{20}{30}$	$\frac{30}{30}$	$\frac{40}{30}$
i_3	0	0	$\frac{10}{20}$

Sums
$\frac{60}{40}$
$\frac{90}{30}$
$\frac{10}{20}$

Evolutionary Algorithms: The continuous decision variable, f

The mutation/crossover procedure

$$\sum_{j \in J} f_{ij} = D_i, \quad \forall i \in I \qquad\qquad \sum_{i \in I} f_{ij} = \sum_{k \in K \cup K'} f_{jk}, \quad \forall j \in J$$

Sum to one

Supplies		j_1	j_2	j_3	**Sums**
$D_1 = 40$	i_1	$\frac{10}{60}$	$\frac{20}{60}$	$\frac{30}{60}$	1
$D_2 = 30$	i_2	$\frac{20}{90}$	$\frac{30}{90}$	$\frac{40}{90}$	1
$D_3 = 20$	i_3	0	0	$\frac{10}{10}$	1

Evolutionary Algorithms: The continuous decision variable, f

The mutation/crossover procedure

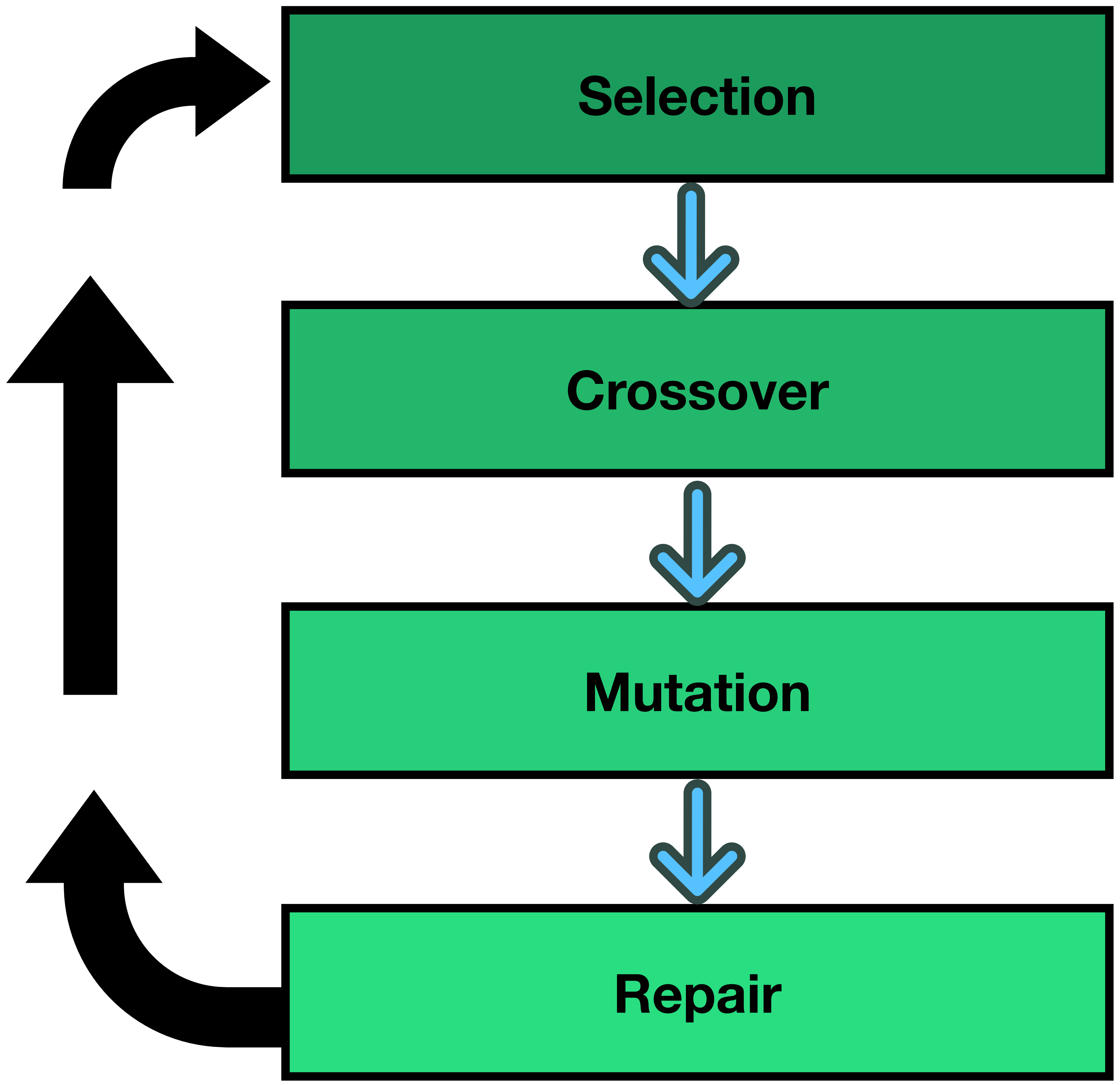
$$\sum_{j \in J} f_{ij} = D_i, \quad \forall i \in I \qquad\qquad \sum_{i \in I} f_{ij} = \sum_{k \in K \cup K'} f_{jk}, \quad \forall j \in J$$

Obtain new population by denormalising

Supplies		j_1	j_2	j_3	Sums
$D_1 = 40$	i_1	$\frac{10}{60} \cdot 40$	$\frac{20}{60} \cdot 40$	$\frac{30}{60} \cdot 40$	40
$D_2 = 30$	i_2	$\frac{20}{90} \cdot 30$	$\frac{30}{90} \cdot 30$	$\frac{40}{90} \cdot 30$	30
$D_3 = 20$	i_3	0	0	20	20

Evolutionary Algorithms: The repair operator

Evolutionary Algorithms: The repair operator



Evolutionary Algorithms: The repair operator

Binary population after mutation

	Small	Medium	Large
j_1	0	0	0
j_2	0	0	1
j_3	0	0	0

Evolutionary Algorithms: The repair operator

Binary population after mutation

	Small	Medium	Large
j_1	0	0	0
j_2	0	0	1
j_3	0	0	0

Continuous population after mutation

Supplies
$D_1 = 60$
$D_2 = 30$
$D_3 = 20$

	j_1	j_2	j_3
i_1	20	40	20
i_2	0	20	10
i_3	5	5	10

Evolutionary Algorithms: The repair operator

Binary population after mutation

	Small	Medium	Large
j_1	0	0	0
j_2	0	0	1
j_3	0	0	0

Continuous population after setting to zero

Supplies		j_1	j_2	j_3
$D_1 = 60$	i_1	0	40	0
$D_2 = 30$	i_2	0	20	0
$D_3 = 20$	i_3	0	5	0

Evolutionary Algorithms: The repair operator

Continuous population after repair

Supplies		j_1	j_2	j_3
$D_1 = 60$	i_1	0	60	0
$D_2 = 30$	i_2	0	30	0
$D_3 = 20$	i_3	0	20	0