GENETIC ALGORITHM MEDIATOR 🧬

Presented By : Group Technologia

Kang, Chun-Yu

Nicoló, Rossi

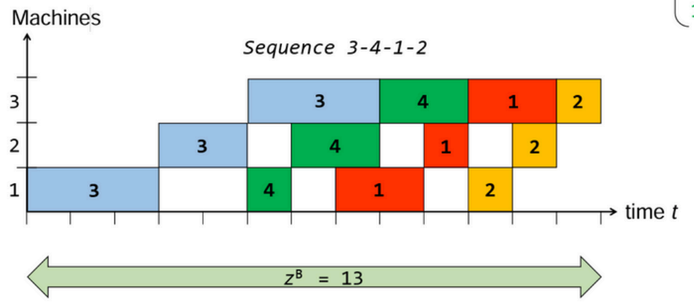
Lin, Chin-Jung

PROBLEM DEFINITION

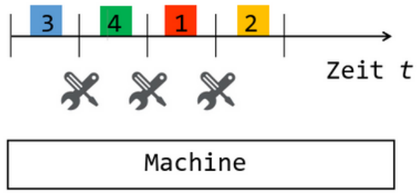
- The project focuses on optimizing production schedules in a supply chain using Automated Negotiation.
- Two agents negotiate the best order to produce a batch of goods, each trying to minimize their own production cost.
- A third agent, the Mediator, generates and evaluates different contract sequences to find a solution that works for both agents.

CUSTOMER COST

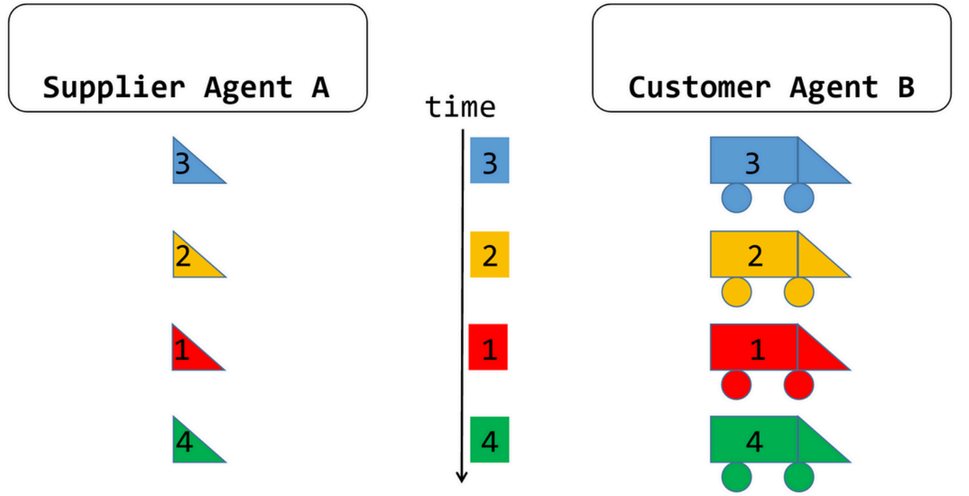
$$(t_{h,i}) = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 1 & 1 \\ 3 & 2 & 3 \\ 1 & 2 & 2 \end{pmatrix}$$



SUPPLIER COST



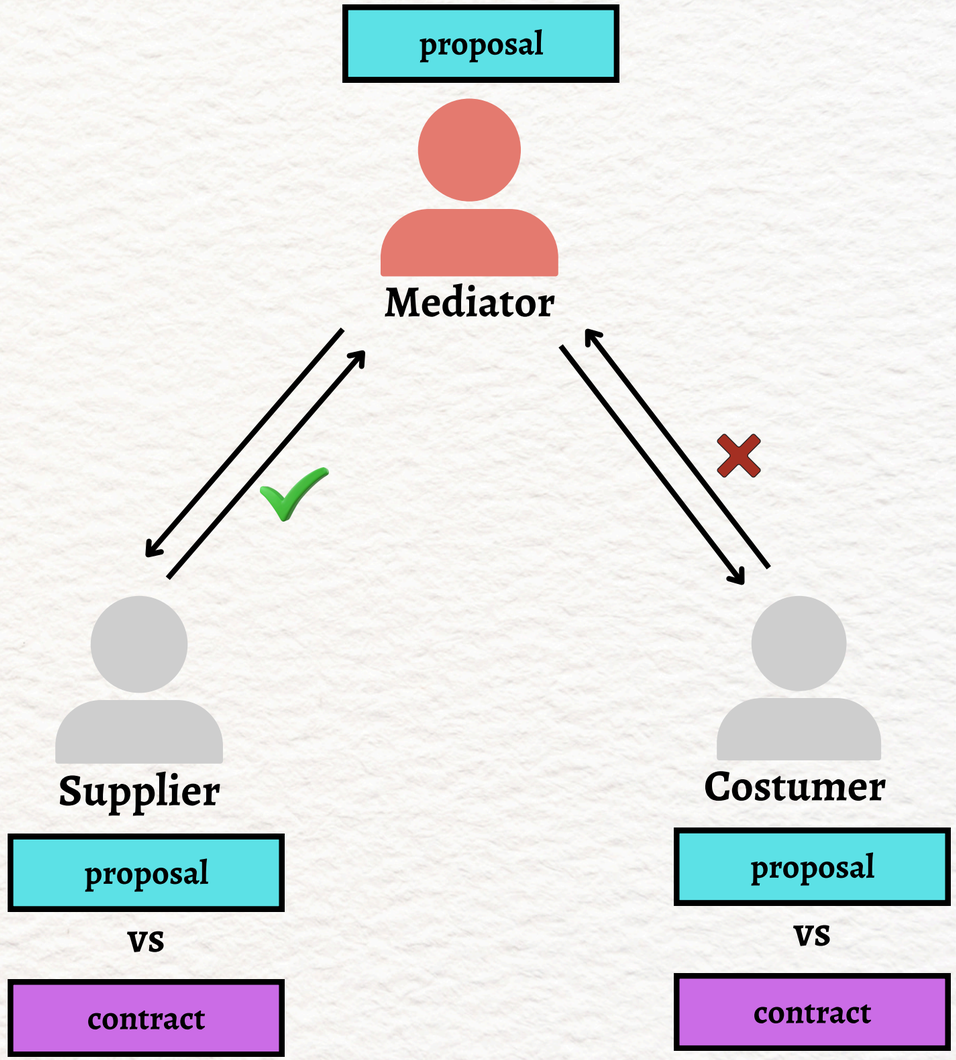
$$(r_{h,i}) = \begin{pmatrix} 0 & 6 & 8 & 2 \\ 4 & 0 & 2 & 3 \\ 3 & 7 & 0 & 6 \\ 2 & 8 & 4 & 0 \end{pmatrix}$$



Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

MEDIATOR

- The mediator creates a new contract (proposal) and sends it to both agents along with a reference contract.
- Each agent compares the cost of both contracts and votes “yes” if the proposal is better for them.
- A contract wins if it receives more favorable votes (i.e., lower cost from the agent’s perspective).



Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

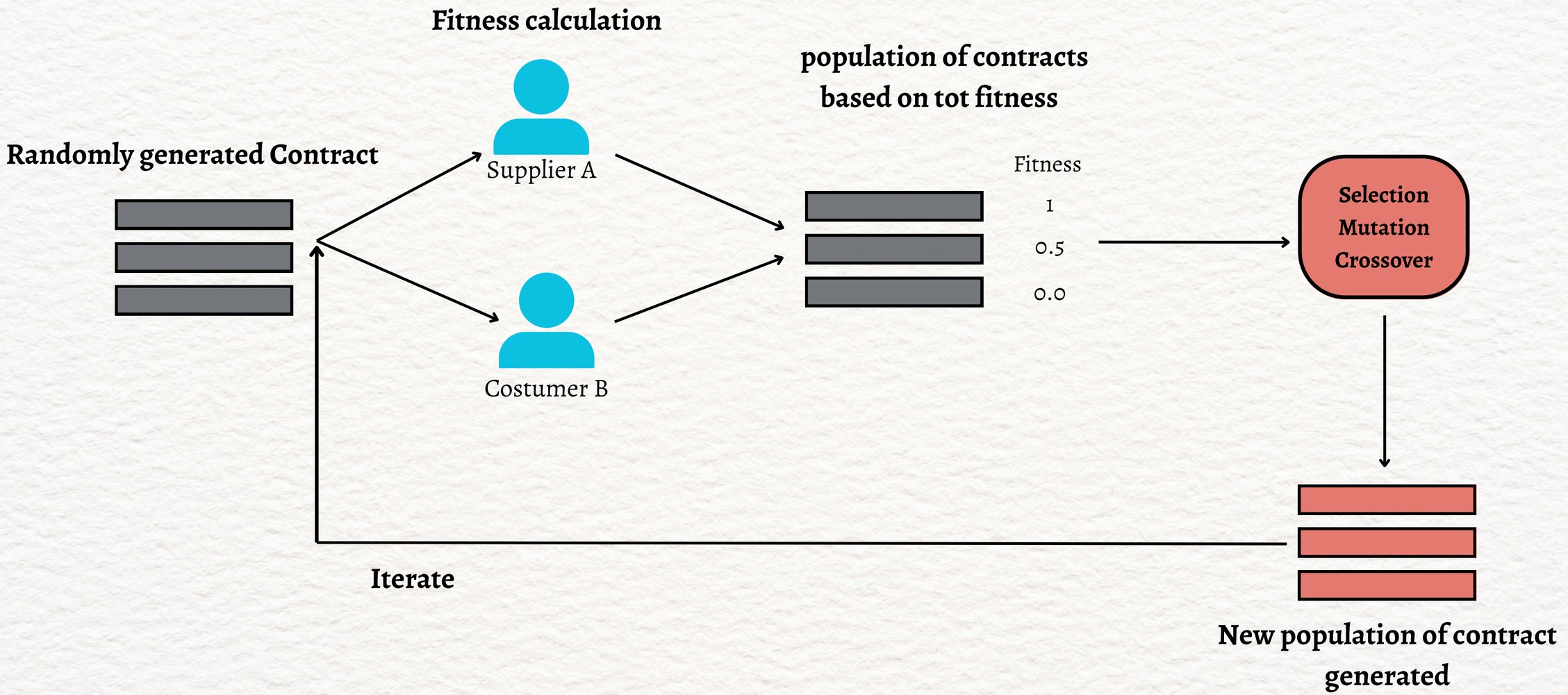
CODING AND DECODING

A contract is a permutation of non-repeatable numbers:

[4,3,2,20...,7]

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

OUR FRAMEWORK



Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

FITNESS SCORE CALCULATION

RANKED CONTRACTS

- Generate random contracts using initContract()
- Use CalculateFitnessScore() for evaluation from both Agents
- Sort contracts by vote score

SORTING

- Return result from Dummy comparison
- Sort the contracts
- Top-performing solutions are prioritized for further methods

DUMMY COMPARISON

- Picks a reference contract, used in scoring or comparisons
- `int[] dummyContract = rankedContracts[(rankedContracts.length / 2) - 1];`
- Fitness scoring how many agents prefer the new contract over the dummy

Score-to-Fitness Mapping:

- 2 votes → fitness = 1.0
- 1 vote → fitness = 0.5
- 0 vote → fitness = 0.0

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

SELECTION

TEMPERATURE

A temperature value is decreased every round:

$$T(t) = T_o \times \alpha$$

this temperature is used for compute the selection probability of a contract:

$$P(i) = \frac{e^{f(i)/T(t)}}{\sum_j e^{f(j)/T(t)}}$$

this ensure both exploration and exploitation.

RANK

Assigns selection probability based on the rank of fitness:

x	*rank(x)*	*P(x)*
contract1	*3*	*0.5*
contract2	*2*	*0.(3)*
contract3	*1*	*0.(1)*

ensuring a balanced chance for all contracts and reducing premature convergence.

TOURNAMENT

Picks a two contracts at random(*contract1, contract2*) and selects the best among them based on fitness:

fitness(contract1) < fitness(contract2) → contract2

fitness(contract1) > fitness(contract2) → contract1

This method promoting strong solutions while maintaining diversity.

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

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

CROSSOVER

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

ODER

Step:

1. **Pick two position from**
   a. **MAMA: [1,5,2,3,6,4]**
2. **Copy the segment**
   a. **Child1: [1,5]**
3. **Fill the rest from another Parent in order, skipping duplicates.**
   a. **Child1: [1,5,4,3,2,6] (PAPA: [4,~~1~~,3,2,~~5~~,6])**

POSITION BASED

1. **Randomly select a few positions**
   a. **Assume we select position: 0, 1, 5**
   b. **MAMA: [1,5,2,3,6,4]**
2. **Copy the genes at those positions**
   a. **Child1: [1,5,?,?,?,4]**
3. **Fill the remaining positions with genes from PAPA, in the order they appear, skipping duplicates.**
   a. **Child1: [1,5,3,2,6,4] (PAPA: [~~4,1~~,3,2,~~5~~,6])**

CYCLE

1. **Find cycles between MAMA and PAPA based on matching positions.**
   a. **Assume our start point is 3**
   b. **MAMA[3] = 3, PAPA[2] = 3, MAMA[2] = 2, PAPA[3] = 2. End Point**
2. **We get the cycle index[2,3]. Copy one cycle from MAMA to the child.**
   a. **Child 1: [?,?,2,3,?,?] (MAMA: [1,5,2,3,6,4])**
3. **Fill the rest from PAPA.**
   a. **Child 1: [4,1,2,3,5,6] (PAPA: [4,1,~~3,2~~,5,6])**

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

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

MUTATION

SCRAMBLE

Step:

1. **Randomly select a subrange (e.g., positions 1 to 3).**
   a. **MAMA: [1,5,2,3,6,4]**
2. **Shuffle the genes within that subrange.**
   a. **[5,2,3] → [5,3,2]**
3. **Keep the rest of the genes unchanged.**
   a. **Result: [1,5,3,2,6,4]**

INVERSE

Step:

1. **Randomly select a subrange (e.g., positions 1 to 3).**
   a. **MAMA: [1,5,2,3,6,4]**
2. **Reverse the order of the genes within that subrange.**
   a. **[5,2,3] → [3,2,5]**
3. **Leave the rest unchanged.**
   a. **Result: [1,3,2,5,6,4]**

DISPLACEMENT

1. **Randomly select a subrange of genes (e.g., positions 1 to 3).**
   a. **We get: [5,2,3]**
2. **Remove that subrange from the chromosome.**
   a. **MAMA After remove: [1,6,4]**
3. **Reinsert it at a different random position. (e.g., insert position: 1)**
   a. **Result: [1,6,5,2,3,4]**

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

TEST CASES

We test the model by:

- Iterating with different selections methods(Rank, Temeprature, Tournament)
- Iterating with different crossover methods(Oder, Cycle, Position Based)
- Iterating with different mutation methods(Scramble, Inverse, Displacment)

and in three different scenarios:

10 ROUNDS

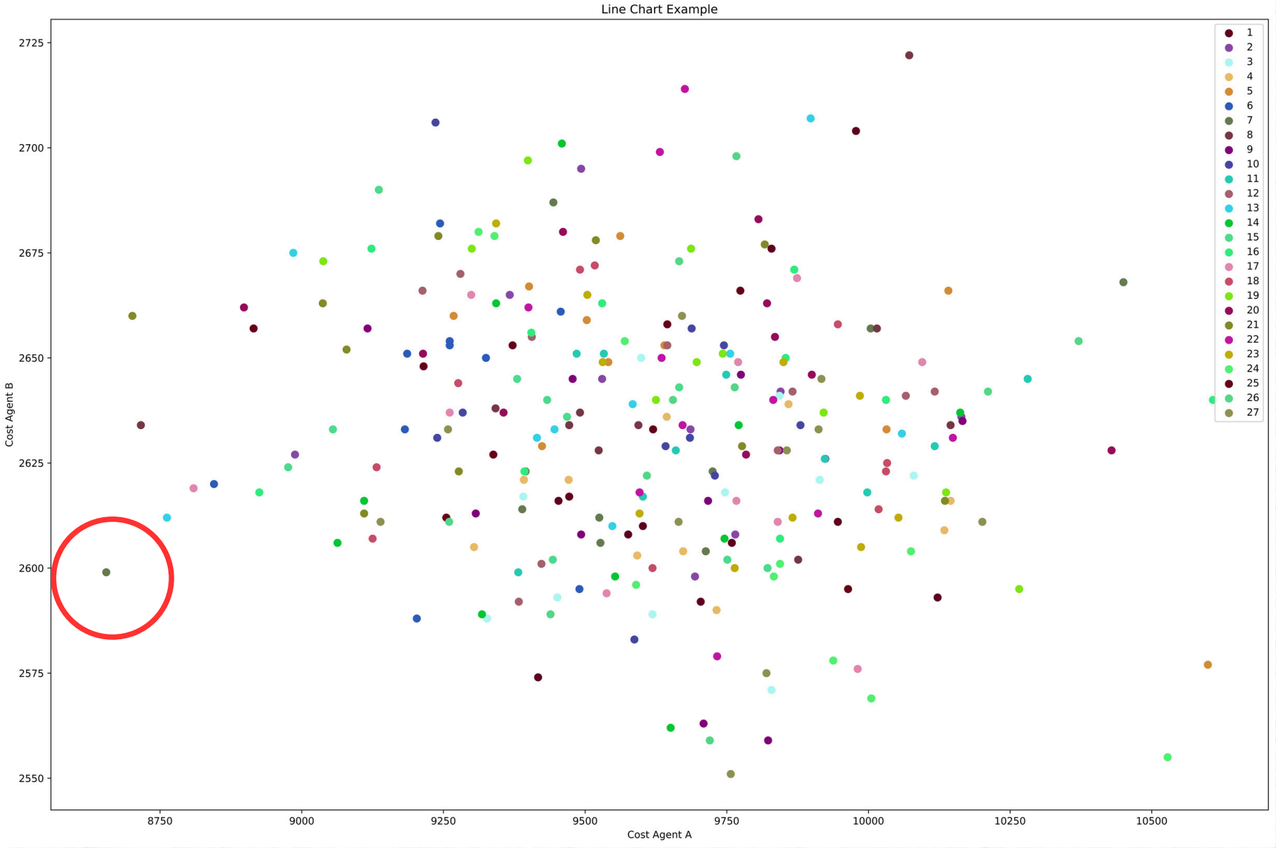
100 ROUNDS

1000 ROUNDS

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

10 ROUNDS RESULTS

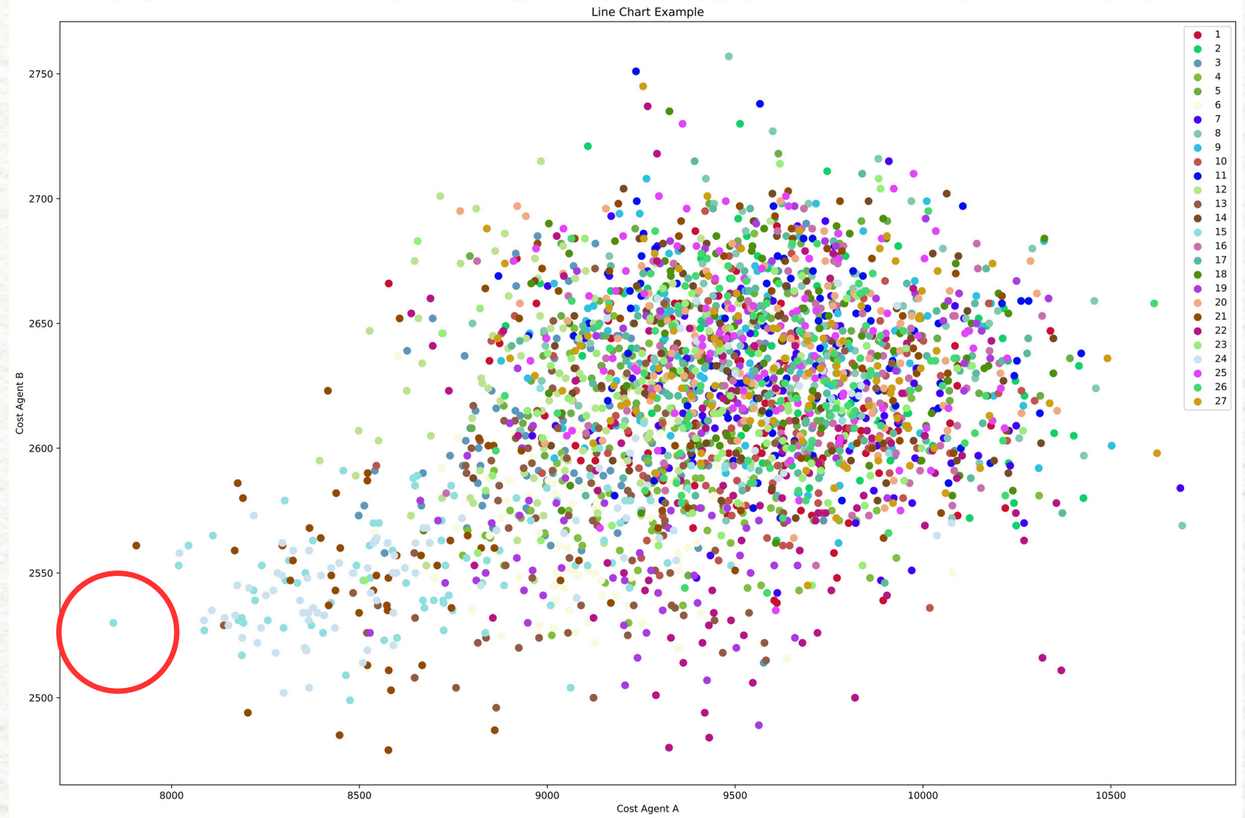
Color	Selection	Crossover	Mutation
	RANK	ORDER	INVERSE
	RANK	ORDER	SCRAMBLE
	RANK	ORDER	DISPLACEMENT
	RANK	CYCLE	INVERSE
	RANK	CYCLE	SCRAMBLE
	RANK	CYCLE	DISPLACEMENT
	RANK	POSITION_BASED	INVERSE
	RANK	POSITION_BASED	SCRAMBLE
	RANK	POSITION_BASED	DISPLACEMENT
	TOURNAMENT	ORDER	INVERSE
	TOURNAMENT	ORDER	SCRAMBLE
	TOURNAMENT	ORDER	DISPLACEMENT
	TOURNAMENT	CYCLE	INVERSE
	TOURNAMENT	CYCLE	SCRAMBLE
	TOURNAMENT	CYCLE	DISPLACEMENT
	TOURNAMENT	POSITION_BASED	INVERSE
	TOURNAMENT	POSITION_BASED	SCRAMBLE
	TOURNAMENT	POSITION_BASED	DISPLACEMENT
	TEMPERATURE	ORDER	INVERSE
	TEMPERATURE	ORDER	SCRAMBLE
	TEMPERATURE	ORDER	DISPLACEMENT
	TEMPERATURE	CYCLE	INVERSE
	TEMPERATURE	CYCLE	SCRAMBLE
	TEMPERATURE	CYCLE	DISPLACEMENT
	TEMPERATURE	POSITION_BASED	INVERSE
	TEMPERATURE	POSITION_BASED	SCRAMBLE
	TEMPERATURE	POSITION_BASED	DISPLACEMENT



Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

100 ROUND RESULTS

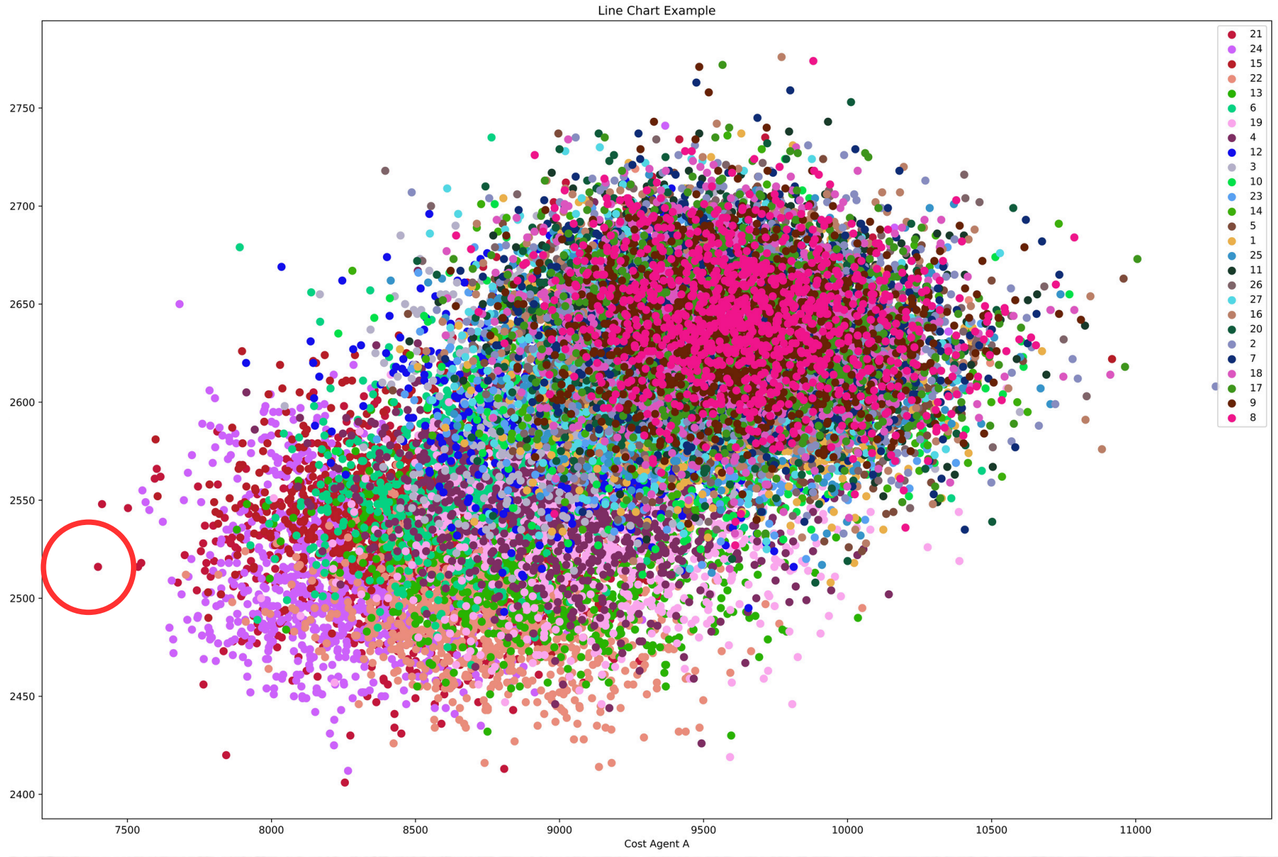
Color	Selection	Crossover	Mutation
	RANK	ORDER	INVERSE
	RANK	ORDER	SCRAMBLE
	RANK	ORDER	DISPLACEMENT
	RANK	CYCLE	INVERSE
	RANK	CYCLE	SCRAMBLE
	RANK	CYCLE	DISPLACEMENT
	RANK	POSITION_BASED	INVERSE
	RANK	POSITION_BASED	SCRAMBLE
	RANK	POSITION_BASED	DISPLACEMENT
	TOURNAMENT	ORDER	INVERSE
	TOURNAMENT	ORDER	SCRAMBLE
	TOURNAMENT	ORDER	DISPLACEMENT
	TOURNAMENT	CYCLE	INVERSE
	TOURNAMENT	CYCLE	SCRAMBLE
	TOURNAMENT	CYCLE	DISPLACEMENT
	TOURNAMENT	POSITION_BASED	INVERSE
	TOURNAMENT	POSITION_BASED	SCRAMBLE
	TOURNAMENT	POSITION_BASED	DISPLACEMENT
	TEMPERATURE	ORDER	INVERSE
	TEMPERATURE	ORDER	SCRAMBLE
	TEMPERATURE	ORDER	DISPLACEMENT
	TEMPERATURE	CYCLE	INVERSE
	TEMPERATURE	CYCLE	SCRAMBLE
	TEMPERATURE	CYCLE	DISPLACEMENT
	TEMPERATURE	POSITION_BASED	INVERSE
	TEMPERATURE	POSITION_BASED	SCRAMBLE
	TEMPERATURE	POSITION_BASED	DISPLACEMENT



Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

1000 ROUND RESULTS

Color	Selection	Crossover	Mutation
	TEMPERATURE	ORDER	DISPLACEMENT
	TEMPERATURE	CYCLE	DISPLACEMENT
	TOURNAMENT	CYCLE	DISPLACEMENT
	TEMPERATURE	CYCLE	INVERSE
	TOURNAMENT	CYCLE	INVERSE
	RANK	CYCLE	DISPLACEMENT
	TEMPERATURE	ORDER	INVERSE
	RANK	CYCLE	INVERSE
	TOURNAMENT	ORDER	DISPLACEMENT
	RANK	ORDER	DISPLACEMENT
	TOURNAMENT	ORDER	INVERSE
	TEMPERATURE	CYCLE	SCRAMBLE
	TOURNAMENT	CYCLE	SCRAMBLE
	RANK	CYCLE	SCRAMBLE
	RANK	ORDER	INVERSE
	TEMPERATURE	POSITION_BASED	INVERSE
	TOURNAMENT	ORDER	SCRAMBLE
	TEMPERATURE	POSITION_BASED	SCRAMBLE
	TEMPERATURE	POSITION_BASED	DISPLACEMENT
	TOURNAMENT	POSITION_BASED	INVERSE
	TEMPERATURE	ORDER	SCRAMBLE
	RANK	ORDER	SCRAMBLE
	RANK	POSITION_BASED	INVERSE
	TOURNAMENT	POSITION_BASED	DISPLACEMENT
	TOURNAMENT	POSITION_BASED	SCRAMBLE
	RANK	POSITION_BASED	DISPLACEMENT
	RANK	POSITION_BASED	SCRAMBLE



Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

RESULT AND FINDINGS

1000 Rounds	daten3ASupplier_200	daten3BSupplier_200	daten4ASupplier_200	daten4BSupplier_200
daten3ACustomer_200_10	CostA, CostB (7398 , 2516) TEMPERATURE ORDER DISPLACEMENT	CostA, CostB (7018, 2502) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (14888,2442) TEMPERATURE CYCLE INVERSE	CostA, CostB (14827,2512) TEMPERATURE CYCLE DISPLACEMENT
daten3BCustomer_200_20	CostA, CostB (7369,6601) TEMPERATURE CYCLE INVERSE	CostA, CostB (7464,6549) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (14880,6514) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (14683,6702) TEMPERATURE CYCLE DISPLACEMENT
daten4ACustomer_200_5	CostA, CostB (7424,982) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (7276,969) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (14833,972) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (14438,988) TEMPERATURE ORDER DISPLACEMENT
daten4BCustomer_200_5	CostA, CostB (7525,8959) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (7317,9119) TEMPERATURE CYCLE DISPLACEMENT	CostA, CostB (14787,9126) TEMPERATURE CYCLE INVERSE	CostA, CostB (14716,8931) TEMPERATURE CYCLE DISPLACEMENT

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

CONCLUSION

- **Larger populations** increase diversity and improve chances of finding optimal inputs.
- **Dummy contract** speeds up fitness evaluation and guides convergence.
- **27 methods** combinations across selection, crossover, and mutation tested.
- Best performance: **Temperature Selection + Cycle Crossover + Displacement Mutation** (1000 rounds).
- **Visualization** helps interpret results and identify **near–Pareto optimal solutions.**

Kang Chun-Yu Nicoló Rossi Lin Chin-Jung

THANK YOU


100
111
001
Selection
000
011
110
Crossover
000
001
111
Mutation
TECNOLÓGGIA