Travelling Salesman Problem

With Genetic Algorithm

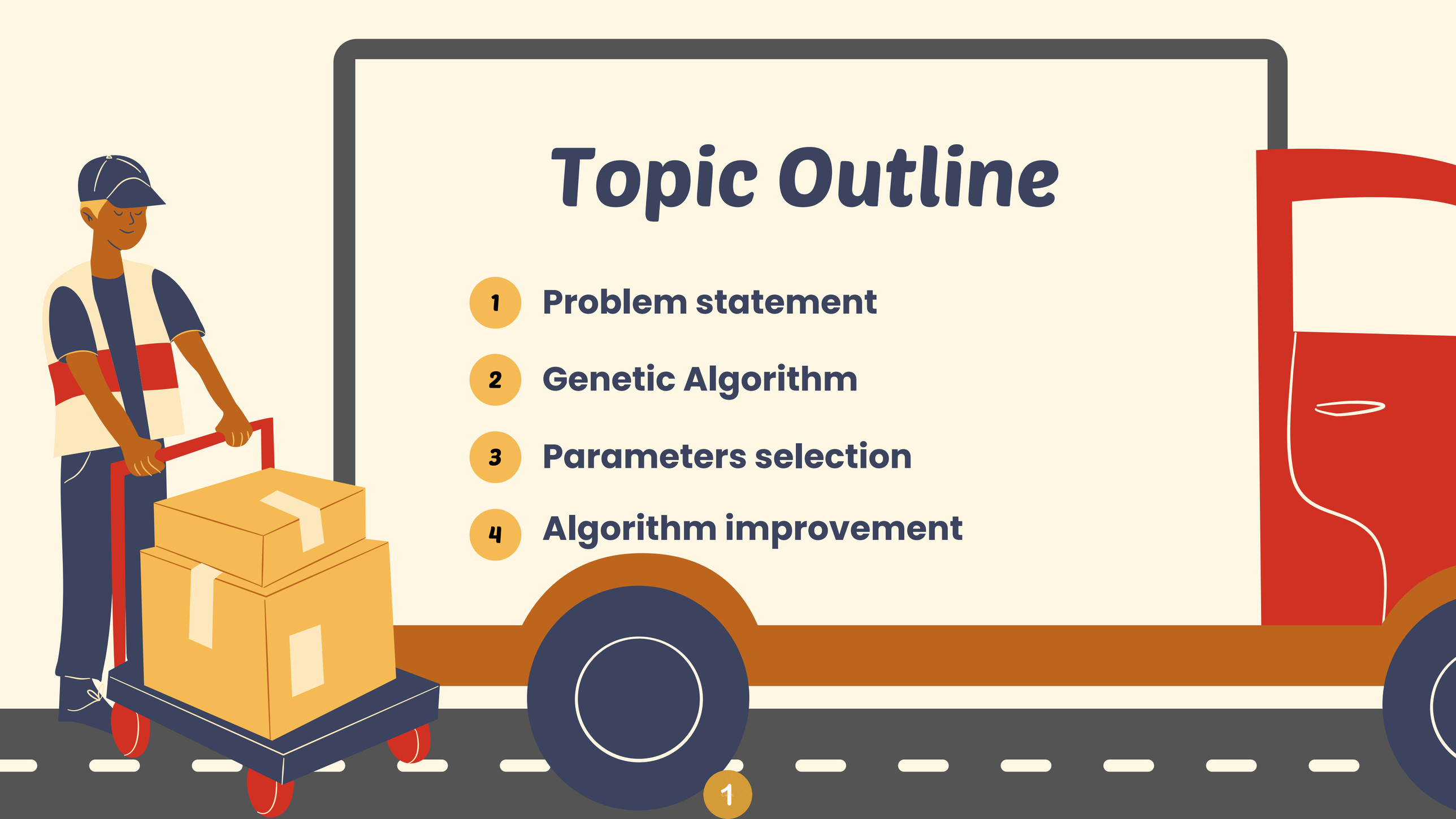
Presented by: Manel Naffati - Hammada Lekehal

3AGI1 - DS

Academic year 2024-2025

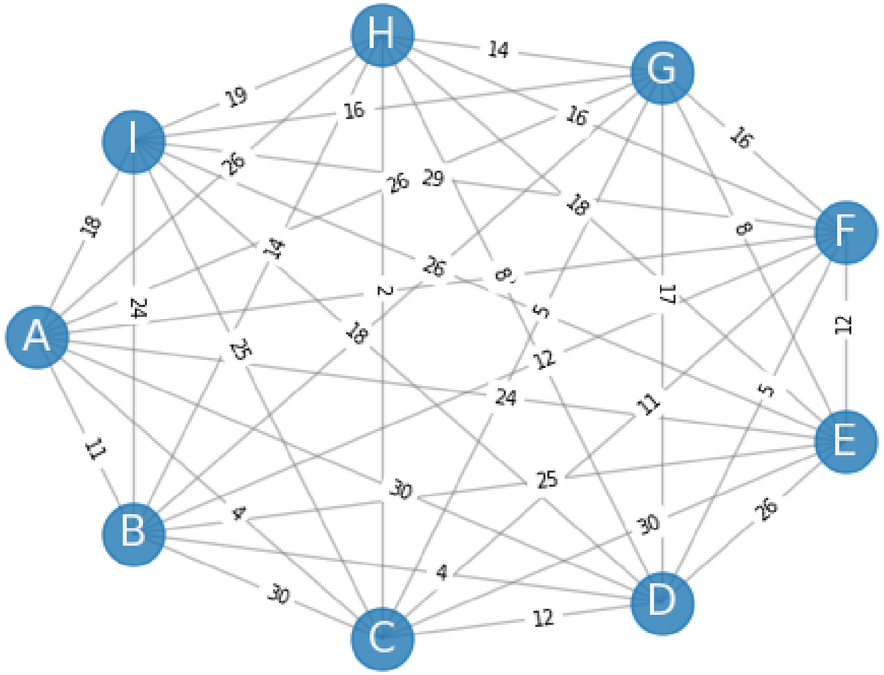
Topic Outline

1. Problem statement
2. Genetic Algorithm
3. Parameters selection
4. Algorithm improvement



Problem statement

There is a salesman who must visit a set of cities exactly once and return to the starting city, with the objective of minimizing the total travel distance or cost.



Problem statement

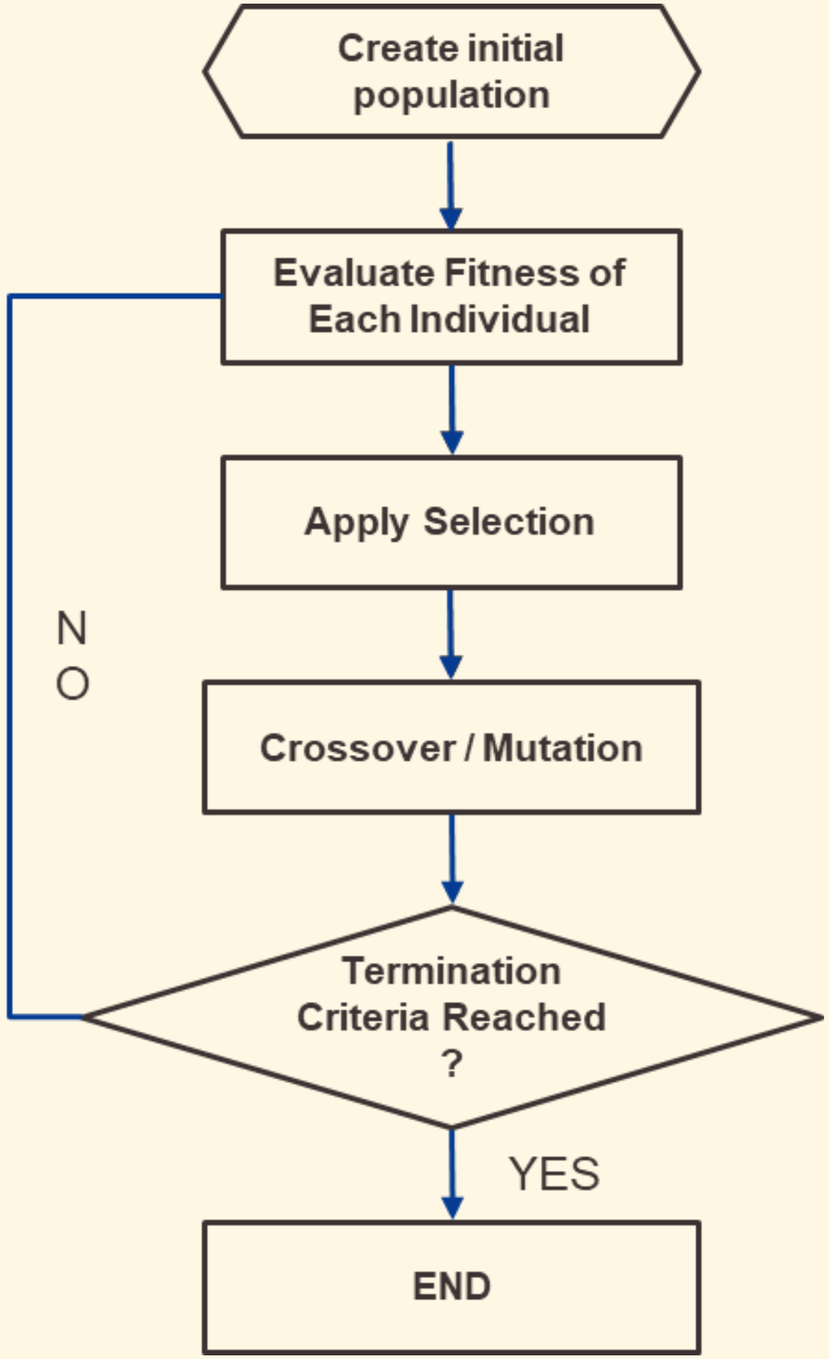
The problem is NP-Hard

linear programming does not work

Which algorithm we will use ?

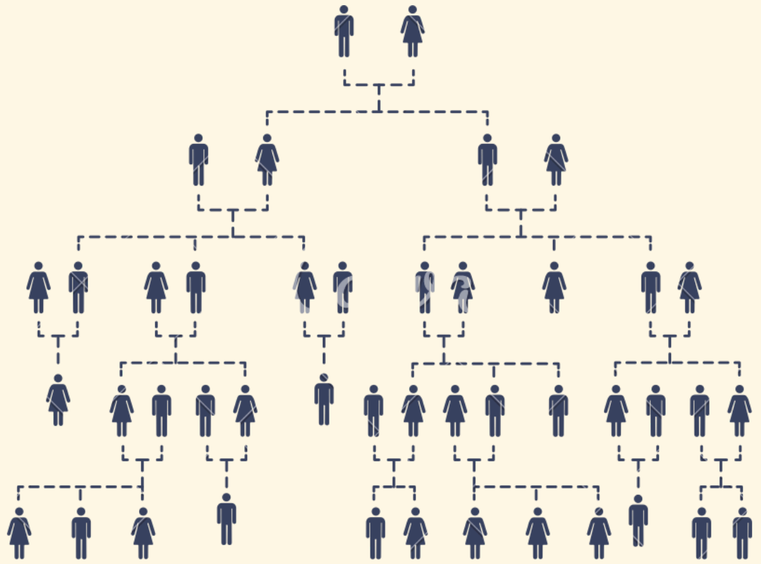
Genetic Algorithm

In AI, GA are metaheuristic algorithms inspired by Darwin's theory of evolution in biological systems, following the principle of survival of the fittest.



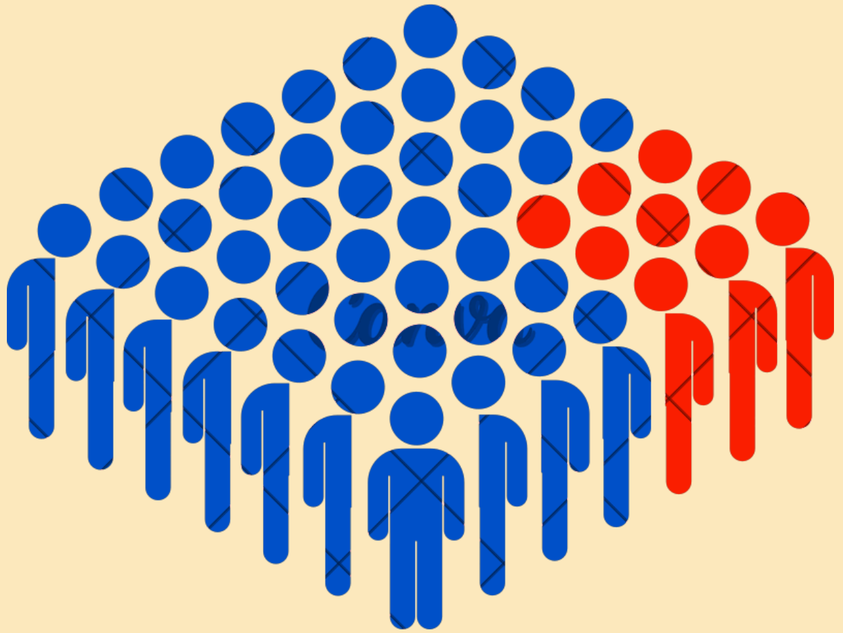
GA Parameters

Number of generations



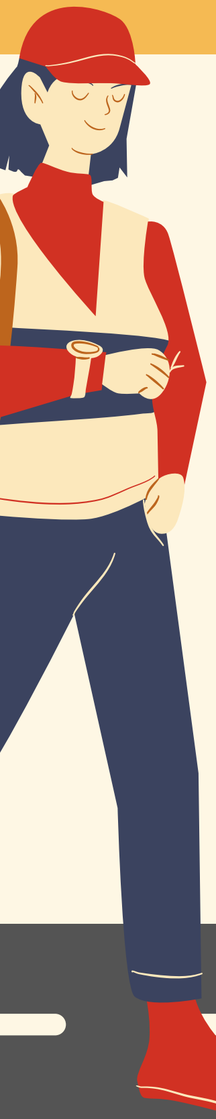
The number of iterations (generations) the GA runs before stopping.

Population size



The number of individuals (candidate solutions) in each generation.

GA refrence Parameters



Number of generations

=

250

Population size

=

300

Reference Solution

Instances	Number of cities	Cost	CPU time
1	12	74.0	7.96
2	6	38.0	4.92
3	8	65.0	5.1993
4	14	87.0	07.04
5	15	115.0	10.10
6	9	58.0	8.35
		CPU time Average	7,38 s

Variable parameter : Number of generation

Instances	Number of cities	cost de référence	cost	GAP	CPU_time (s)
1	12	74.0	59.0	-20.27%	2.75
2	6	38.0	44.0	15.79%	1.67
3	8	65.0	85.0	30.77%	2.625
4	14	87.0	70.0	-19.54%	3.68
5	15	115.0	100.0	-13.04%	3.5
6	9	58.0	57.0	-1.75%	2.45
			Average	16.86%	2.78s

Number of generation = 150

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	70.0	-5.41%	4.578
2	6	38.0	50.0	31.58%	3.48
3	8	65.0	76.0	16.92%	2.51
4	14	87.0	97.0	11.49%	3.67
5	15	115.0	95.0	-17.39%	4.82
6	9	58.0	46.0	-20.69%	2.26
			Average	17.24%	3.56s

Number of generation = 300

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	66.0	-10.81%	5.82
2	6	38.0	53.0	39.47%	04.03
3	8	65.0	89.0	36.92%	4.98
4	14	87.0	80.0	-0,08	5.70
5	15	115.0	79.0	-31.3%	6.46
6	9	58.0	51.0	-12.06%	4.57
			Average	23,10%	5,26s

Number of generation = 200

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	95.0	28.37%	6.62
2	6	38.0	48.0	26.31%	4.31
3	8	65.0	60.0	-7.7%	05.01
4	14	87.0	107.0	0,23	7.53
5	15	115.0	80.0	-30.4%	9.79
6	9	58.0	72.0	24.14%	5.45
			Average	10.62%	6.47s

Number of generation = 350

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	97.0	31.1 %	7.17
2	6	38.0	83.0	118 %	6.68
3	8	65.0	35.0	-46.15%	06.01
4	14	87.0	74.0	-14.9%	8.5
5	15	115.0	109.0	-5.2%	7.96
6	9	58.0	55.0	-5.17%	6.81
			Average	12.95%	7.19s

Number of generation = 400

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	63.0	-14.86%	7.85
2	6	38.0	57.0	0,5	5.23
3	8	65.0	77.0	18.46%	6.37
4	14	87.0	83.0	-4.6%	8.390
5	15	115.0	82.0	-28.7%	12.125
6	9	58.0	74.0	27.6%	9.76
			Average	24,03%	8,29s

Number of generation = 450

Population size fix to 300 & Number of generation variable						
Number of generation	150	200	300	350	400	450
GAP Average	16.86%	17.24%	23,10%	10.62%	12.95%	24,03%
CPU_time Average	2.78s	3.56s	5,26s	6.47s	7.19s	8,29s

The best number of generation is 350

Variable parameter : Population size

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	72.0	-2.7%	3.65
2	6	38.0	44.0	15.79%	3.31
3	8	65.0	78.0	0,2	3.75
4	14	87.0	90.0	3.44%	3.84
5	15	115.0	98.0	-14.78%	3.98
6	9	58.0	69.0	18.96%	1.98
			Average	12.61%	3.42s

Population size = 200

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	109.0	47.3%	4.84
2	6	38.0	52.0	36.84%	2.20
3	8	65.0	43.0	-33.85%	3.53
4	14	87.0	89.0	2.3%	4.43
5	15	115.0	96.0	-16.52%	4.29
6	9	58.0	68.0	17.24%	2.75
			Average	5.135%	3.68s

Population size =250

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	72.0	-2.7%	06.01
2	6	38.0	66.0	73.7%	4.82
3	8	65.0	68.0	4.6%	4.65
4	14	87.0	86.0	-1.15%	5.87
5	15	115.0	64.0	-44.34%	5.76
6	9	58.0	73.0	25.86%	3.25
			Average	9.33%	5.1s

Population size = 350

Instances	Number of cities	refernce cost	cost	GAP	CPU_time (s)
1	12	74.0	74.0	0	6.140
2	6	38.0	52.0	36.8%	4.17
3	8	65.0	79.0	21.53%	5.125
4	14	87.0	105.0	20.6%	6.95
5	15	115.0	71.0	-38.26%	8.23
6	9	58.0	77.0	32.75%	4.81
			Average	24,99%	5,24s

Population size = 400

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	84.0	13,51%	8.45
2	6	38.0	58.0	52,60%	6.70
3	8	65.0	64.0	-1,54%	5.73
4	14	87.0	78.0	-10,34%	7.57
5	15	115.0	74.0	-35,65%	8.35
6	9	58.0	45.0	-22,41%	6.70
			Average	22,68%	7,25 s

Population size = 450

Instances	Number of cities	reference cost	cost	GAP	CPU_time (s)
1	12	74.0	76.0	2,70%	8.48
2	6	38.0	58.0	52,63%	5.421
3	8	65.0	57.0	-12,30%	6.84
4	14	87.0	75.0	-13,80%	10.40
5	15	115.0	99.0	-13,91%	10.31
6	9	58.0	73.0	25,86%	6.79
			Average	6.86%	8.04s

Population size = 500

Number of generation fix to 250 & Population size variable						
Population size	200	250	350	400	450	500
GAP Average	12.61%	5.135%	9.33%	24,99%	22,68%	6.86%
CPU_time Average	3.42s	3.68s	5.1s	5,24s	7,25 s	8.04s

The best population size is 250

The best parameters

Number of generation
=
350

Population size
=
250

Reference Solution

Instances	Number of cities	Cost	CPU time
1	12	74.0	7.96
2	6	38.0	4.92
3	8	65.0	5.1993
4	14	87.0	07.04
5	15	115.0	10.10
6	9	58.0	8.35
		CPU time Average	7,38 s

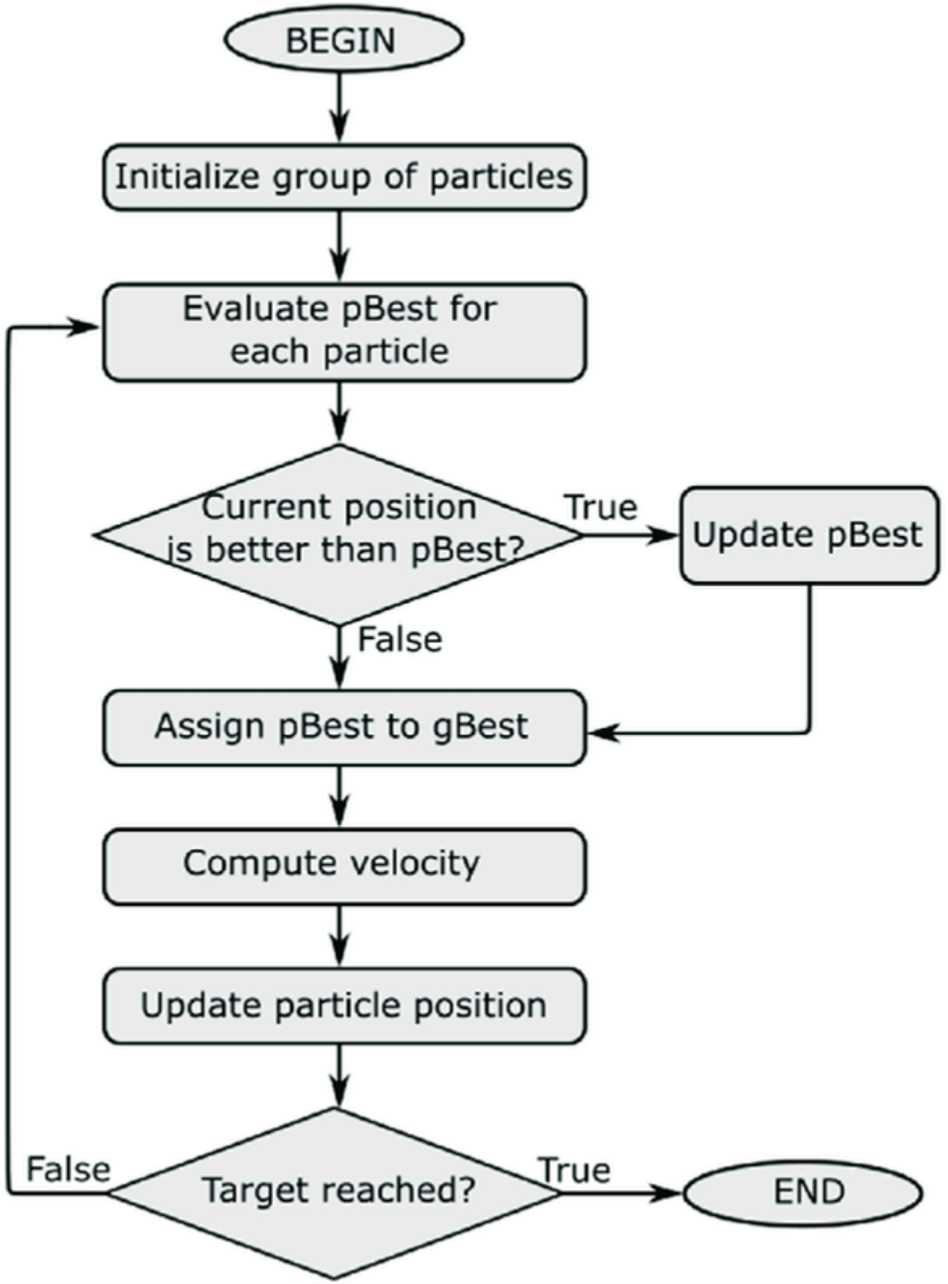
Result with The best parameters

Instances	Number of cities	cost	reference cost	GAP	CPU_time (s)
1	12	97.0	74.0	31%	7.25
2	6	72.0	38.0	89%	3.3125
3	8	68.0	65.0	4.6%	4.0625
4	14	94.0	87.0	8%	5.828
5	15	116.0	115.0	0.87%	6.2019
6	9	81.0	58.0	39.65%	4.59
			Average	28.85%	5,17s

Algorithm improvement

Combining Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)

PSO is an optimization algorithm where particles explore the search space by following their own and the group's best positions.



Algorithm improvement

Combining Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)

Step1

PSO Particle Initialization

Step2

PSO Particle Update

Step3

GA + PSO Hybridization

Algorithm improvement

Combining Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)

Instances	Number of cities	cost	reference cost	GAP	CPU_time (s)
1	12	109.0	74.0	47.29%	36.12
2	6	37.0	38.0	-2.63%	31.39
3	8	61.0	65.0	-6.15%	22.94
4	14	95.0	87.0	9.2%	18.39
5	15	155.0	115.0	34.78%	11.28
6	9	67.0	58.0	15.52%	05.03
			Average	19.26%	20.86s

Thank you for listening!

Don't hesitate to ask any questions!