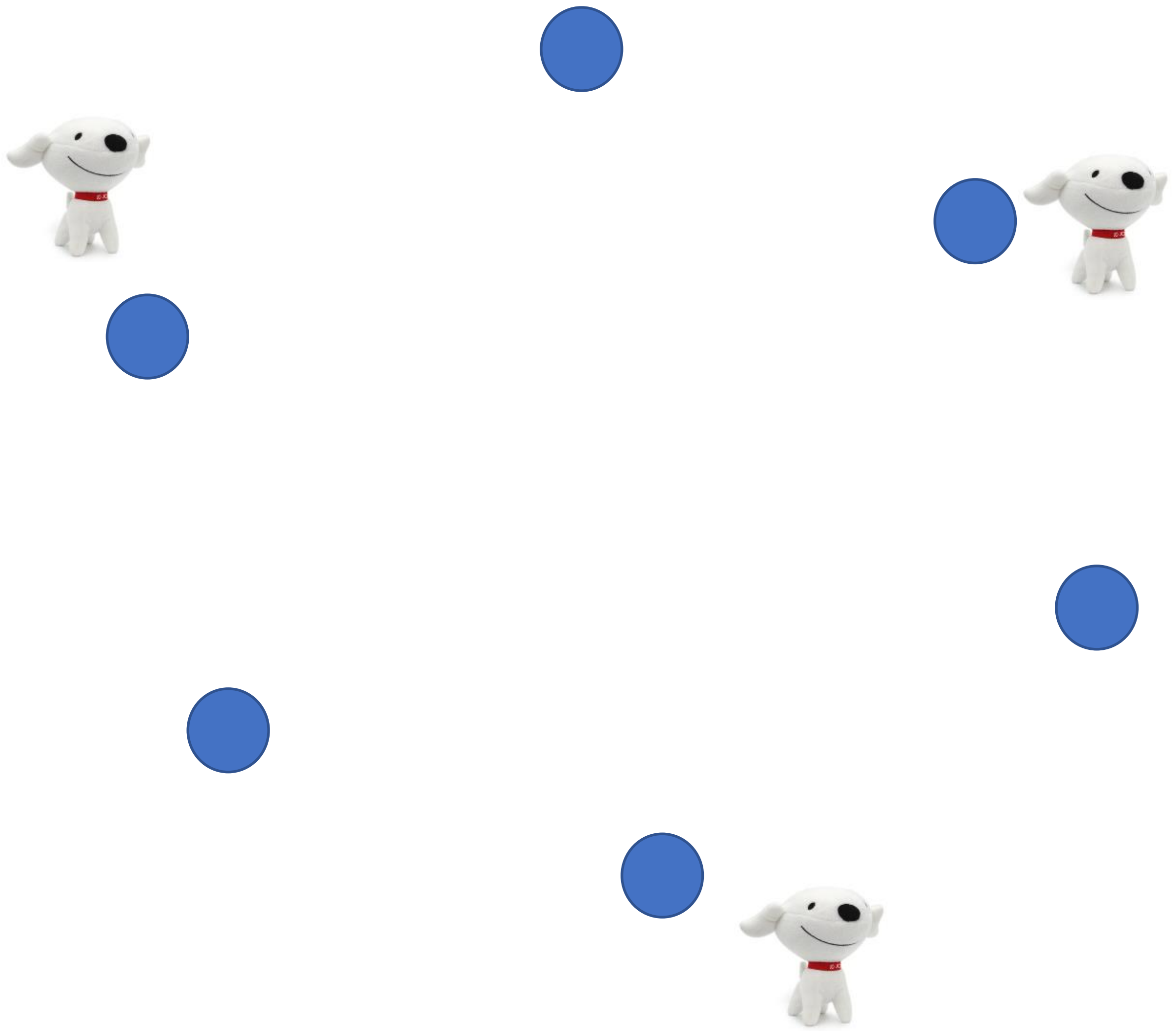
Reliable Facility Location Problem

——Wenxing Lan

Outline:

- Introduction to RFLP
- EA with Memorable Local Search (EAMLS)
- Reproduction Result of EAMLS
- Some Ideas
- Conclusion

Introduction to RFLP:

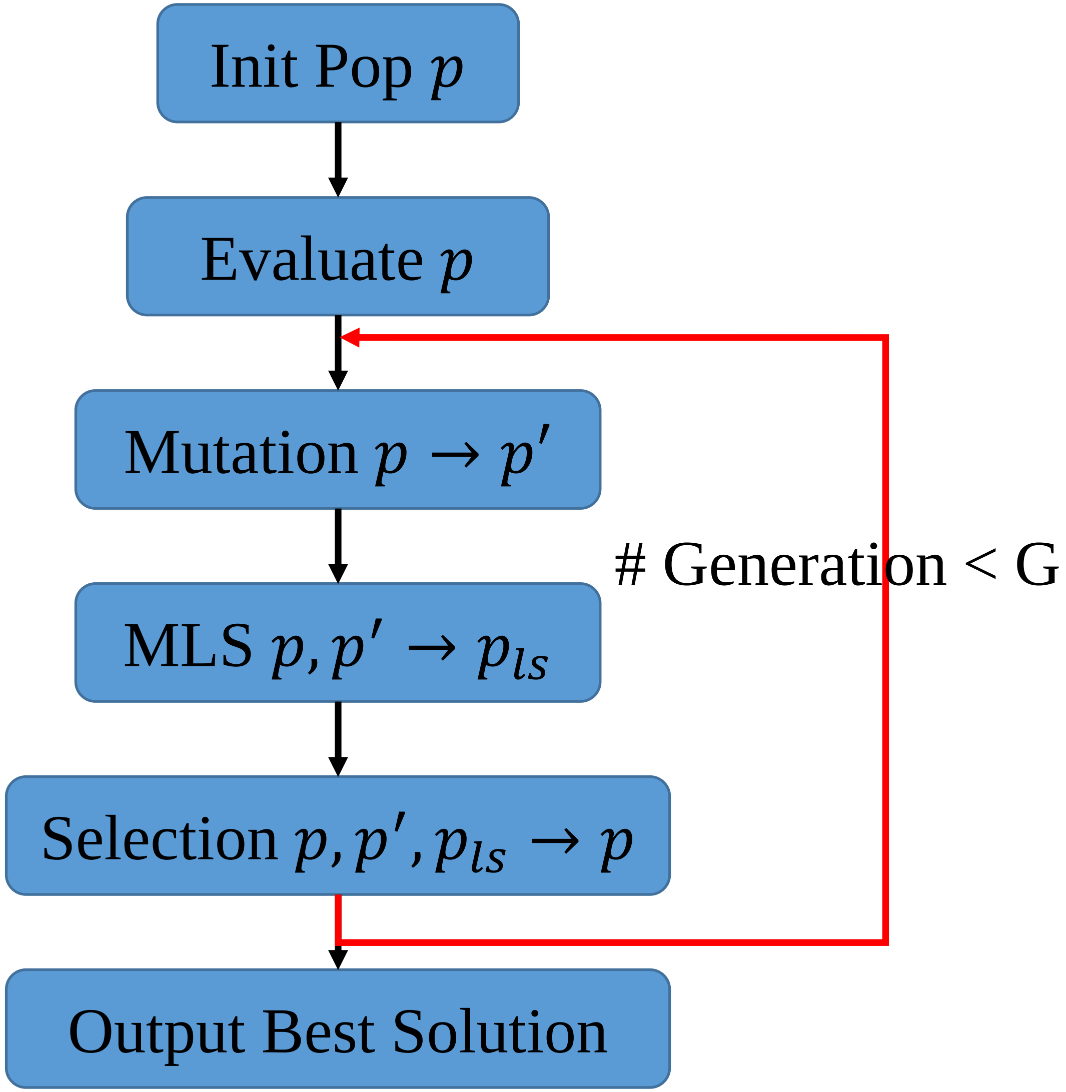


Reliable?

⇒# Candidate Facility fixed or not

- ➢ # Candidate Facility
- ➢ Position of Candidate Facility

EAMLS:



Initialization Method:

- Stochastic initialization
- Binary Representation
- Every gene of an individual takes 0 or 1 with equal probability

Memorable Local Search:

- Do local search for the individuals which have not been search before
- At most do local search for n individual each generation

Dynamic Population Size:

- Change p_{size} with the $l3\_value$.
- $p_{size} \mathrel{+}= step\_size$

[1] H. Zhang, J. Liu, and X. Yao, “A hybrid evolutionary algorithm for reliable facility location problem,” in Parallel Problem Solving from Nature – PPSN XVI, T. B¨ack, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann, Eds. Cham: Springer International Publishing, 2020, pp. 454–467.

Reproduction Result of EAMLS:

Runtime Environment:

All programs have been written in C++ 11 and executed on an Intel(R) Core(TM) i5-10400F CPU working at 2.90 GHz on Windows 10 20H2, using a single thread.

Parameters Setting (Same as ones in [1]):

Parameters	Value
Mutation Rate, m	0.1
# Local search individual, n	10
$l3$-value threshold, β	0.8
Step size of population	100

Instance Scale (# nodes)	# Generation	Population Size
10	10	20
50	20	20
100	50	100

Wilcox Sign Rank test is done with the level of significance 0.05.

[1] H. Zhang, J. Liu, and X. Yao, “A hybrid evolutionary algorithm for reliable facility location problem,” in Parallel Problem Solving from Nature – PPSN XVI, T. B¨ack, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann, Eds. Cham: Springer International Publishing, 2020, pp. 454–467.

Reproduction Result of EAMLS: 10 nodes instances:

Instance No.	**My Implementation**			**Hu Zhang's Implementation**		
	AVERAGE	STD	BEST	AVERAGE	STD	BEST
0	3346.929	2.27×10^{-12}	3346.929	3346.929	2.27×10^{-12}	3346.929
1	2608.603	0	2608.603	2608.603	0	2608.603
2	2381.656	4.55×10^{-13}	2381.656	2381.656	4.55×10^{-13}	2381.656
3	3104.342	4.55×10^{-13}	3104.342	3104.342	4.55×10^{-13}	3104.342
4	3063.061	0	3063.061	3063.061	0	3063.061
5	2258.037	9.09×10^{-13}	2258.037	2258.037	9.09×10^{-13}	2258.037
6	2369.84	0	2369.84	2369.84	0	2369.84
7	1808.556	0	1808.556	1808.556	0	1808.556
$+/-/\approx$		/	/	0/0/8	/	/

Reproduction Result of EAMLS: 50 nodes instances:

Instance No.	My Implementation			Hu Zhang's Implementation			GAP	
	AVERAGE	STD	BEST	AVERAGE	STD	BEST	AVERAGE %	BEST %
0	7256.336	288.57	6857.798	6814.142*	4.5×10^{-12}	6814.142	6.09	0.64
1	7840.407	187.00	7556.475	7514.328*	7.83	7512.875	4.16	0.58
2	7369.272	174.30	7098.768	7083.504*	23.58	7073.701	3.88	0.35
3	8030.533	160.82	7721.426	7633.463*	37.30	7625.132	4.94	1.25
4	8557.142	228.39	8225.232	8108.956*	21.50	8103.476	5.24	1.48
5	8094.892	195.48	7687.733	7689.739*	10.80	7687.733	5.01	0
6	8197.092	173.14	7890.151	7782.568*	25.05	7772.954	5.06	1.49
7	7086.206	167.17	6796.706	6799.642*	15.81	6796.706	4.04	0
+/−/≈		/	/	8/0/0	/	/		

Idea 1: Change Initialization

Initialization Method in [1]:

- ➢ Stochastic initialization
- ➢ Binary Representation
- ➢ Every gene of an individual takes 0 or 1 with equal probability

$\Rightarrow$ E(# candidate facility) $= \frac{1}{2} \times$ # nodes

- ➢ # Candidate Facility
- ➢ Position of Candidate Facility

Change: Make # candidate facility more diversity

nodes $\leq \mu$

按照$m \in \{2,3,\ldots,\#node\}$生成facility位置随机的个体，剩余部分的个体按照m = $uniform[2,\#node]$生成facility位置随机的个体

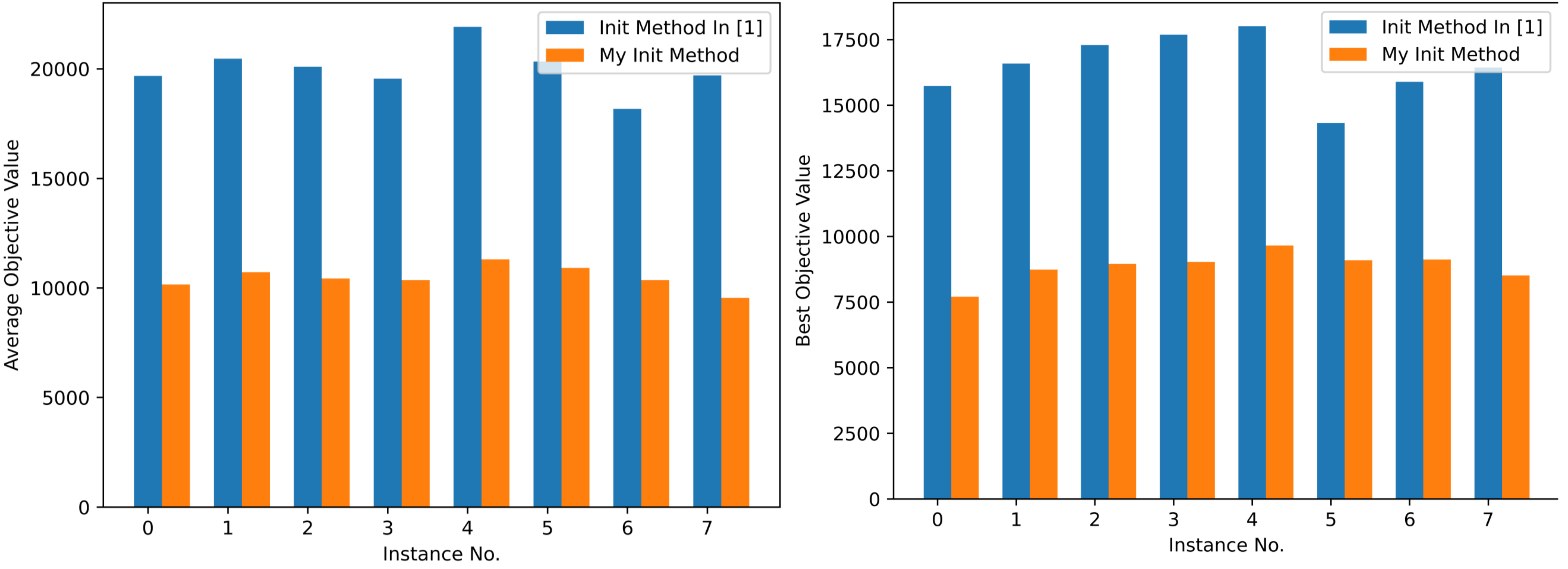
$\mu <$# nodes $< 2\mu$

按照$m \in \{2,4,\ldots,2\lfloor\frac{\#node}{2}\rfloor\}$生成facility位置随机的个体，剩余部分的个体按照m = $uniform[2,\#node]$生成facility位置随机的个体

nodes $\geq 2\mu$

$a = \left\lfloor\frac{\#node}{\mu}\right\rfloor$,按照$m \in \{2,2+a,2+2a,\ldots\}(m \leq \#node)$生成facility位置随机的个体，剩余部分的个体按照m = $uniform[2,\#node]$生成facility位置随机的个体

Experimental Result: 50 node instances:



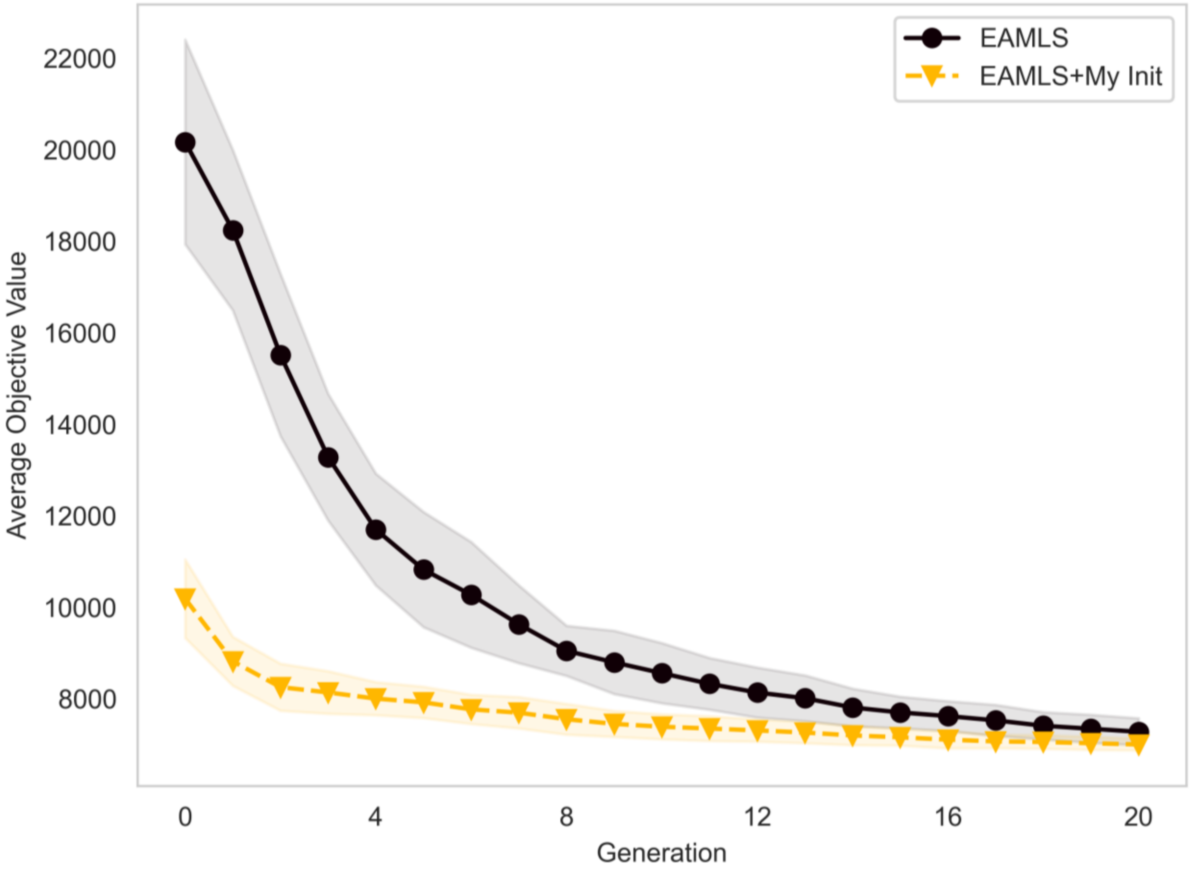
[1] H. Zhang, J. Liu, and X. Yao, “A hybrid evolutionary algorithm for reliable facility location problem,” in Parallel Problem Solving from Nature – PPSN XVI, T. B¨ack, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann, Eds. Cham: Springer International Publishing, 2020, pp. 454–467.

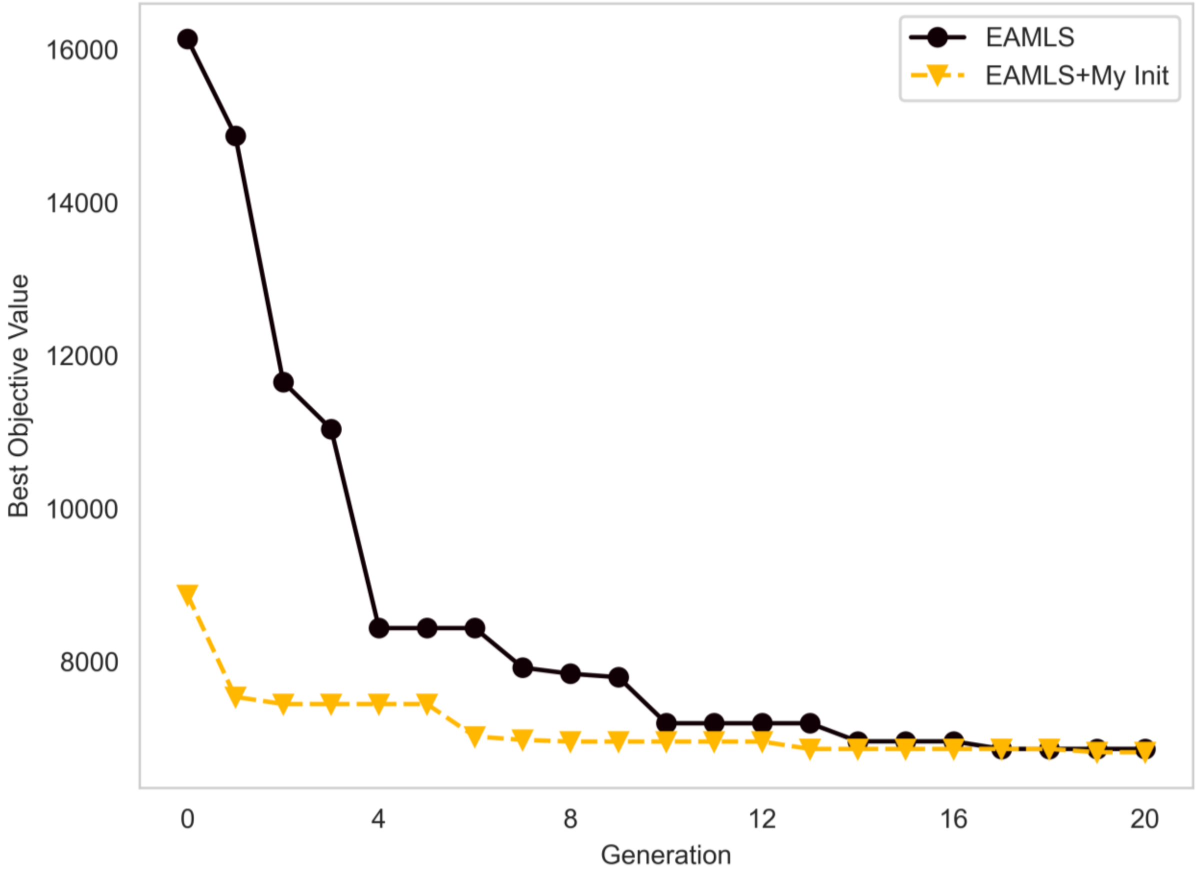
Experimental Result:

50 nodes instances:

Instance No.	EAMLS			EAMLS + My Init			GAP	
	AVERAGE	STD	BEST	AVERAGE	STD	BEST	AVERAGE %	BEST %
0	7298.548	240.96	6857.798	6989.502*	117.23	6814.142	4.42	0.64
1	7844.357	279.38	7556.475	7665.536*	118.26	7512.875	2.33	0.58
2	7399.545	169.24	7126.035	7207.391*	94.69	7073.701	2.67	0.74
3	8127.141	179.49	7759.225	7799.453*	122.19	7625.132	4.2	1.76
4	8566.518	279.54	8103.476	8312.809*	114.16	8160.579	3.05	-0.7
5	8054.054	210.78	7687.733	7845.046*	123.4	7687.733	2.66	0
6	8229.953	198.53	7866.683	7992.135*	147.66	7772.954	2.98	1.21
7	7162.556	197.19	6884.774	7004.941*	104.94	6834.758	2.25	0.73
+/−/≈		/	/	8/0/0	/	/		

Experimental Result: 50_0 instance:





Idea 2: Change Repair Strategy

Repair Strategy in [1]:

- check every gene in ascending order of fixed cost
- change the gene with 0-value to 1 until the individual satisfies the constraint $m \geq 2$

$$\min \boxed{\sum_{j \in J} f_j X_j} + \alpha \boxed{\sum_{i \in \mathrm{I}} \sum_{j \in J} \sum_{r=0}^{m-1} h_i c_{i,j} p^r (1-p) Y_{ijr}}$$

Change: ascending order of Fixed cost + $\sum_{i \in I} \sum_{j \in J} h_i c_{i,j}$

Experimental Result:

50 nodes instances:

Instance No.	EAMLS			EAMLS + My Repair			GAP	
	AVERAGE	STD	BEST	AVERAGE	STD	BEST	AVERAGE %	BEST %
0	7298.548	240.96	6857.798	7230.945	264.76	6814.142	0.93	0.64
1	7844.357	279.38	7556.475	7830.142	143.86	7560.287	0.18	-0.05
2	7399.545	169.24	7126.035	7460.267	229.37	7146.44	-0.81	-0.29
3	8127.141	179.49	7759.225	8037.818*	137.05	7794.663	1.11	-0.45
4	8566.518	279.54	8103.476	8509.999	200.7	8103.476	0.66	0
5	8054.054	210.78	7687.733	7988.162	204.25	7687.733	0.82	0
6	8229.953	198.53	7866.683	8130.014	211.05	7772.954	1.23	1.21
7	7162.556	197.19	6884.774	7170.06	175.94	6834.758	-0.1	0.73
+/−/≈		/	/	1/0/7	/	/	/	/

Idea 3: Change Local Search

Neighborhood in [1]:

The set of individuals whose Hamming distance is 1 from that individual

1,0,0,1,1,0,1 $\Rightarrow$ 0,0,0,1,1,0,1

1,1,0,1,1,0,1

… …

Add: Same m value, but different position

1,0,0,1,1,0,1 $\Rightarrow$ 0,1,0,1,1,0,1

0,0,1,1,1,0,1

… …

Experimental Result:

50 nodes instances:

Instance No.	EAMLS			EAMLS + My Neighborhood Search			GAP	
	AVERAGE	STD	BEST	AVERAGE	STD	BEST	AVERAGE %	BEST %
0	7298.548	240.96	6857.798	7218.511	209.65	6956.724	1.11	-1.42
1	7844.357	279.38	7556.475	7733.453	170.61	7512.875	1.43	0.58
2	7399.545	169.24	7126.035	7434.712	211.93	7173.969	-0.47	-0.67
3	8127.141	179.49	7759.225	7989.844*	198.11	7625.132	1.72	1.76
4	8566.518	279.54	8103.476	8479.381	185.62	8103.476	1.03	0
5	8054.054	210.78	7687.733	8074.323	232.29	7747.924	-0.25	-0.78
6	8229.953	198.53	7866.683	8135.349*	166.08	7896.164	1.16	-0.37
7	7162.556	197.19	6884.774	7146.154	167.23	6834.758	0.23	0.73
+/−/≈		/	/	2/0/6	/	/	/	/

Conclusion:

Instance No.	Avg # Repair
0	0
1	0
2	0
3	4
4	0
5	0
6	0
7	0

- Change Initialization
  - Can get better initial population than the initialization method in [1]
- Change Repair Strategy
  - Add more computation
  - Performance is poor. ⇒ The number of repair operation is less when # nodes is large.
- Add Local Search
  - Add more computation
  - Performance is poor.
    - The neighborhood added by me may have been cover in other operators.