Efficient Extended Ant Colony Optimization for Capacitated Electric Vehicle Routing

2022.12

Bo-Cheng Lin[1], Xiao-Fang Liu*[1, 2], Yi Mei[3]

[1]Institute of Robotics and Automatic Information System, College of Artificial Intelligence, Nankai University, Tianjin 30050, China

[2]Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 30050, China

[3]School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62103202. (Corresponding author: Xiao-Fang Liu)

Electric Vehicle Routing

- Strict CO_2 emissions targets
- Demand from markets
- Compulsory phasing out of traditional vehicles
- Long charging time
- Less charging and swapping stations

Variants of EVRP

- Capacitated EVRP (CEVRP)
- EVRP with partial recharging (EVRPPR)
- EVRP with time window and recharging stations (EVRPTW)

...

Variants of EVRP

- Capacitated EVRP (CEVRP)
- EVRP with partial recharging (EVRPPR)
- EVRP with time window and recharging stations (EVRPTW)

...

have not been further studied for the complicated constraints in the new problems

Variants of EVRP

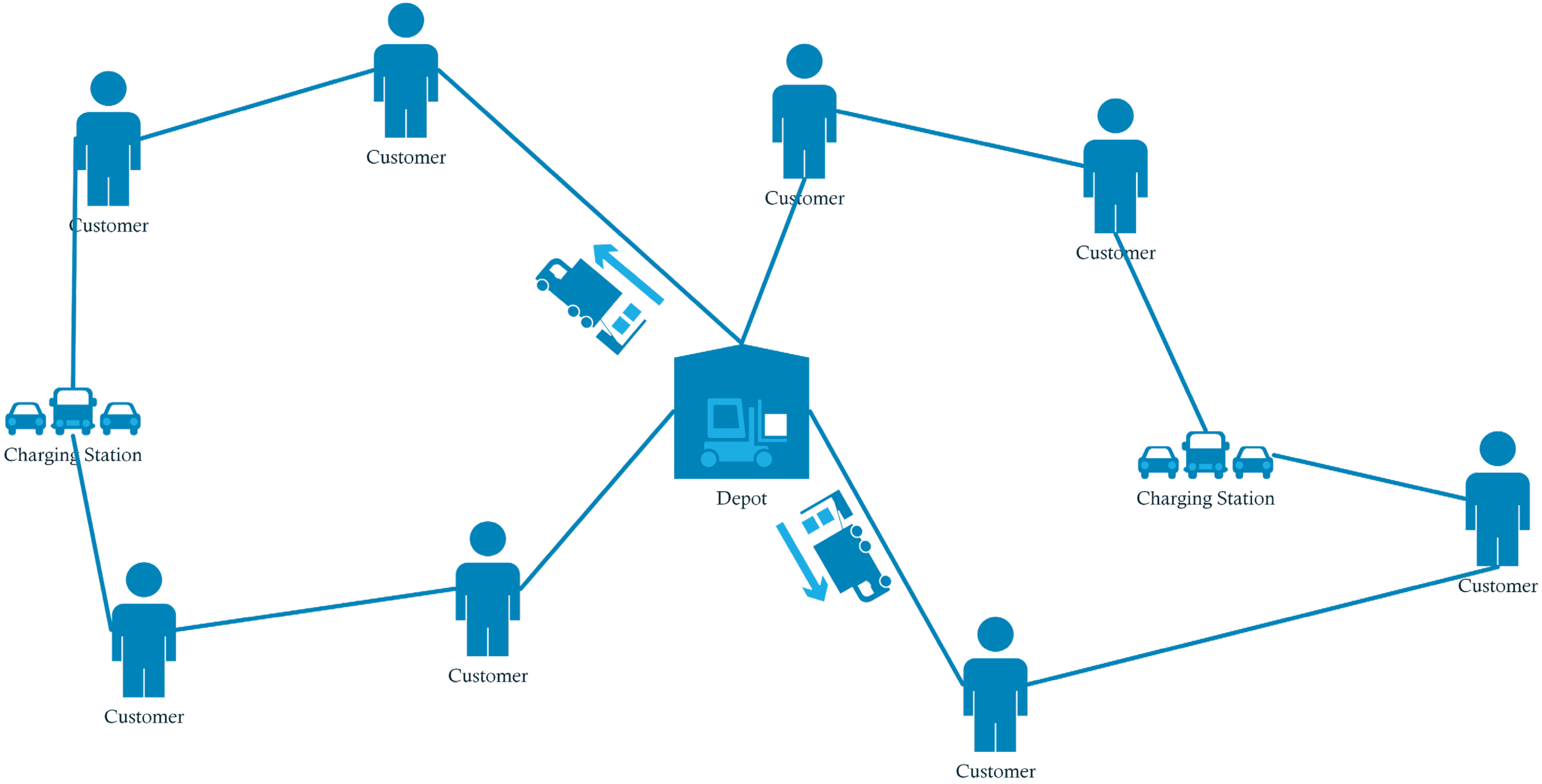
- **Capacitated EVRP (CEVRP)**

  the fundamental problem of other variants

- EVRP with partial recharging (EVRPPR)
- EVRP with time window and recharging stations (EVRPTW)

...

Capacitated Electric Vehicle Routing



Solving CEVRPs

Exact algorithms

- **mixed integer linear programming (MILP)**

  may fail on large-scale instances

Solving CEVRPs

Exact algorithms

- mixed integer linear programming (MILP)

    may fail on large-scale instances

Heuristic algorithms

- Variable neighborhood search (VNS)
- Iterative local search (ILS)

    poor exploration ability, easily trapped into local optima

Solving CEVRPs

Exact algorithms

- mixed integer linear programming (MILP)

  may fail on large-scale instances

Heuristic algorithms

- Variable neighborhood search (VNS)
- Iterative local search (ILS)

  poor exploration ability, easily trapped into local optima

- Genetic algorithms (GA)
- Ant colony optimization(ACO) algorithms

  stronger exploration and global search ability

Solving CEVRPs

Exact algorithms

- mixed integer linear programming (MILP)

  may fail on large-scale instances

Heuristic algorithms

- Variable neighborhood search (VNS)
- Iterative local search (ILS)

  poor exploration ability, easily trapped into local optima

- Genetic algorithms (GA)
- **Ant colony optimization(ACO) algorithms**

  stronger exploration and global search ability

CEVRP

bi-level problem

Upper-level sub-problem

Constructing customer service sequence without recharging

ACO

Lower-level sub-problem

Inserting charging stations into the customer service sequence

Heuristic

CEVRP

bi-level problem

Upper-level sub-problem

Constructing customer service sequence without recharging

connection between battery capacity and cargo capacity is diluted

Lower-level sub-problem

Inserting charging stations into the customer service sequence

Can the ants construct

scheduling routing with charging stations and customers

at the same time?

Pheromone matrixes in existing ACO methods for CEVRP only record route information between customers.

We extend the pheromone matrix to record the historical information between customers and charging stations.

A new probability distribution is designed for the roulette wheel selection strategy, which considers the remaining battery level and is able to guide ants to select a suitable charging station at a proper time.

So, we call it

EACO

Traditional roulette wheel selection strategy

Move from customer to next customer

pheromone value of travelling from i to j

$$p_{ij} = \frac{\varphi_{ij}^{\alpha} / d_{ij}^{\beta}}{\sum_{l \in I_s} \varphi_{il}^{\alpha} / d_{il}^{\beta}}$$

the distance between i and j

New roulette wheel selection strategy

Move from customer/recharging station to next customer/recharging station

$$p_{i\to j} = \frac{\frac{\left[\varphi_{ij}\left(\left(\frac{kQ}{Rq_i}\right)^{\gamma}\tau_j+(1-\tau_j)\right)\right]^{\alpha}}{d_{ij}^{\beta}}}{\sum_{l\in I'}\left(\frac{\left[\varphi_{il}\left(\left(\frac{kQ}{Rq_i}\right)^{\gamma}\tau_l+(1-\tau_l)\right)\right]^{\alpha}}{d_{il}^{\beta}}\right)}$$

$$\tau_j \in \{0, 1\}$$

New roulette wheel selection strategy

Move from customer/recharging station to next customer/recharging station

the maximum battery capacity of each EV

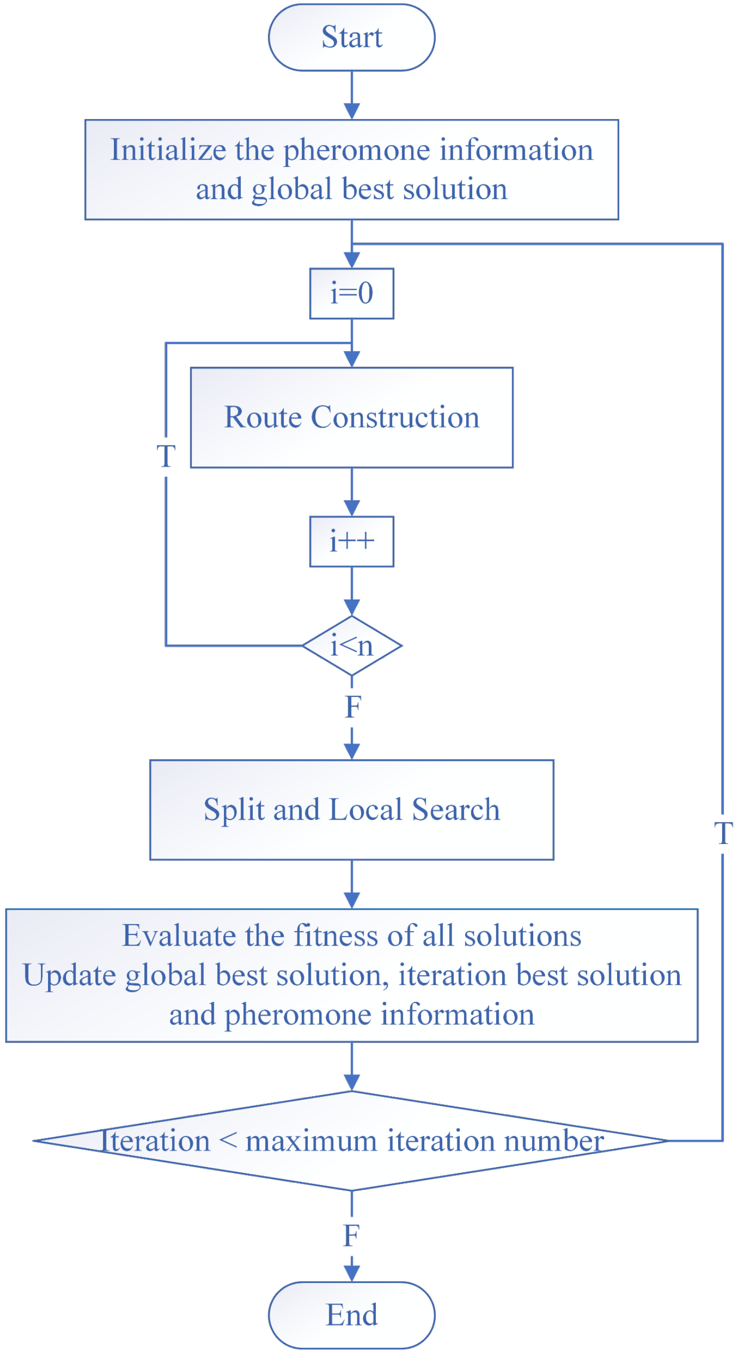
$$p_{i\to j} = \frac{\frac{\left[\varphi_{ij}\left(\left(\frac{kQ}{Rq_i}\right)^{\gamma}\tau_j+(1-\tau_j)\right)\right]^{\alpha}}{d_{ij}^{\beta}}}{\sum_{l\in I'}\left(\frac{\left[\varphi_{il}\left(\left(\frac{kQ}{Rq_i}\right)^{\gamma}\tau_l+(1-\tau_l)\right)\right]^{\alpha}}{d_{il}^{\beta}}\right)}$$

a fixed parameter ranging from 0 to 1

remaining battery level of an EV after it arrives at node i

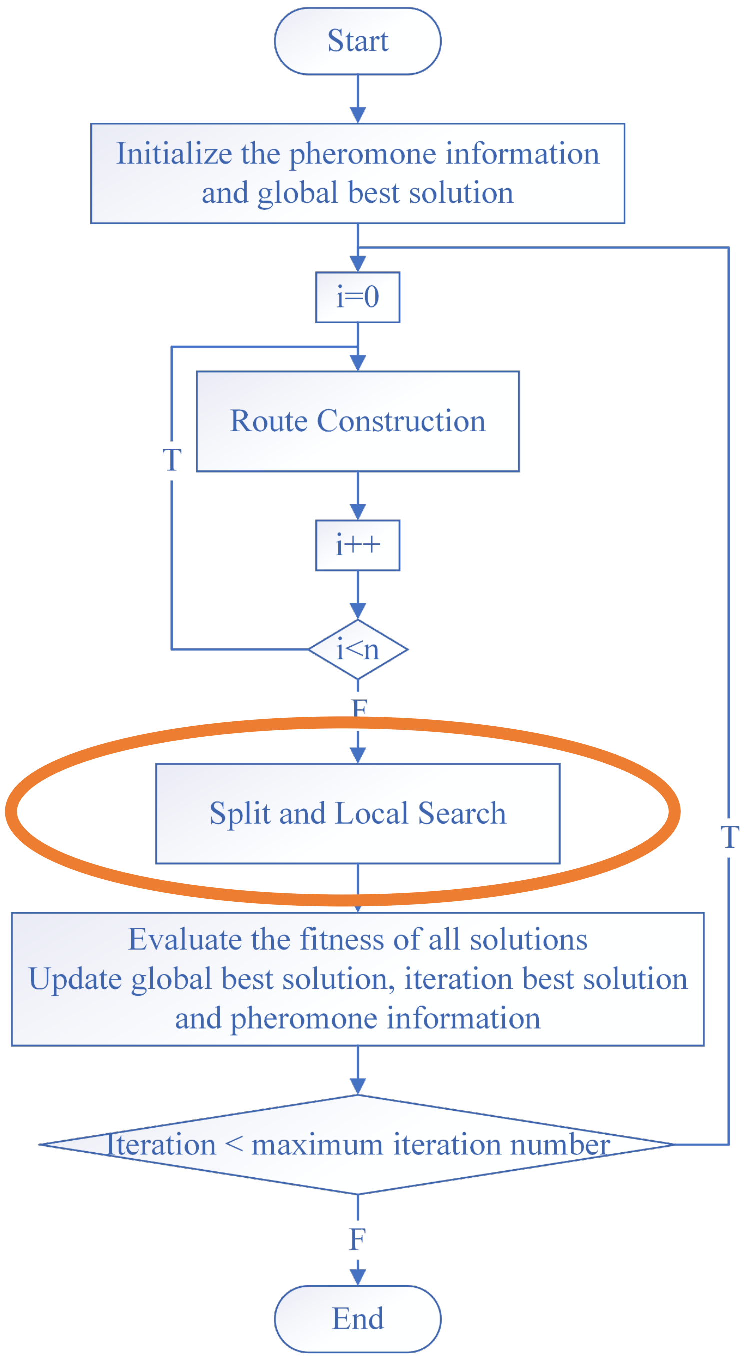
$$\tau_j \in \{0, 1\}$$

If j is the customer, τ_j equals 0, otherwise equals 1

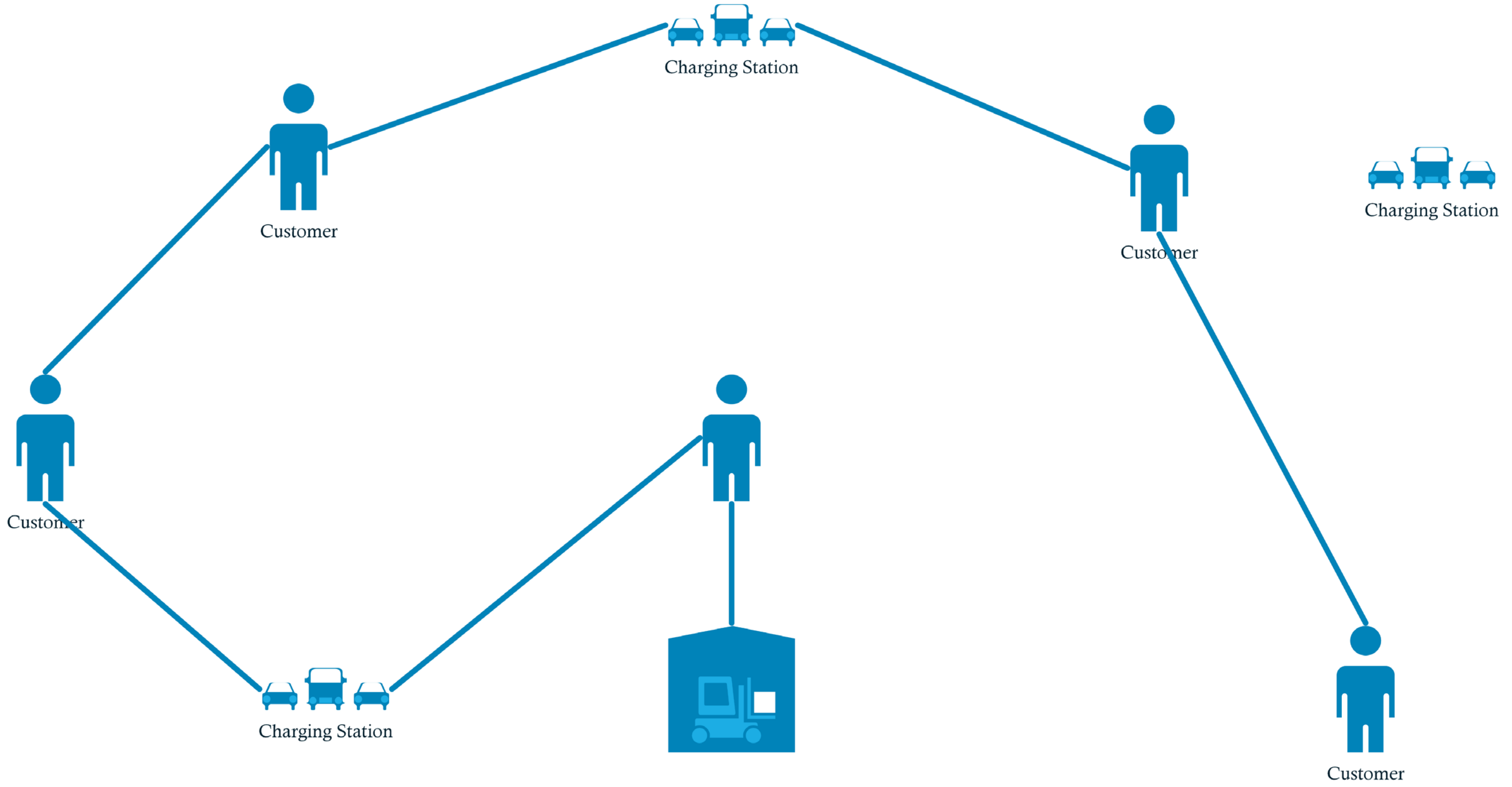


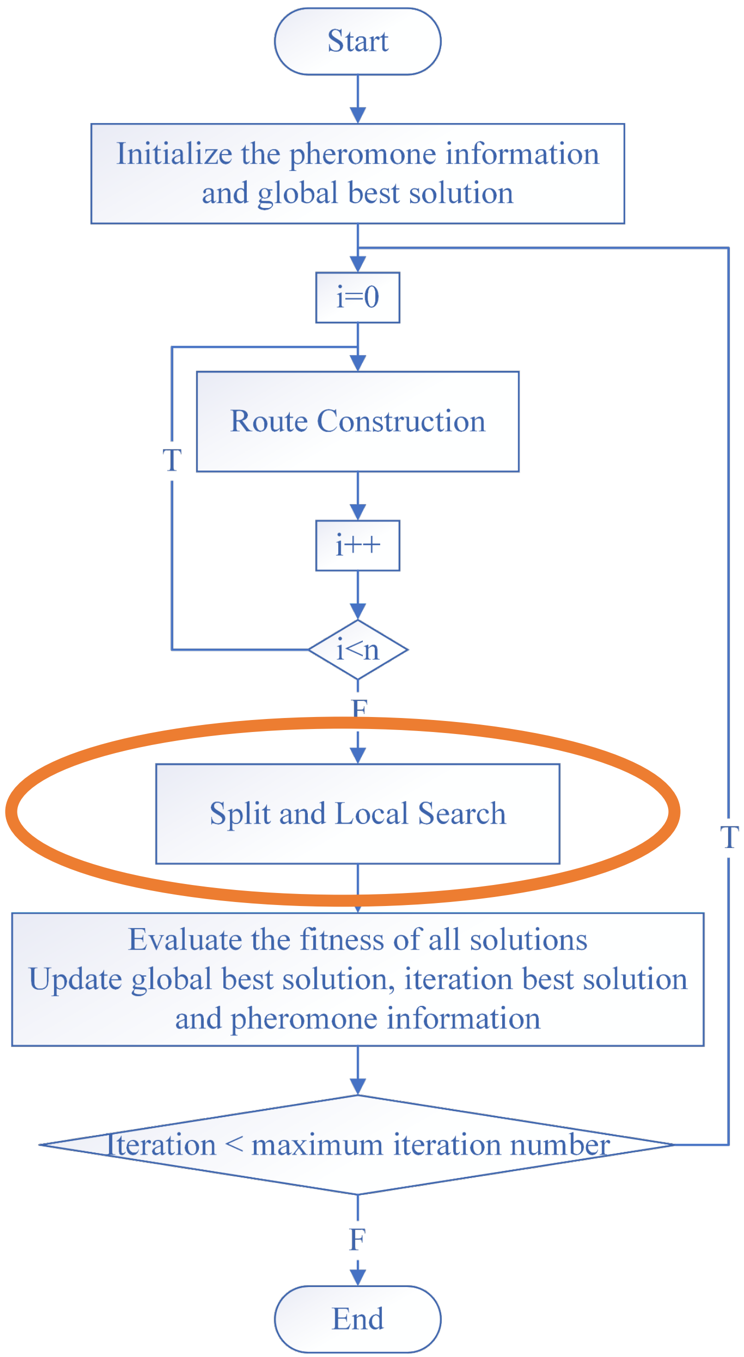
Overall Process

of EACO

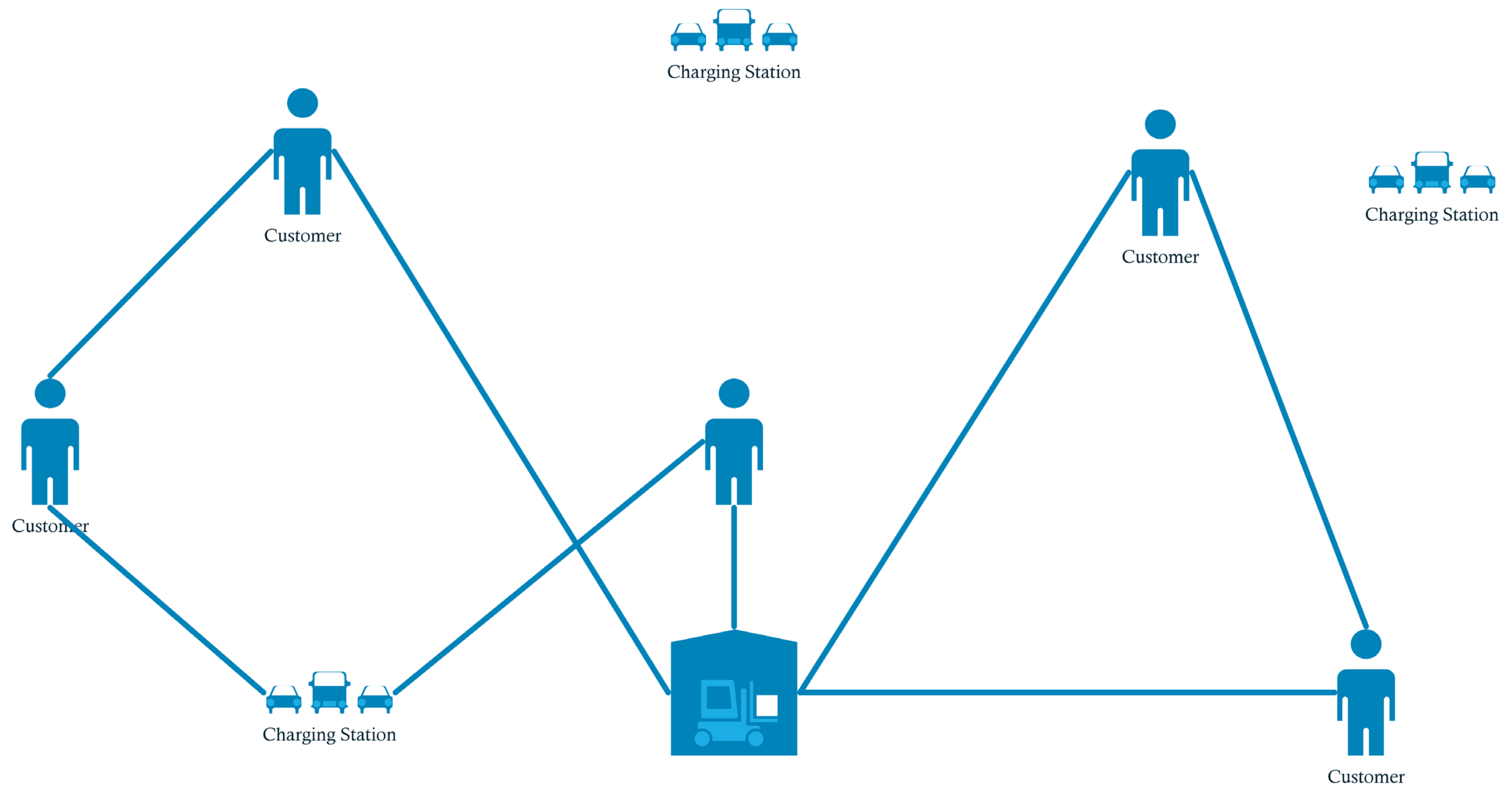

Start
Initialize the pheromone information and global best solution
i=0
Route Construction
T
i++
i<n
F
Split and Local Search
T
Evaluate the fitness of all solutions
Update global best solution, iteration best solution and pheromone information
Iteration < maximum iteration number
F
End

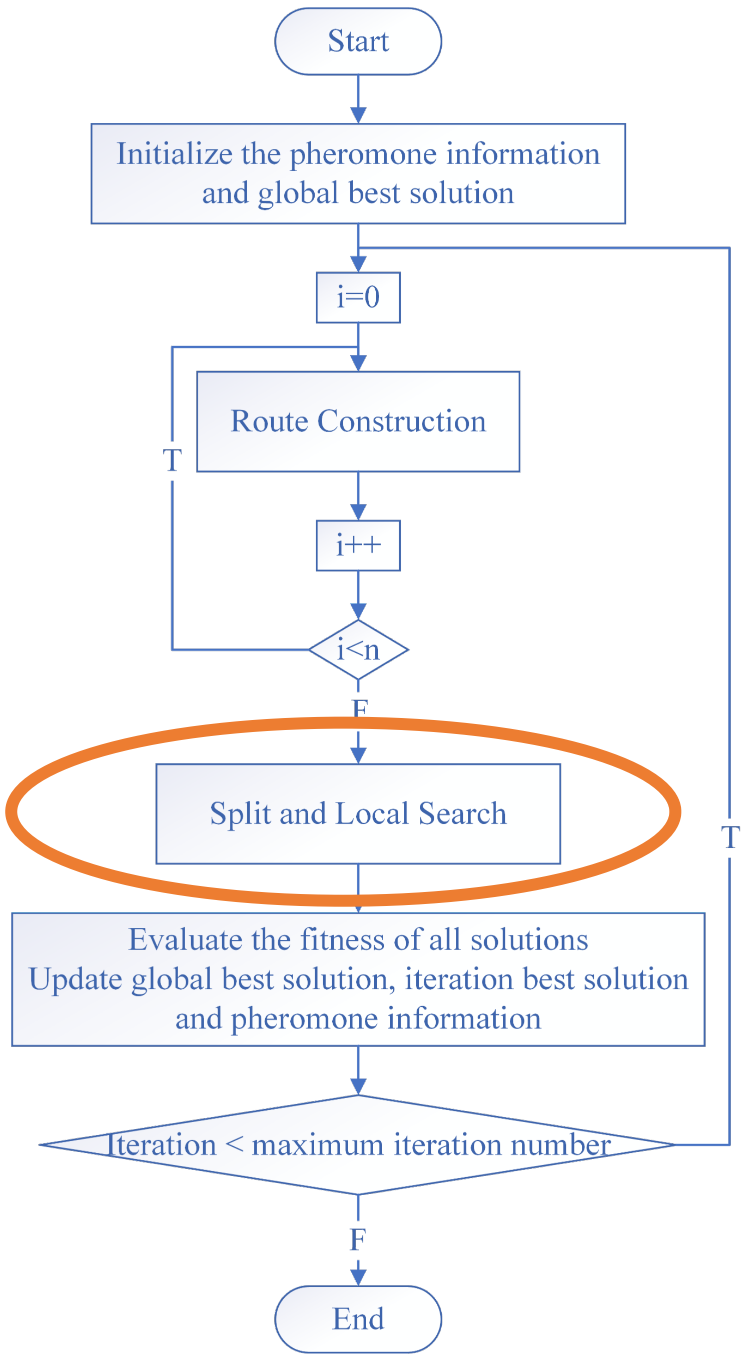
Giant Tour


Charging Station
Customer
Customer
Charging Station
Customer
Charging Station
Customer

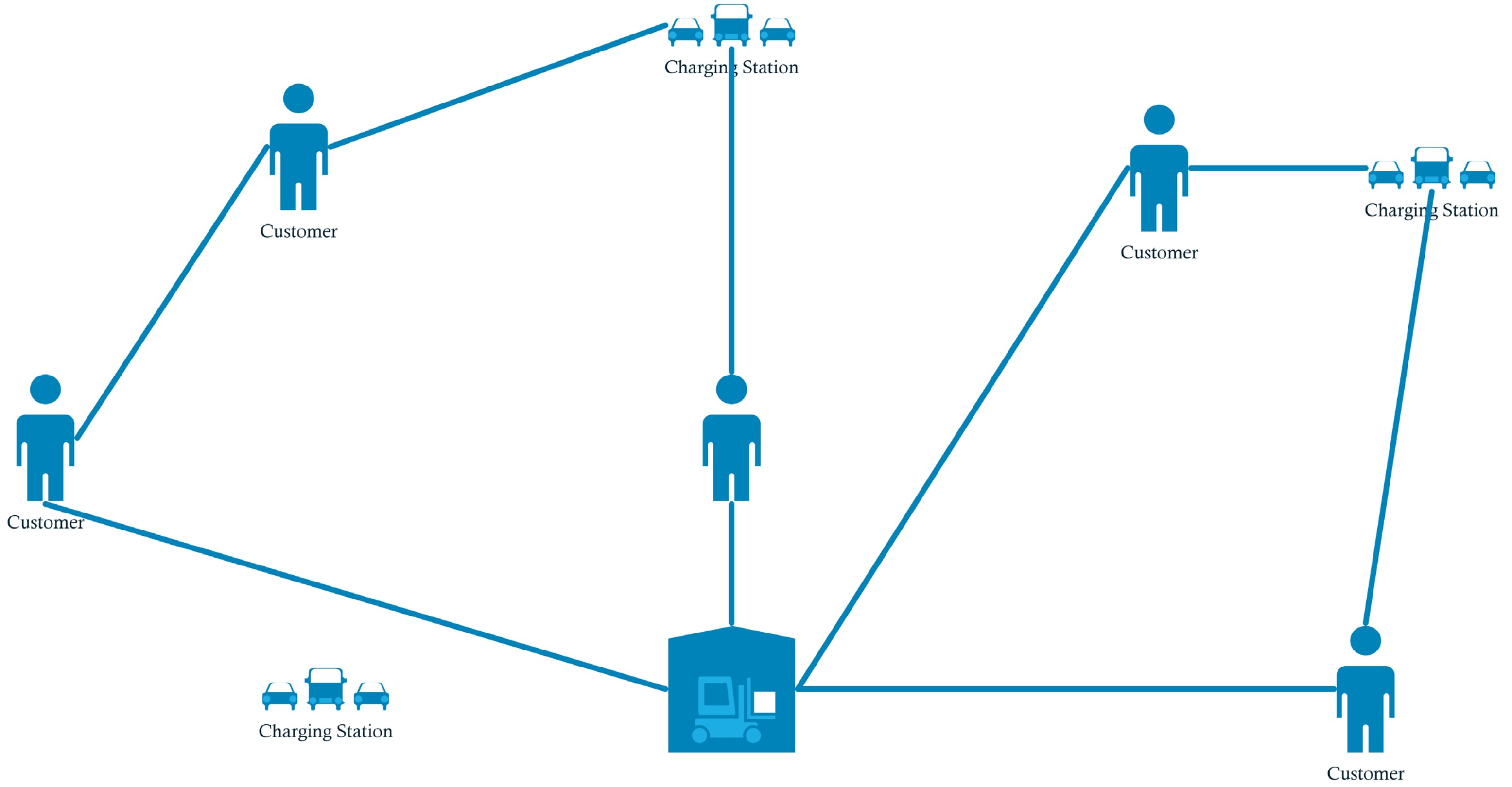

Start
Initialize the pheromone information and global best solution
i=0
Route Construction
T
i++
i<n
F
Split and Local Search
T
Evaluate the fitness of all solutions
Update global best solution, iteration best solution and pheromone information
Iteration < maximum iteration number
F
End

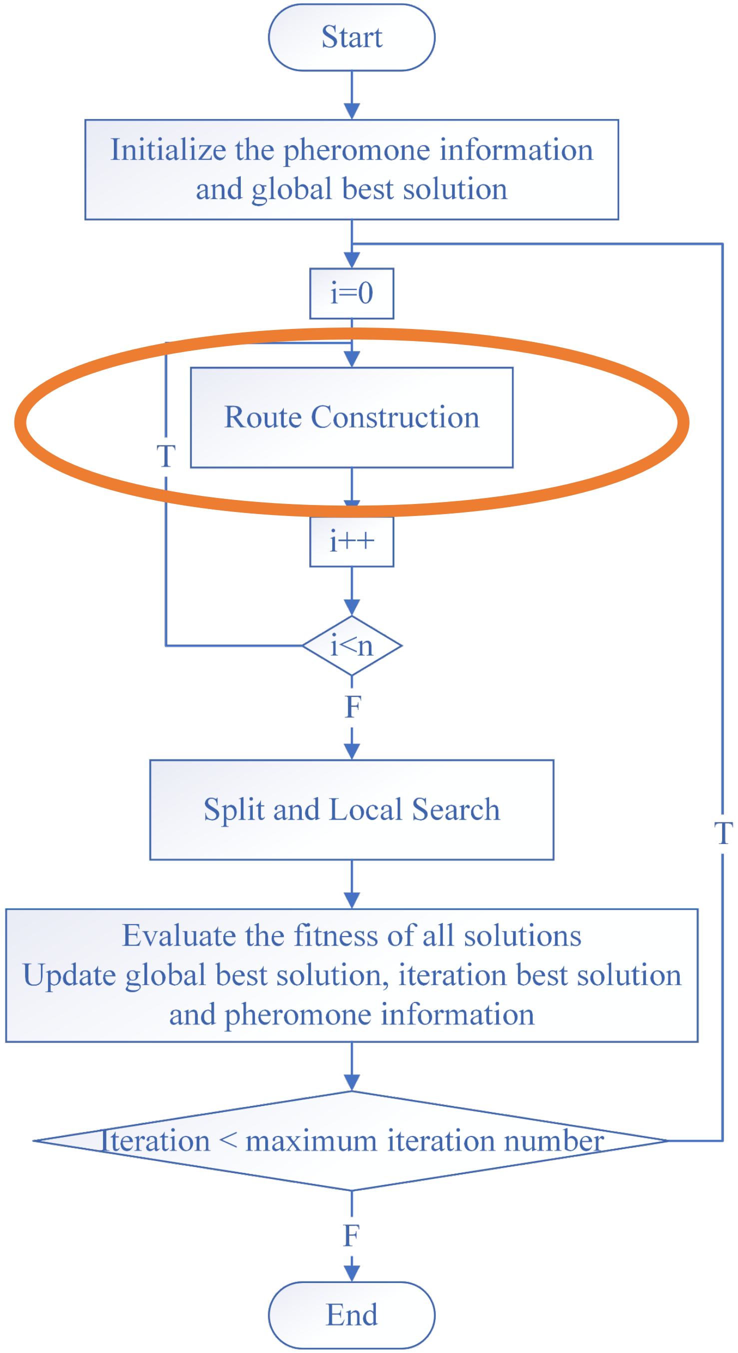
Split


Charging Station
Customer
Customer
Charging Station
Customer
Customer
Charging Station
Customer



Local Search



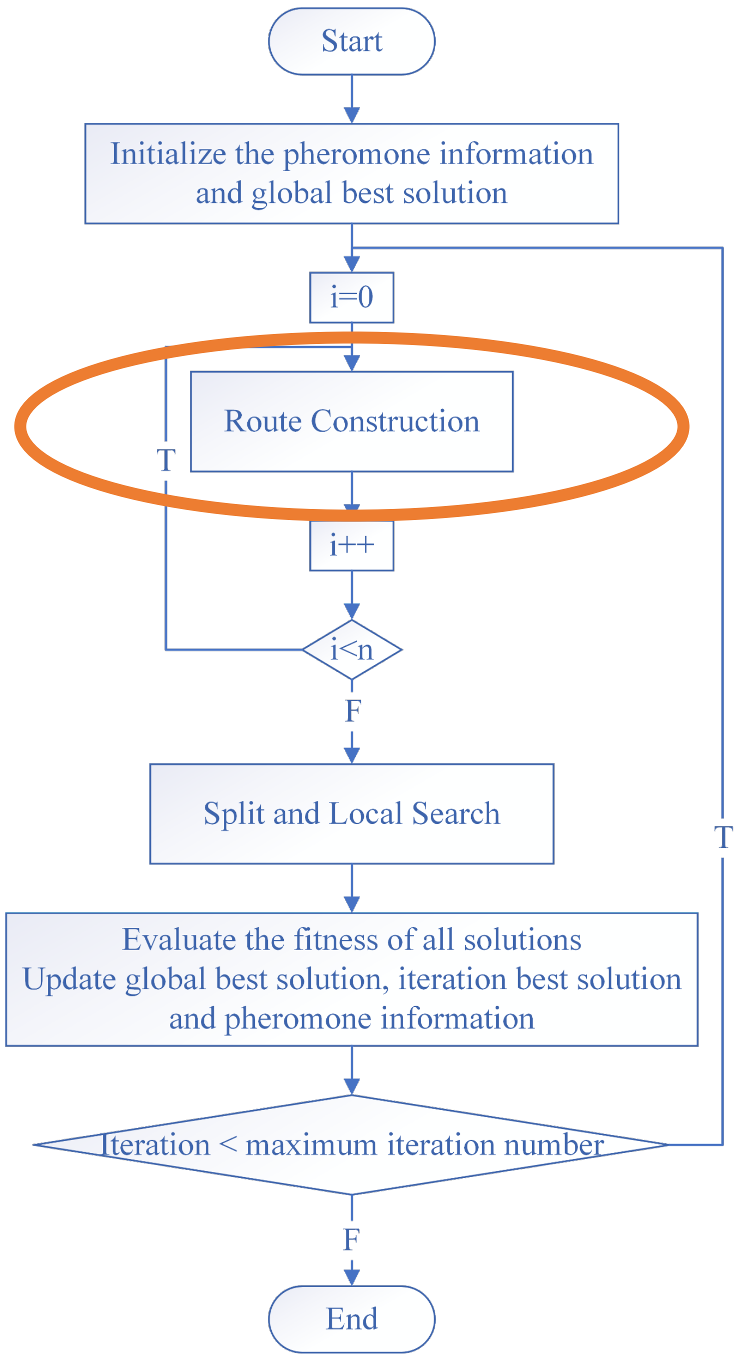


Algorithm 1: Coarse Route Construction

Input: Pheromone matrix Φ, arc set E, node set $\hat{I}$, maximum battery capacity Q

Output: a set of giant route T

initialize $T = \{0\}$, $I' = I - \{0\}$;
while I' *is not empty* **do**
    take the last node in T ;
    $j = roulette\_wheel\left(\hat{I}, i\right)$;
    append j to T ;
    remove j from I' ;
end



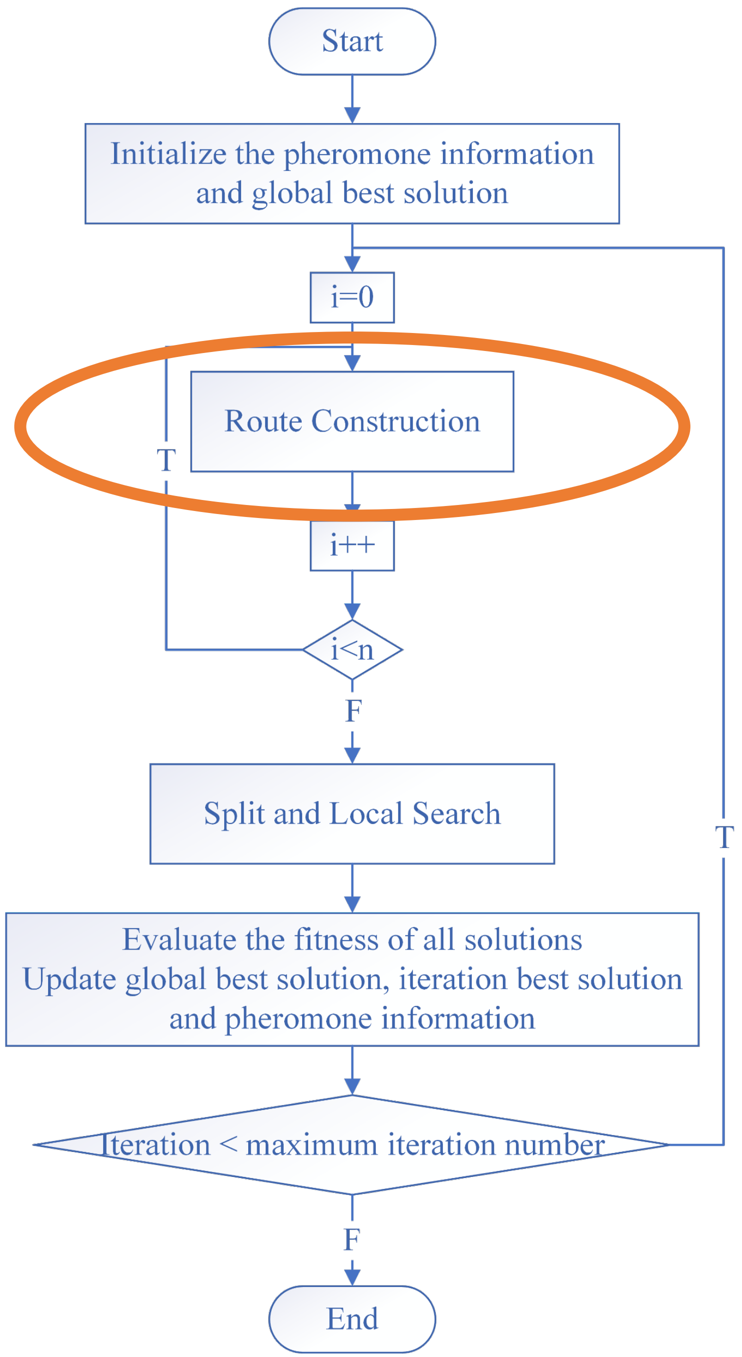
Algorithm 1: Coarse Route Construction

Input: Pheromone matrix Φ, arc set E, node set $\hat{I}$, maximum battery capacity Q

Output: a set of giant route T

```
initialize T = {0}, I' = I − {0} ;
while I' is not empty do
    take the last node in T ;
    j = roulette_wheel(Î, i) ;
    append j to T ;
    remove j from I' ;
end
```

However, the giant route will be split by capacity constraints, which leads to the fact that the battery required for returning to a depot should be considered.



Algorithm 2: Fine Route Construction

Input: Pheromone matrix Φ, arc set E, node set $\hat{I}$, maximum battery capacity Q, maximum capacity of cargo demand C

Output: a set of giant route T

initialize $T = \{0\}, I' = I - \{0\}$;
while I' *is not empty* **do**
  take the last node in T ;
  if $Rc_i < average_{demand}$ **then**
    $Rq_i = Q$;
    $Rc_i = C$;
  end
  $j = roulette\_wheel\left(\hat{I}, i\right)$;
  append j to T ;
  remove j from I' ;
end

we re-evaluate the remaining battery level in the fine route construction method to include the battery consumption for returning to depots.

OK, let's compare EACO with SLMMAS.

SLMMAS inserts charging stations during the construction of a route. Whenever the next selected customer cannot be reached due to the electricity constraint, one charging station that brings minimum extra distance is visited before serving the customer.

TABLE I
SEVENTEEN INSTANCES OF CEVRP

Case	Customer	Station	Route
E22	21	8	4
E23	22	9	3
E30	29	6	4
E33	32	6	4
E51	50	5	5
E76	75	7	7
E101	100	9	8
X143	142	4	7
X214	213	9	11
X351	350	35	40
X459	458	20	26
X573	572	6	30
X685	684	25	75
X749	748	30	98
X819	818	25	171
X916	915	9	207
X1001	1000	9	43

Benchmark

IEEE WCCI2020 competition

Small-scale

Medium-scale

Large-scale

Instance		EACO				SLMMAS	
		EACO-C		EACO-F		AS-S	AS-O
E22	best	384.68		384.68		**386.15**	385.44
	mean	385.62	=	**384.68**	+	**386.15**	385.44
	std.	2.10		0.00		0.00	0.00
E23	best	585.99		579.07		**582.61**	**582.61**
	mean	587.16	-	**584.03**	=	**582.88**	582.94
	std.	1.62		3.40		0.60	0.60
E30	best	513.63		**513.63**		512.72	514.42
	mean	514.27	-	**513.63**	-	**512.72**	516.11
	std.	1.41		0.00		0.00	0.61
E33	best	846.79		847.07		860.62	**859.25**
	mean	**850.44**	+	850.98	+	863.44	**860.06**
	std.	2.55		2.22		2.18	0.74
E51	best	**572.36**		573.26		551.19	549.81
	mean	578.27	-	**575.14**	-	**559.34**	564.93
	std.	5.73		1.82		5.60	8.70
E76	best	748.04		**746.79**		744.07	724.24
	mean	756.67	=	**752.54**	=	**754.03**	762.09
	std.	13.23		8.62		8.43	13.23
E101	best	915.00		**904.38**		872.98	891.77
	mean	926.67	-	**908.39**	-	**885.28**	904.62
	std.	11.74		4.67		10.09	4.64
X143	best	16434.32		16465.70		**16885.63**	17238.76
	mean	**16619.80**	+	16627.03	+	**17099.20**	17985.91
	std.	217.13		108.45		271.24	415.20
X214	best	12160.14		**12088.35**		12197.75	11421.43
	mean	12369.90	=	**12197.89**	=	12256.69	**12023.72**
	std.	187.91		74.21		41.90	129.22
X351	best	29116.15		**28750.95**		27749.48	31927.71
	mean	29406.93	-	**29079.51**	-	**27820.63**	31927.71
	std.	294.22		187.41		65.59	0.00
X459	best	27986.71		27468.29		**28251.70**	28448.27
	mean	28081.82	+	**27986.53**	+	**28469.61**	29434.48
	std.	85.34		299.65		153.36	540.16
X573	best	56738.10		55986.28		57670.41	**56609.94**
	mean	57068.45	+	**56752.22**	+	58167.74	**57170.35**
	std.	306.68		433.90		289.61	199.81
X685	best	**79673.84**		79822.20		78862.87	84435.72
	mean	**80076.58**	-	80168.38	-	**79012.35**	84435.72
	std.	264.30		266.54		104.89	0.00
X749	best	87040.68		87731.50		**88070.85**	90441.60
	mean	**87450.01**	+	87807.86	+	**88381.83**	90441.36
	std.	265.64		63.69		214.63	0.00
X819	best	176356.85		**176184.24**		175315.97	171553.21
	mean	**176535.74**	-	176726.23	-	175540.33	**172355.34**
	std.	170.28		315.63		184.18	311.09
X916	best	**363784.87**		364092.53		363537.41	353046.07
	mean	**364557.39**	=	364755.93	=	364091.13	**354262.15**
	std.	613.26		623.46		321.96	464.68
X1001	best	83425.31		83103.57		**86064.64**	89611.93
	mean	83569.87	+	**83451.84**	+	**86064.64**	89611.93
	std.	148.98		278.26		0.00	0.00

Significant Better 6&7

Equal 4

Significant Worse 7&6

Better* 11

All algorithms use the same population size and stopping criterion for fair comparisons and SLMMAS is denoted as AS-S. In addition, the results of SLMMAS are also presented with the same parameter setting as that in the original study for analysis, which is denoted as AS-O.

*We cannot apply the Wilcoxon rank-sum test between EACO and AS-O, for the detailed result of AS-O is not available.

Instance		EACO				SLMMAS	
		EACO-C		EACO-F		AS-S	AS-O
E22	best	384.68		384.68		**386.15**	385.44
	mean	385.62	=	**384.68**	+	**386.15**	385.44
	std.	2.10		0.00		0.00	0.00
E23	best	585.99		579.07		**582.61**	**582.61**
	mean	587.16	-	**584.03**	=	**582.88**	582.94
	std.	1.62		3.40		0.60	0.60
E30	best	513.63		**513.63**		512.72	514.42
	mean	514.27	-	**513.63**	-	**512.72**	516.11
	std.	1.41		0.00		0.00	0.61
E33	best	846.79		847.07		860.62	**859.25**
	mean	**850.44**	+	850.98	+	863.44	**860.06**
	std.	2.55		2.22		2.18	0.74
E51	best	**572.36**		573.26		551.19	549.81
	mean	578.27	-	**575.14**	-	**559.34**	564.93
	std.	5.73		1.82		5.60	8.70
E76	best	748.04		**746.79**		744.07	724.24
	mean	756.67	=	**752.54**	=	**754.03**	762.09
	std.	13.23		8.62		8.43	13.23
E101	best	915.00		**904.38**		872.98	891.77
	mean	926.67	-	**908.39**	-	**885.28**	904.62
	std.	11.74		4.67		10.09	4.64
X143	best	16434.32		16465.70		**16885.63**	17238.76
	mean	**16619.80**	+	16627.03	+	**17099.20**	17985.91
	std.	217.13		108.45		271.24	415.20
X214	best	12160.14		**12088.35**		12197.75	11421.43
	mean	12369.90	=	**12197.89**	=	12256.69	**12023.72**
	std.	187.91		74.21		41.90	129.22
X351	best	29116.15		**28750.95**		27749.48	31927.71
	mean	29406.93	-	**29079.51**	-	**27820.63**	31927.71
	std.	294.22		187.41		65.59	0.00
X459	best	27986.71		27468.29		**28251.70**	28448.27
	mean	28081.82	+	**27986.53**	+	**28469.61**	29434.48
	std.	85.34		299.65		153.36	540.16
X573	best	56738.10		55986.28		57670.41	**56609.94**
	mean	57068.45	+	**56752.22**	+	58167.74	**57170.35**
	std.	306.68		433.90		289.61	199.81
X685	best	**79673.84**		79822.20		78862.87	84435.72
	mean	**80076.58**	-	80168.38	-	**79012.35**	84435.72
	std.	264.30		266.54		104.89	0.00
X749	best	87040.68		87731.50		**88070.85**	90441.60
	mean	**87450.01**	+	87807.86	+	**88381.83**	90441.36
	std.	265.64		63.69		214.63	0.00
X819	best	176356.85		**176184.24**		175315.97	171553.21
	mean	**176535.74**	-	176726.23	-	175540.33	**172355.34**
	std.	170.28		315.63		184.18	311.09
X916	best	**363784.87**		364092.53		363537.41	353046.07
	mean	**364557.39**	=	364755.93	=	364091.13	**354262.15**
	std.	613.26		623.46		321.96	464.68
X1001	best	83425.31		83103.57		**86064.64**	89611.93
	mean	83569.87	+	**83451.84**	+	**86064.64**	89611.93
	std.	148.98		278.26		0.00	0.00

The proposed EACO outperforms SLMMAS on medium-scale instances and the largest instance.

All algorithms use the same population size and stopping criterion for fair comparisons and SLMMAS is denoted as AS-S. In addition, the results of SLMMAS are also presented with the same parameter setting as that in the original study for analysis, which is denoted as AS-O.

Instance		EACO				SLMMAS	
		EACO-C		EACO-F		AS-S	AS-O
E22	best	384.68		384.68		386.15	385.44
	mean	385.62	=	384.68	+	386.15	385.44
	std.	2.10		0.00		0.00	0.00
E23	best	585.99		579.07		582.61	582.61
	mean	587.16	-	584.03	=	582.88	582.94
	std.	1.62		3.40		0.60	0.60
E30	best	513.63		513.63		512.72	514.42
	mean	514.27	-	513.63	-	512.72	516.11
	std.	1.41		0.00		0.00	0.61
E33	best	846.79		847.07		860.62	859.25
	mean	850.44	+	850.98	+	863.44	860.06
	std.	2.55		2.22		2.18	0.74
E51	best	572.36		573.26		551.19	549.81
	mean	578.27	-	575.14	-	559.34	564.93
	std.	5.73		1.82		5.60	8.70
E76	best	748.04		746.79		744.07	724.24
	mean	756.67	=	752.54	=	754.03	762.09
	std.	13.23		8.62		8.43	13.23
E101	best	915.00		904.38		872.98	891.77
	mean	926.67	-	908.39	-	885.28	904.62
	std.	11.74		4.67		10.09	4.64
X143	best	16434.32		16465.70		16885.63	17238.76
	mean	16619.80	+	16627.03	+	17099.20	17985.91
	std.	217.13		108.45		271.24	415.20
X214	best	12160.14		12088.35		12197.75	11421.43
	mean	12369.90	=	12197.89	=	12256.69	12023.72
	std.	187.91		74.21		41.90	129.22
X351	best	29116.15		28750.95		27749.48	31927.71
	mean	29406.93	-	29079.51	-	27820.63	31927.71
	std.	294.22		187.41		65.59	0.00
X459	best	27986.71		27468.29		28251.70	28448.27
	mean	28081.82	+	27986.53	+	28469.61	29434.48
	std.	85.34		299.65		153.36	540.16
X573	best	56738.10		55986.28		57670.41	56609.94
	mean	57068.45	+	56752.22	+	58167.74	57170.35
	std.	306.68		433.90		289.61	199.81
X685	best	79673.84		79822.20		78862.87	84435.72
	mean	80076.58	-	80168.38	-	79012.35	84435.72
	std.	264.30		266.54		104.89	0.00
X749	best	87040.68		87731.50		88070.85	90441.60
	mean	87450.01	+	87807.86	+	88381.83	90441.36
	std.	265.64		63.69		214.63	0.00
X819	best	176356.85		176184.24		175315.97	171553.21
	mean	176535.74	-	176726.23	-	175540.33	172355.34
	std.	170.28		315.63		184.18	311.09
X916	best	363784.87		364092.53		363537.41	353046.07
	mean	364557.39	=	364755.93	=	364091.13	354262.15
	std.	613.26		623.46		321.96	464.68
X1001	best	83425.31		83103.57		86064.64	89611.93
	mean	83569.87	+	83451.84	+	86064.64	89611.93
	std.	148.98		278.26		0.00	0.00

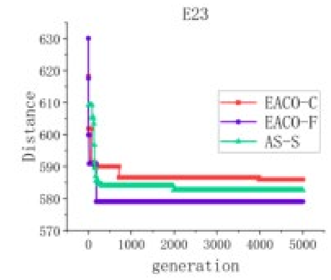
AS-S has overtaken AS-O* on 12 instances, by comparing the mean values**, especially on the medium-scale and large-scale instances.

It indicates that the SLMMAS can find better solutions using more iterations rather than larger population sizes.

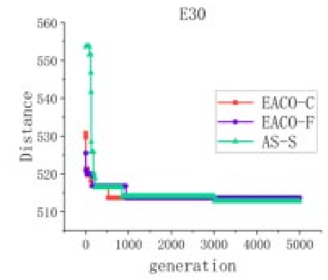
All algorithms use the same population size and stopping criterion for fair comparisons and SLMMAS is denoted as AS-S. In addition, the results of SLMMAS are also presented with the same parameter setting as that in the original study for analysis, which is denoted as AS-O.

*The algorithm setting of AS-O is that the stopping criterion is defined as fixed execution time and population size is set to n = |I| + 1 for each instance.

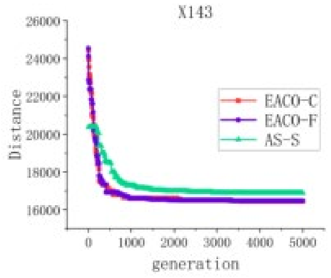
**We cannot apply the Wilcoxon rank-sum test between EACO and AS-O, for the detailed result of AS-O is not available.



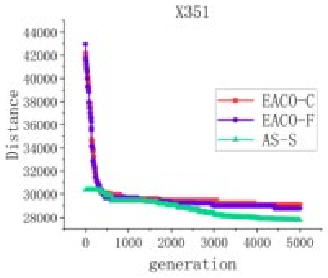
(a) E23



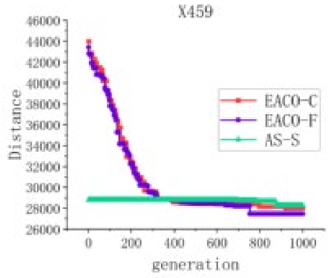
(b) E30



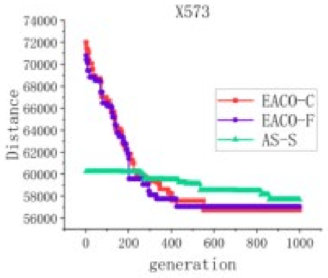
(c) X143



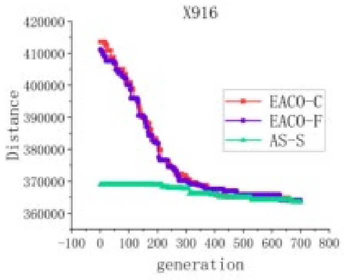
(d) X351



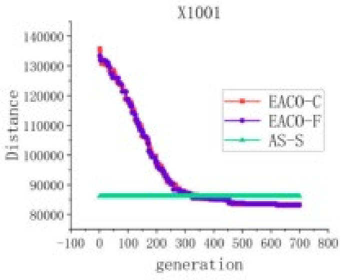
(e) X459



(f) X573



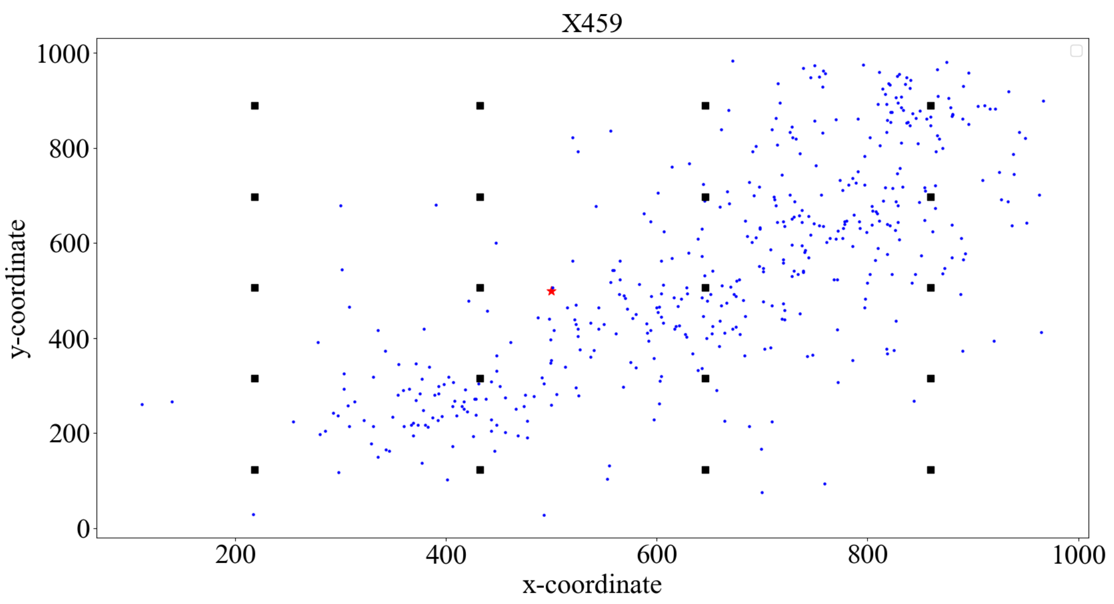
(g) X916

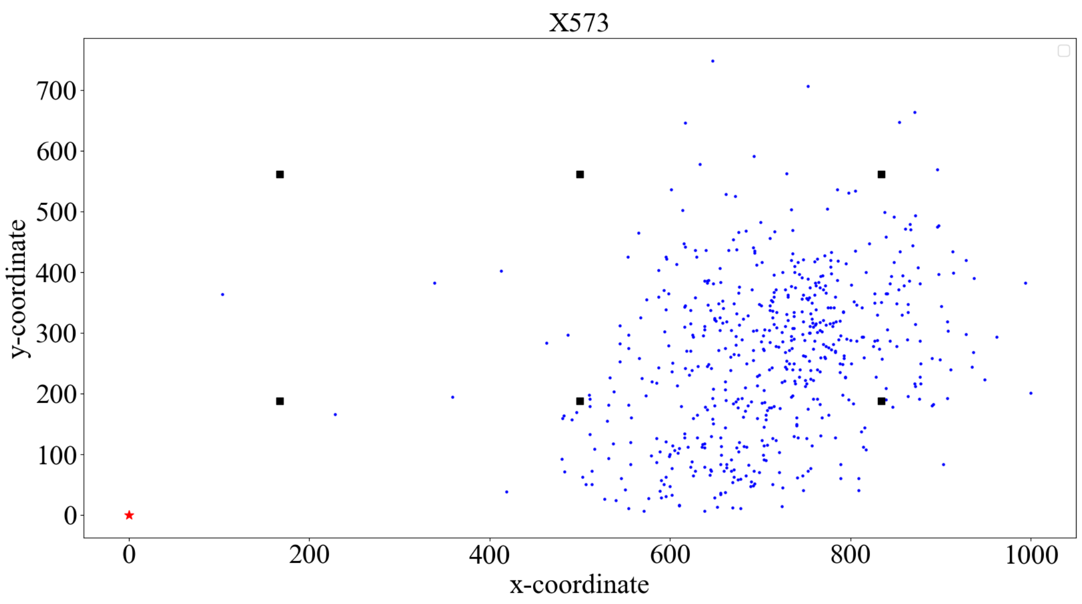


(h) X1001

Generally, SLMMAS can construct good solutions in early stage and may stagnate quickly. It can deal with small-scale problems, but failed on large-scale instances. By contrast, though EACO finds worse solutions in the early stage but converges quickly and surpasses SLMMAS on most instances. In addition, EACO presents stronger exploration ability, especially on medium-scale and large-scale instances.

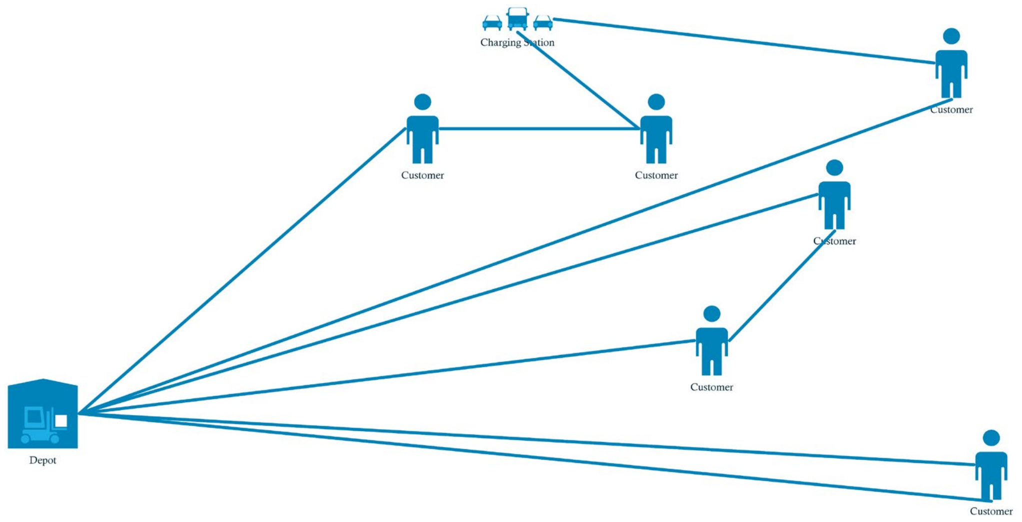
All algorithms use the same population size and stopping criterion for fair comparisons and SLMMAS is denoted as AS-S.





EACO-F is better

EACO-C is better



The EVs run frequently between depot and customers with a long distance, which makes the search space of EACO-F extremely huge and hard to find better solutions.

We generally recommend

EACO

with fine route construction method

THANKS

2022.12

Efficient Extended Ant Colony Optimization for Capacitated Electric Vehicle Routing

Bo-Cheng Lin, Xiao-Fang Liu, Yi Mei