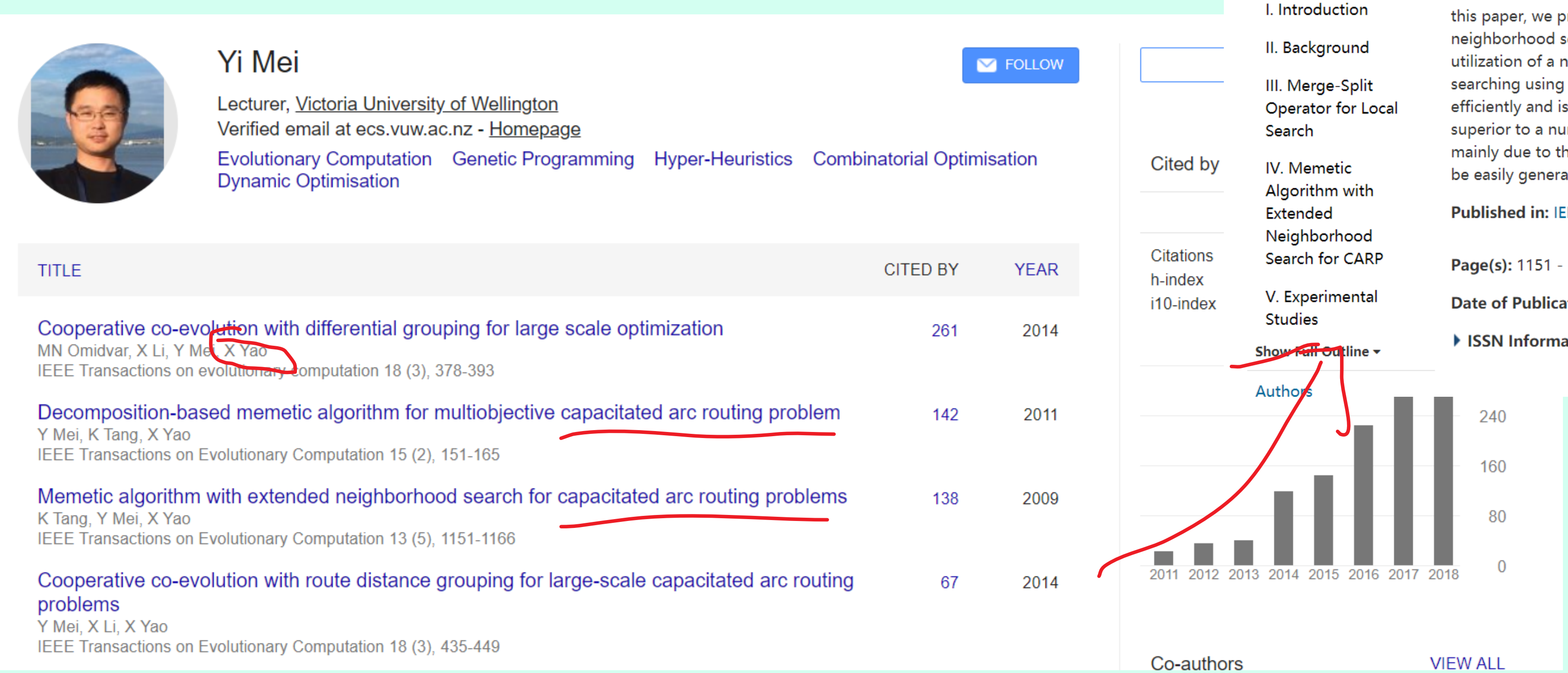
The (Undirected) Capacitated Arc Routing Problem (CARP) Introduction and Methods

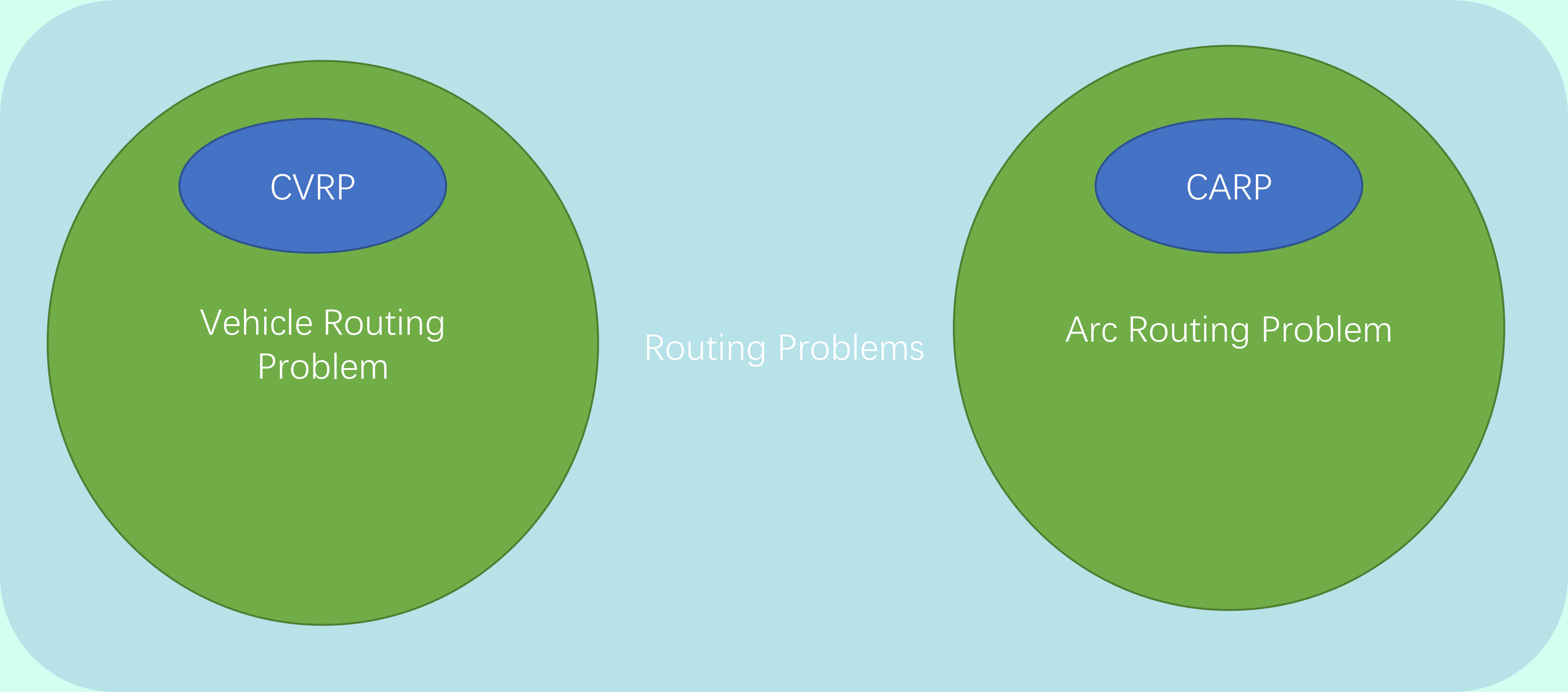
Yidong FANG（方一栋）
fangyd1997@outlook.com

Some Fun


IEEE.org | IEEE Xplore Digital Library | IEEE-SA | IEEE Spectrum | More Sites
IEEE Xplore® Digital Library
Institutional Sign In
Browse
My Settings
Get Help
Subscribe
All
Enter keywords or phrases (Note: Searches metadata only by default. A search for 'smart grid' = 'smart AND grid')
Browse Journals & Magazines > IEEE Transactions on Evolutio... > Volume: 13 Issue: 5
Memetic Algorithm With Extended Neighborhood Search for Capacitated Arc Routing Problems
3 Author(s)
Ke Tang ; Yi Mei ; Xin Yao
View All Authors
92 Paper Citations
751 Full Text Views
Abstract
Document Sections
I. Introduction
II. Background
III. Merge-Split Operator for Local Search
IV. Memetic Algorithm with Extended Neighborhood Search for CARP
V. Experimental Studies
Show Full Outline
Authors
Abstract:
The capacitated arc routing problem (CARP) has attracted much attention during the last few years due to its wide applications in real life. Since CARP is NP-hard and exact methods are only applicable to small instances, heuristic and metaheuristic methods are widely adopted when solving CARP. In this paper, we propose a memetic algorithm, namely memetic algorithm with extended neighborhood search (MAENS), for CARP. MAENS is distinct from existing approaches in the utilization of a novel local search operator, namely Merge-Split (MS). The MS operator is capable of searching using large step sizes, and thus has the potential to search the solution space more efficiently and is less likely to be trapped in local optima. Experimental results show that MAENS is superior to a number of state-of-the-art algorithms, and the advanced performance of MAENS is mainly due to the MS operator. The application of the MS operator is not limited to MAENS. It can be easily generalized to other approaches.
Published in: IEEE Transactions on Evolutionary Computation (Volume: 13 , Issue: 5 , Oct. 2009)
Page(s): 1151 - 1166
Date of Publication: 11 August 2009
ISSN Information:
INSPEC Accession Number: 10879801
DOI: 10.1109/TEVC.2009.2023449
Publisher: IEEE
Sponsored by: IEEE Computational Intelligence

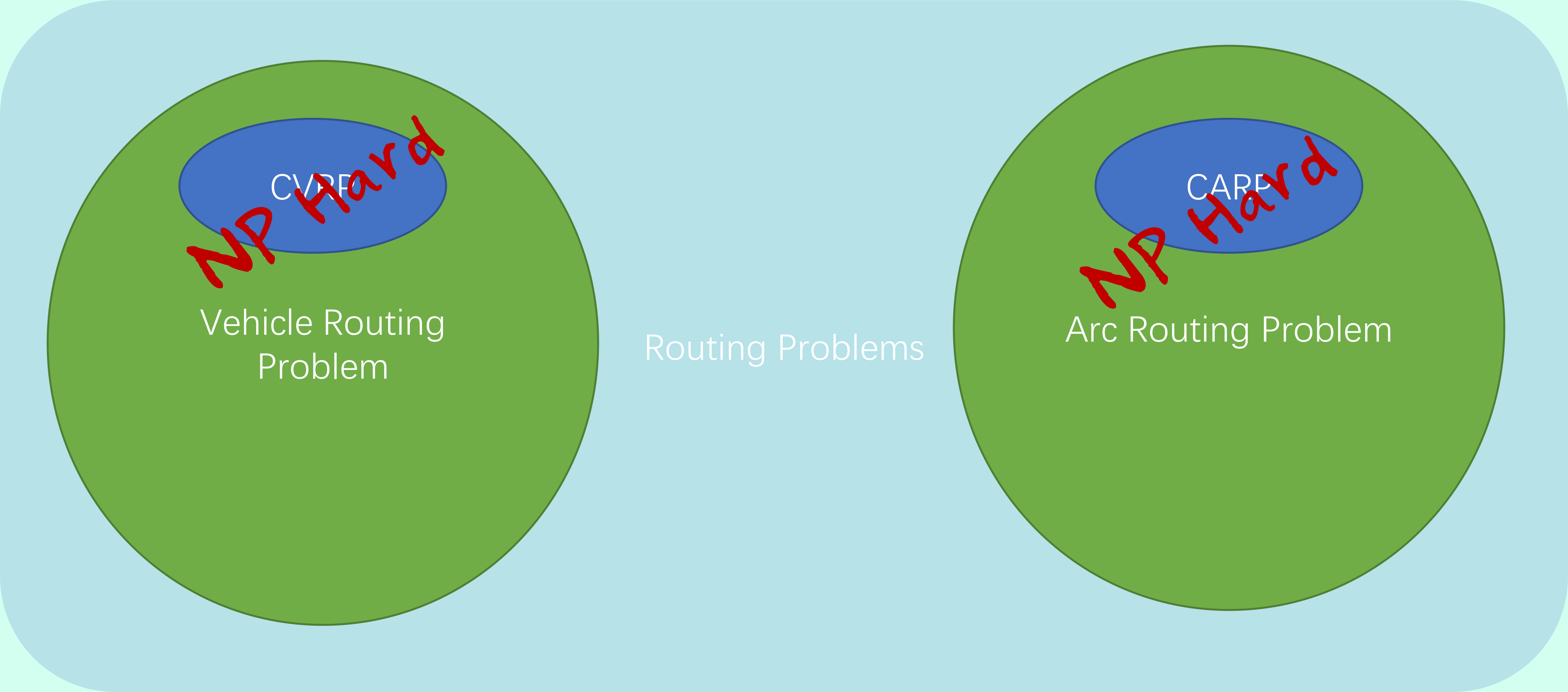

Yi Mei
FOLLOW
Lecturer, Victoria University of Wellington
Verified email at ecs.vuw.ac.nz - Homepage
Evolutionary Computation
Genetic Programming
Hyper-Heuristics
Combinatorial Optimisation
Dynamic Optimisation
TITLE
CITED BY
YEAR
Cooperative co-evolution with differential grouping for large scale optimization
MN Omidvar, X Li, Y Mei, X Yao
IEEE Transactions on evolutionary computation 18 (3), 378-393
261
2014
Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem
Y Mei, K Tang, X Yao
IEEE Transactions on Evolutionary Computation 15 (2), 151-165
142
2011
Memetic algorithm with extended neighborhood search for capacitated arc routing problems
K Tang, Y Mei, X Yao
IEEE Transactions on Evolutionary Computation 13 (5), 1151-1166
138
2009
Cooperative co-evolution with route distance grouping for large-scale capacitated arc routing problems
Y Mei, X Li, X Yao
IEEE Transactions on Evolutionary Computation 18 (3), 435-449
67
2014
Cited by
Citations
h-index
i10-index
2011
2012
2013
2014
2015
2016
2017
2018
240
160
80
0
Co-authors
VIEW ALL

Problem Bound



A set of **combinational optimization problems** that many research communities are researching on. Two main branch: Operational Research & Computer Science

Problem Bound



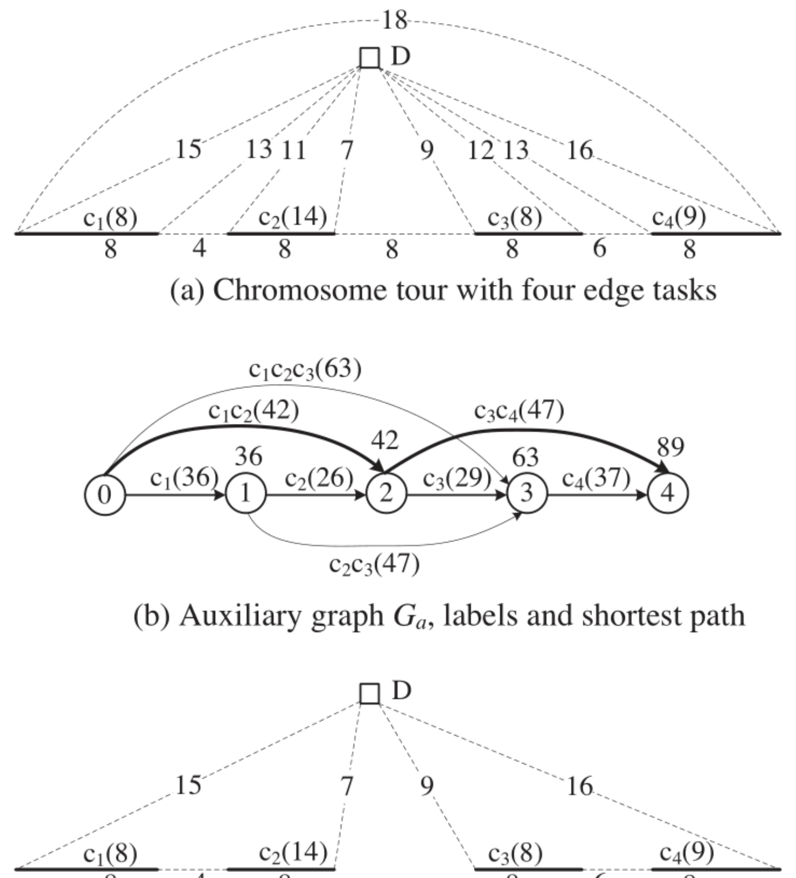
Heuristics Methods Catogries

- Heuristics
  - problem-dependent techniques
- Meta-heuristics
  - problem-independent techniques
- Hyper-heuristics
  - "heuristics to search for heuristics"

What I Used Last Year: A brief Intro

- Metaheuristics: Memetic algorithms
  - Encoding
  - Crossover
  - Mutation
  - **Iterative local search**
  - Other techniques: multiprocessing, restarting, etc.

Given a chromosome sequence, the optimal solution is set, which means the **chromosomes and the solutions are one-to-one mapping**.



(a) Chromosome tour with four edge tasks

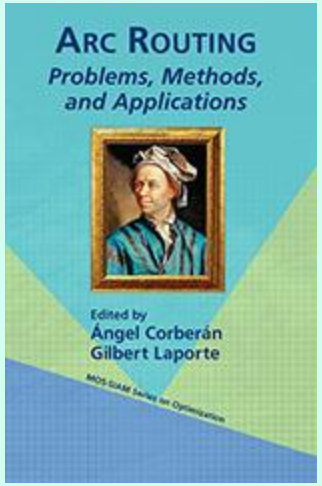
(b) Auxiliary graph G_a, labels and shortest path

(c) Resulting trips

Tiantang Liu, Zhibin Jiang & Na Geng (2013) A memetic algorithm with iterated local search for the capacitated arc routing problem, International Journal of Production Research, 51:10, 3075-3084, DOI: 10.1080/00207543.2012.753165

Initial Solution - Heuristics

- Path-scanning (Golden, DeArmon, and Baker)
- Augment-merge (Tried, but failed)
- Ulusoy’s route-first cluster-second method
- Others



A book strongly recommended:

ARC Routing: Problems, Methods, and Applications. (2015). Erscheinungsort nicht ermittelbar: Society for Industrial & Applied Mathematics, U.S.

Chapter 7 The Capacitated Arc Routing Problem Heuristics

Iterative Local Search

- Flip one task a, i.e., replace a by inv(a) in its trip,
- Move one task a after another task or after the depot,
- Move two consecutive tasks a and b after another task or after the depot
- Swap two tasks a and b
- 2-opt moves

Remember to the advantages of local search is that it helps you to reduce the computation by only calculate the different between original solution and the new one!

Iterative: You can set a score function to adaptively change the local search chance for better saving the time. For example you may add more possibility at the starting stage of local searching even some of the solutions violate the capacity. However, after iterative local search, you may also get the valid solution jump out the local minimal.

Other things You Need to Note

- Make efficient use of Python Data structure (dict, list, etc.)
- Those researcher often use C++ to write their program, so what you do here is just a proof of thought.
- Use Multi-processing instead of multi-threading (what you learn by yourself will help you in the future Operating System course)
- Cross-process communication/Synchronization?
- Take your reference seriously and give credit to the authors of paper/book/website in your report even no one will read your report in the future (who knows).

State of Art Result – Not Mine

Table 7.3. *Performance on the 34 val instances* ($n = 24–50$, $t = 34–97$).

Heuristic	Section	Reference	D_{avg}	D_{max}	Opt	Time
Double Outer-Scan	7.3.1	Wøhlk [51]	34.44	50.59	0	
Path-Scanning (PS)	7.2.2	Lacomme et al. [31]	16.27	35.10	0	1 ms
Modified-Path-Scanning	7.3.1	Wøhlk [51]	14.78	29.04	0	
A-ALG	7.3.1	Wøhlk [51]	12.31	27.65	0	
Node duplication heur	7.3.1	Wøhlk [51]	12.14	28.52	0	
Merge	7.3.2	Belenguer et al. [7]	12.06	20.25	0	2 ms
Augment-Merge	7.2.3	Belenguer et al. [7]	10.24	17.03	0	2 ms
Improved Merge	7.3.2	Belenguer et al. [7]	5.54	10.13	1	2 ms
URT-B (best 50)	7.3.3	Prins et al. [43]	9.16	17.22	0	3 ms
URTF-S (best 50)	7.3.3	Prins et al. [43]	5.91	12.63	0	4 ms
URTF-S (best 10,000)	7.3.3	Prins et al. [43]	2.81	9.85	6	0.68
DYPSA (best 5)	7.5.1	Wøhlk [51]	8.74	17.39	0	
CARPET	7.4.2	Hertz et al. [26]	1.51	8.10	17	26.61
TS	7.4.2	Belenguer et al. [6]	0.83	5.41	24	
TSA-1	7.5.4	Brandão & Eglese [12]	0.75	4.61	20	1.23
MA	7.5.6	Lacomme et al. [31]	0.18	2.08	29	15.98
MAENS (mean 30)	7.5.6	Tang et al. [47]	0.18	1.64	27	56.75
GRASP (mean 15)	7.5.3	Usberti et al. [49]	0.17	1.65	26	76.87
TSA-2	7.5.4	Brandão & Eglese [12]	0.14	1.54	30	11.78
RTS*	7.5.4	Mei et al. [38]	0.13	1.71	30	5.67
ACO (mean 15)	7.5.8	Santos et al. [46]	0.11	1.39	30	10.54
VNS (mean 10)	7.5.2	Polacek et al. [42]	0.08	0.83	26	[1]90.00
MAENS (best 30)	7.5.6	Tang et al. [47]	0.07	1.14	31	56.75
GRASP (best 15)	7.5.3	Usberti et al. [49]	0.07	1.04	30	76.87
GLS (5×10^5 iter)	7.5.5	Muyldermans [39]	0.05	0.77	31	16.94
ACO (best 15)	7.5.8	Santos et al. [46]	0.04	0.77	31	10.54
VNS (best 10)	7.5.2	Polacek et al. [42]	0.01	0.26	32	[1]90.00

[1]Corresponding to a fixed time of 60 s allocated to each run, on a 3.6 GHz PC.

Thank you!

方一栋
Email: Fangyd1997@outlook.com