香港城市大學
City University of Hong Kong

電腦科學系
Department of
Computer Science

Exact Methods for VRP

Fei Liu

Department of Computer Science
City University of Hong Kong

April 7, 2021

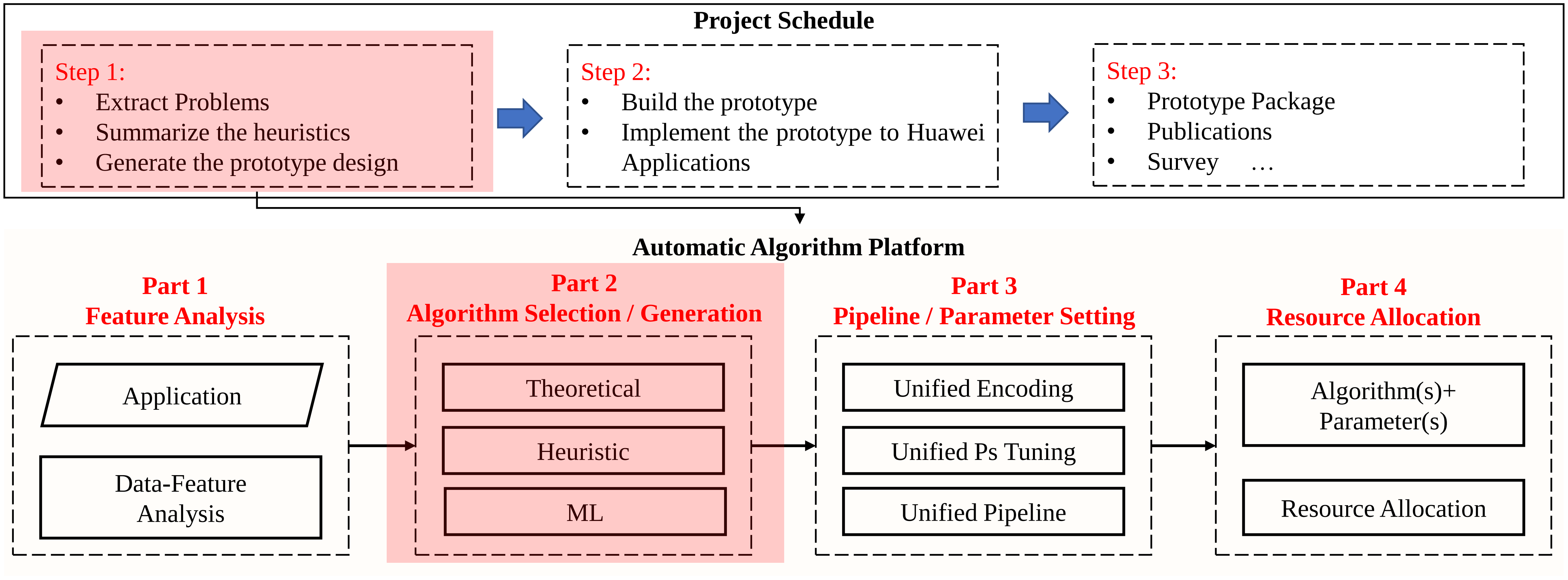
Outline

1. Roadmap
2. Exact Methods for VRP
3. Conclusion

Outline

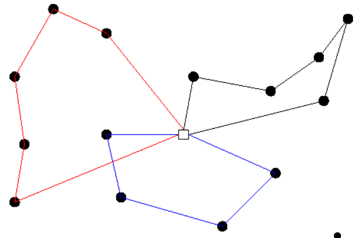
1. Roadmap
2. Exact Methods for VRP
3. Conclusion

Roadmap


Project Schedule
Step 1:
• Extract Problems
• Summarize the heuristics
• Generate the prototype design
Step 2:
• Build the prototype
• Implement the prototype to Huawei Applications
Step 3:
• Prototype Package
• Publications
• Survey …
Automatic Algorithm Platform
Part 1
Feature Analysis
Application
Data-Feature Analysis
Part 2
Algorithm Selection / Generation
Theoretical
Heuristic
ML
Part 3
Pipeline / Parameter Setting
Unified Encoding
Unified Ps Tuning
Unified Pipeline
Part 4
Resource Allocation
Algorithm(s)+ Parameter(s)
Resource Allocation

香港城市大學
City University of Hong Kong
電腦科學系
Department of Computer Science

Methods for VRP



Dantzig and Ramser (**1959**) the first to introduce **"Truck Dispatching Problem"**

Clarke and Wright (**1964**) generalized this problem to a linear optimization problem **"Vehicle Routing Problem"**

Lenstra and Rinnooy Kan (**1981**) proved VRP is **an NP-hard problem**, exact algorithms are only efficient for small problems

time

1950 — 1960 — 1970 — 1980 — 2000 — 2021

1. Constructive Heuristics：

Savings heuristic

Sweep algorithm

2. Improvement Heuristics：

K-opt

λ-interchange

3. Exact algorithms：

Branch and Bound

Cutting Plane

Network-flows

Dynamic Programming

4. Metaheuristics：

Tabu search

Genetic Algorithm

Simulated Annealing

GRASP

Local search methods

Partical Swarm Optimization

Ant Colony Algorithm

5. Machine Learning：

Reinforcement Learning

Pointer network

Perhaps the most famous heuristic of this category is the Clarke and Wright (**1964**) savings heuristic

The development of exact algorithms for the VRP took off in **1981** with the publication of two papers by Christofides

The development of modern heuristics for the VRP really started in the **1990s** with the advent of metaheuristics.

- Laporte, Gilbert, Paolo Toth, and Daniele Vigo. "**Vehicle routing: historical perspective and recent contributions.**" (2013): 1-4.

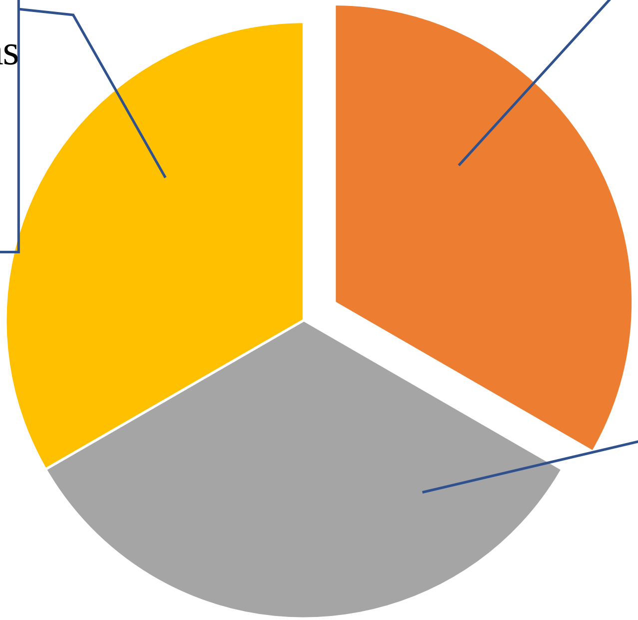
Topics of previous talks

- **2021.1.19**

Metaheuristics for Vehicle Routing Problems

- **2021.3.12**

A **Metaheuristic** Approach for 3L-SDVRP



- **2021.4.7**
- **Exact Methods** for VRP

- **2021.1.31**

ML for Combinational Optimization

We are trying to answer following two questions:

1. State-of-the-art exact methods

2. To what scale can the problem be solved by exact methods?

Outline

1. Roadmap
2. Exact Methods for VRP
3. Conclusion

Notations

For a CVRP problem we give the following notations:

We have a complete directed graph $G(V, A, c, x)$

Node set $V = \{0, 1, \cdots, n\}$ $V_C = V / \{0\}$

All arc set A A_C is the set where $\{i, j\} \in A$ with $i, j \in V_C$

Sub node $S \in V$

Sub arc set $\delta^+(S)$ denotes the set of arcs (i, j) with $i \in S$ and $j \in V / S$

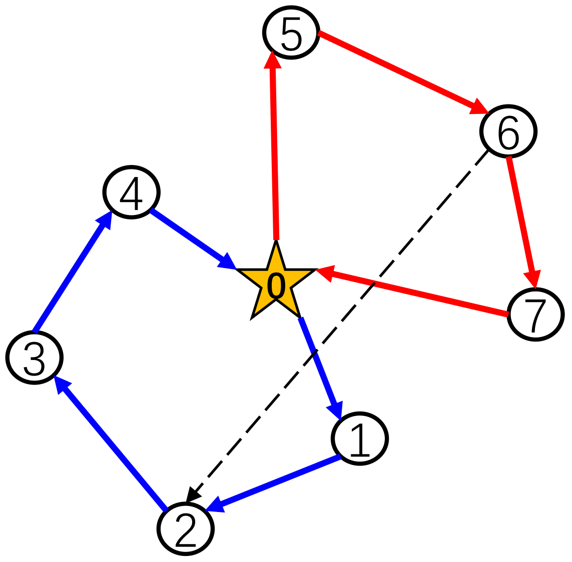
$\delta^-(S)$ denotes the set of arcs (i, j) with $j \in S$ and $i \in V / S$

V={0,1,2,3,4,5,6,7} A={(0,1), (1,2), (6,2),…}

V_C={1,2,3,4,5,6,7} A_C={(1,2), (6,2),…}

S={6,7} $\delta^+(S)$ ={(7,0), …}

$\delta^-(S)$ ={(5,6), …}



Integer programming formulations for VRP

The VRP problems together with its variants are formulated as integer programming formulations in order to implement the exact algorithms. Most existed formulations can be divided into four categories. They are:

1. Two-index, three-index vehicle-flow formulations
2. Set partitioning formulations
3. Single-commodity, two commodity, and multi-commodity flow formulations
4. Arc-packing formulation

Two-index formulation

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

s.t.

$$x(\delta^+(i)) = x(\delta^-(i)) = 1 \quad (\mathrm{i} \in \mathrm{V}_c) \qquad \text{(C1)}$$

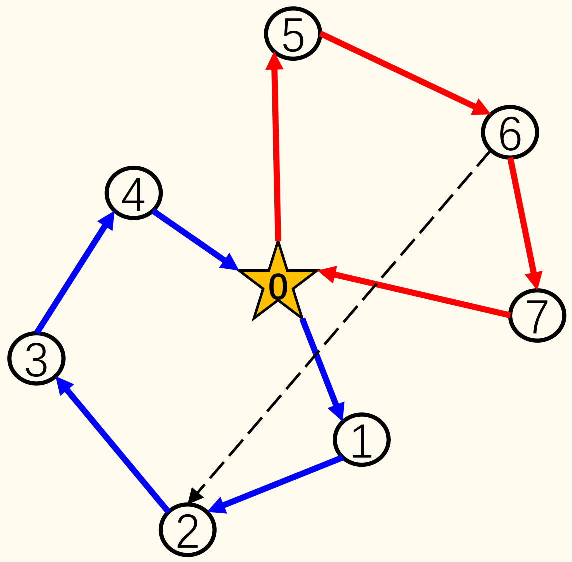
$$x(\delta^+(S)) \ge \lceil q(S)/Q \rceil \quad (\mathrm{S} \subseteq \mathrm{V}_c) \qquad \text{(C2)}$$

$$x_{ij} \in \{0,1\} \quad ((i,j) \in A) \qquad \text{(C3)}$$

- C1 are degree constraints
- C3 are binary constraints
- C2 are called rounded capacity (RC) inequalities

V={0,1,2,3,4,5,6,7}   A={(0,1), (1,2), (6,2),⋯}
$\mathrm{V_C}$={1,2,3,4,5,6,7} $\mathrm{A_C}$={(1,2), (6,2),⋯}

S={6,7} $\delta^+(S)$ ={(7,0), ⋯}
$\delta^-(S)$ ={(5,6), ⋯}



香港城市大學
City University of Hong Kong
電腦科學系
Department of Computer Science

Two-index formulation

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$s.t.$

$$x(\delta^+(i)) = x(\delta^-(i)) = 1 \quad (\mathrm{i} \in \mathrm{V}_c) \qquad \text{(C1)}$$

$$x(\delta^+(S)) \geq \lceil q(S)/Q \rceil \quad (\mathrm{S} \subseteq \mathrm{V}_c) \qquad \text{(C2)}$$

$$x_{ij} \in \{0,1\} \quad ((i,j) \in A) \qquad \text{(C3)}$$

- Subtour elimination (SE) inequalities[1]

$$x(\delta^+(S)) \geq 1 \quad (\mathrm{S} \subseteq \mathrm{V}_c)$$

- Fractional capacity (RC) inequalities [1]

$$x(\delta^+(S)) \geq q(S)/Q \quad (\mathrm{S} \subseteq \mathrm{V}_c)$$

- Rounded capacity (RC) inequalities [1]

$$x(\delta^+(S)) \geq \lceil q(S)/Q \rceil \quad (\mathrm{S} \subseteq \mathrm{V}_c)$$

- Generalized large multistar (GLM) inequalities [2]

$$x(\delta^+(S)) \geq \frac{1}{Q} \sum_{i\in S} (q_i + \sum_{j\in V_C \setminus S} q_j (x_{ij} + x_{ji})) \quad (\mathrm{S} \subseteq \mathrm{V}_c)$$

- Knapsack large multistar (KLM) inequalities [3]

$$x(\delta^+(S)) \geq \frac{1}{\beta} \sum_{i\in S} (\alpha_i + \sum_{j\in V_C \setminus S} \alpha_j (x_{ij} + x_{ji})) \quad (\mathrm{S} \subseteq \mathrm{V}_c)$$

$$\alpha^T y \leq \beta$$

$$conv\left\{ y \in \{0,1\}^n : \sum_{i\in V_C} q_i y_i \leq Q \right\}$$

[1] Naddef, Denis, and Giovanni Rinaldi. "Branch-and-cut algorithms for the capacitated VRP." *The vehicle routing problem*. Society for Industrial and Applied Mathematics, 2002. 53-84.
[2] Gouveia, Luis. "A result on projection for the vehicle routing problem." European Journal of Operational Research 85.3 (1995): 610-624.
[3] Letchford, Adam N., Richard W. Eglese, and Jens Lysgaard. "Multistars, partial multistars and the capacitated vehicle routing problem." Mathematical Programming 94.1 (2002): 21-40.

Set partitioning formulations

$$\min \sum_{r\in\Omega} c_r z_r$$

$s.t.$

$$\sum_{r\in\Omega} a_{ir} z_r = 1 \quad (\mathrm{i} \in \mathrm{V}_c) \qquad (\mathrm{C1})$$

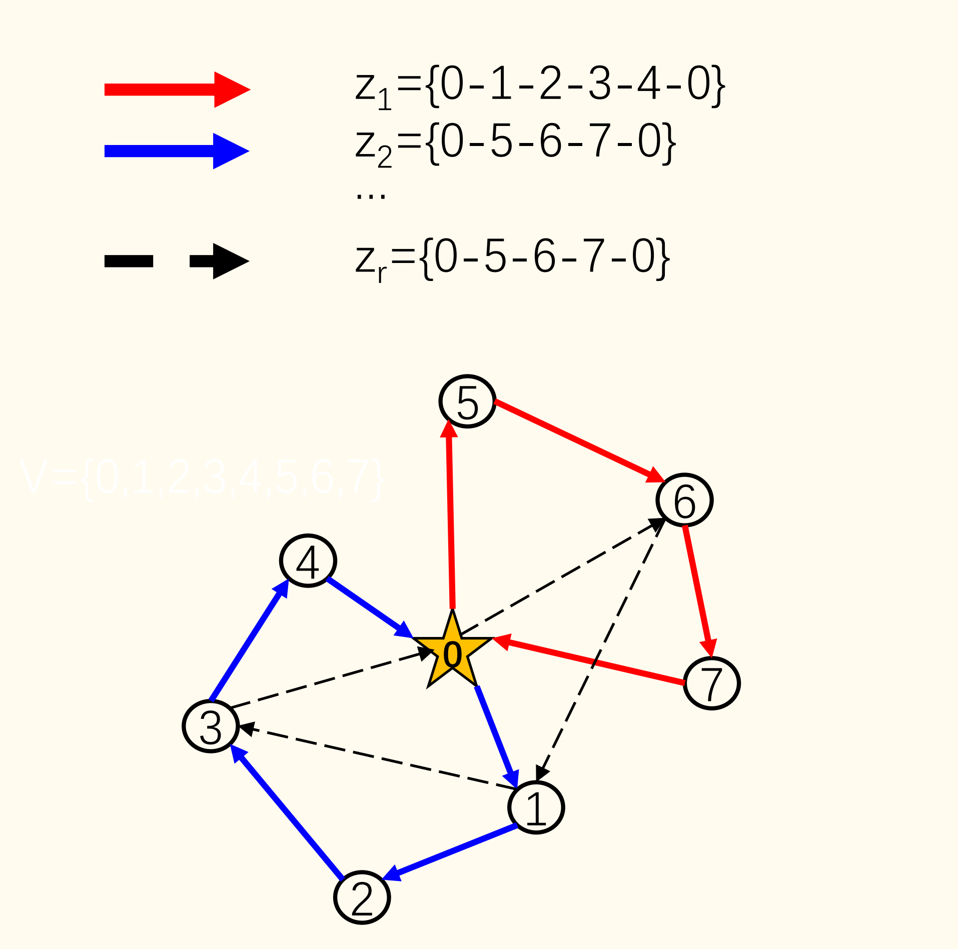
$$z_r \in \{0,1\} \quad (r \in \Omega) \qquad (\mathrm{C2})$$

Ω denote the set of possible routes

z_r is a binary variable, it takes 1 if route r is used

c_r denote the cost of route r

a_{ir} is a binary variable, it takes 1 if customer i is used in route r



Multi-commodity flow formulations (MCF2b)

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$s.t.$

$$x(\delta^+(i)) = x(\delta^-(i)) = 1 \quad (\mathrm{i} \in \mathrm{V}_c)$$

$$x_{ij} \in \{0,1\} \quad ((i,j)\in A)$$

$$f^k(\delta^+(0)) = f^k(\delta^-(k)) = g^k(\delta^+(k)) = g^k(\delta^-(0)) = 1 \quad (k \in V_C)$$

$$f^k(\delta^-(0)) = f^k(\delta^+(k)) = g^k(\delta^-(k)) = g^k(\delta^+(0)) = 0 \quad (k \in V_C)$$

$$f^k(\delta^-(l)) = f^k(\delta^+(l)) = g^l(\delta^-(k)) = g^l(\delta^+(k)) \quad (k,l \in V_C : l \neq k)$$

$$\sum_{k\in V_C\setminus\{i,j\}} q_k (f_{ij}^k + g_{ij}^k) \le (Q - q_i - q_j) x_{ij} \quad ((i,j)\in A)$$

$$f_{ij}^k + g_{ij}^k \le x_{ij} \quad (k \in V_C, (i,j)\in A)$$

$$f_{ij}^k, g_{ij}^k \in \{0,1\} \quad (k \in V_C, (i,j)\in A)$$

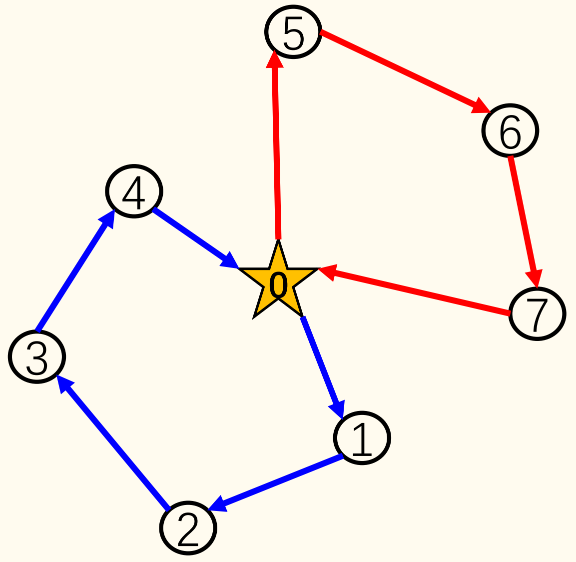
f_{ij}^k and g_{ij}^k indicate whether a vehicle traverses the arc (i , j) on the way to customer k or after visiting customer k , respectively.

$$f^6(\delta^+(0)) = f^6(\delta^-(6)) = g^6(\delta^+(6)) = g^6(\delta^-(0)) = 1$$

$$f^6(\delta^-(0)) = f^6(\delta^+(6)) = g^6(\delta^-(6)) = g^6(\delta^+(0)) = 0$$

$$f^6(\delta^-(7)) = f^6(\delta^+(7)) = g^7(\delta^-(6)) = g^7(\delta^+(6))$$

$$f_{56}^6 + g_{56}^6 \le x_{56}$$



Arc-packing formulation

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$s.t.$

$$x(\delta^+(i)) = x(\delta^-(i)) = 1 \quad (\mathrm{i} \in \mathrm{V}_c)$$

$$x_{ij} \in \{0,1\} \quad ((i,j) \in A)$$

$$f^k(\delta^-(0)) = f^k(\delta^+(0)) = g^k(\delta^-(0)) = g^k(\delta^+(0)) = 0 \quad (k \in V_C)$$

$$f^k(\delta^-(l)) = f^k(\delta^+(l)) = g^l(\delta^-(k)) = g^l(\delta^+(k)) \quad (k, l \in V_C : l \neq k)$$

$$\sum_{k\in V_C \setminus \{i,j\}} q_k (f_{ij}^k + g_{ij}^k) \leq (Q - q_i - q_j) x_{ij} \quad ((i,j) \in A)$$

$$f_{ij}^k + g_{ij}^k \leq x_{ij} \quad (k \in V_C, (i,j) \in A)$$

$$f_{ij}^k, g_{ij}^k \in \{0,1\} \quad (k \in V_C, (i,j) \in A)$$

$$x_{ij} = f_{ij}^j \quad (j \in V_C, i \in V \setminus \{j\})$$

$$x_{ij} = g_{ij}^i \quad (i \in V_C, j \in V \setminus \{i\})$$

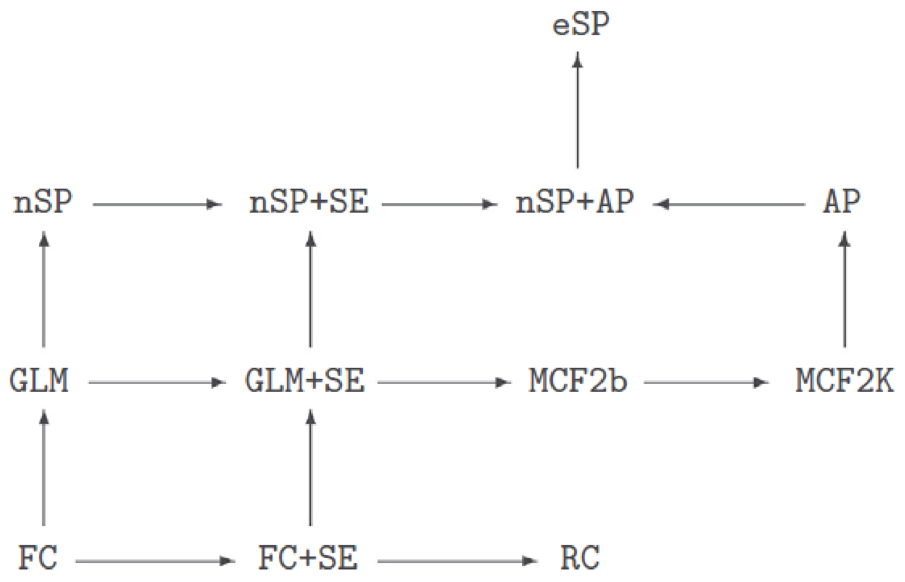
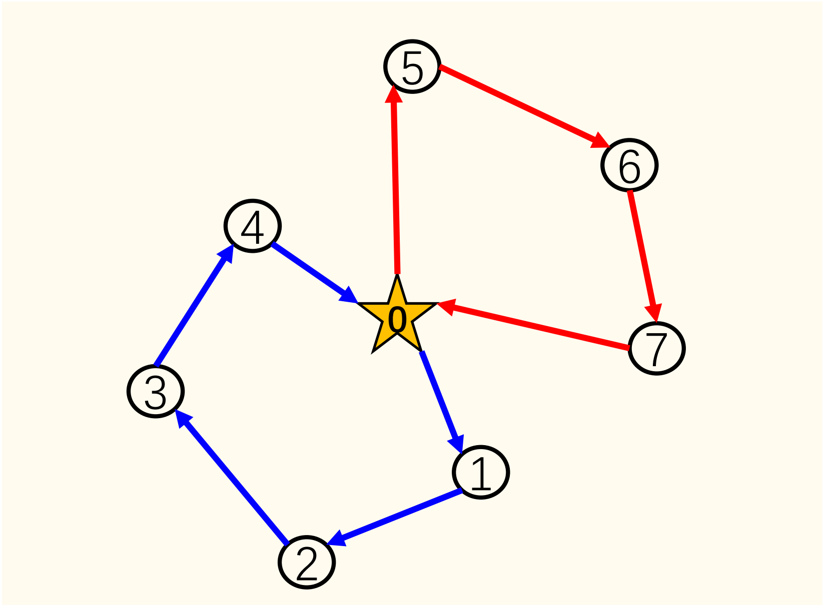


Fig. 1. Hierarchy of CVRP formulations.

[1] Letchford, Adam N., and Juan-José Salazar-González. "The capacitated vehicle routing problem: Stronger bounds in pseudo-polynomial time." *European Journal of Operational Research* 272.1 (2019): 24-31.

Exact algorithms

1. Integer linear programming
2. Dynamic programming
3. Branch-and-Bound
4. Branch-and-Cut
5. Branch-Cut-and-Price
6. ⋯



Key works about branch-cut-and-price (BCP)

[Balinski and Quandt, 1964]	Set-partitioning formulation for CVRP
[Laporte and Nobert, 1983]	Branch-and-bound
[Desrochers et al., 1992]	First branch-and-price
[Lysgaard et al., 2004]	Best branch-and-cut algorithm
[Fukasawa et al., 2006]	Robust branch-cut-and-price
[Baldacci et al., 2008]	Enumeration technique
[Jepsen et al., 2008]	(non-robust) subset-row cuts
[Baldacci et al., 2011b]	Ng-route relaxation
[Pecin et al., 2017b]	Best branch-cut-and-price
[Costa et al., 2019]	Recent surveys

Key works about branch-cut-and-price (BCP)

- [Balinski and Quandt, 1964]
- [Laporte and Nobert, 1983]
- [Desrochers et al., 1992]
- [Lysgaard et al., 2004]
- [Fukasawa et al., 2006]
- [Baldacci et al., 2008]
- [Jepsen et al., 2008]
- [Baldacci et al., 2011b]
- [Pecin et al., 2017b]
- [Costa et al., 2019]

Set-partitioning formulation for CVRP

$$\min \sum_{r\in\Omega} c_r z_r$$

$$s.t.$$

$$\sum_{r\in\Omega} a_{ir} z_r = 1 \quad (\mathrm{i}\in \mathrm{V}_c) \qquad \text{(C1)}$$

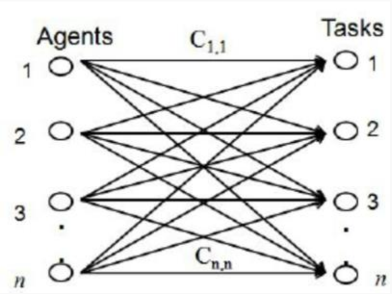
$$z_r \in \{0,1\} \quad (r\in\Omega) \qquad \text{(C2)}$$

Key works about branch-cut-and-price (BCP)

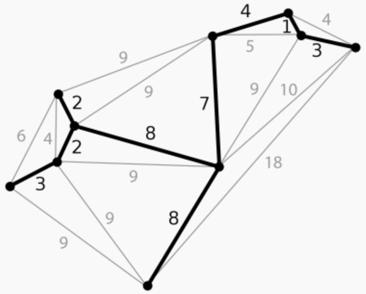
- [Balinski and Quandt, 1964]
- [Laporte and Nobert, 1983]
- [Desrochers et al., 1992]
- [Lysgaard et al., 2004]
- [Fukasawa et al., 2006]
- [Baldacci et al., 2008]
- [Jepsen et al., 2008]
- [Baldacci et al., 2011b]
- [Pecin et al., 2017b]
- [Costa et al., 2019]

MIP formulation with edge variables, rounded capacity cuts, and branch-and-bound

Assignment problem



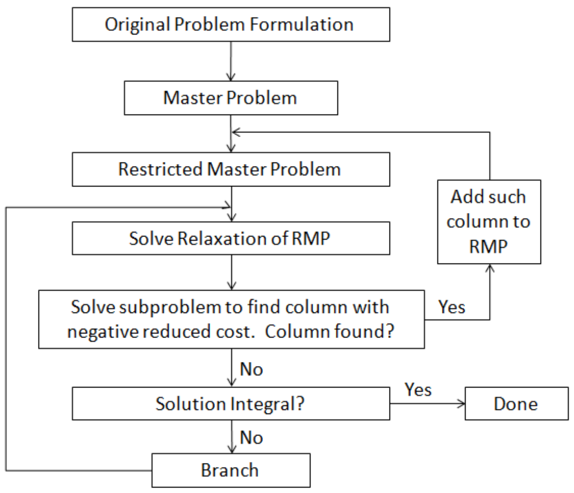
Shortest spanning tree



Key works about branch-cut-and-price (BCP)

- [Balinski and Quandt, 1964]
- [Laporte and Nobert, 1983]
- [Desrochers et al., 1992]
- [Lysgaard et al., 2004]
- [Fukasawa et al., 2006]
- [Baldacci et al., 2008]
- [Jepsen et al., 2008]
- [Baldacci et al., 2011b]
- [Pecin et al., 2017b]
- [Costa et al., 2019]

First branch-and-price



1. For a subset of paths $P' \subset P$, define a Restricted Master Problem (RMP), containing subset P' of variables λ
2. Solve RMP by an LP solver, obtain an optimal primal solution $(\bar{x}, \bar{\lambda})$ and dual solution $(\bar{\pi}, \bar{\mu})$.
3. Solve the pricing problem to verify whether there is a variable λ_p with a negative reduced cost:
$$\min_{p \in P} \sum_{a \in A} \bar{\pi}_a h_a^p - \bar{\mu}. \tag{1}$$
4. If solution value of (1) is negative, add one or several variables λ_p to (RMP) and go to stage 2
5. Otherwise, run a separation algorithm to find constrains $Bx \leq b$ violated by $\bar{x}$. If violated inequalities are found, add them to (RMP) and go to stage 2, otherwise stop.

[1] Sadykov, Ruslan. "Tutorial: Modern Branch-and-Cut-and-Price for Vehicle Routing Problems Plan of the talk." *INOC 2019-9th International Network Optimization Conference*. 2019.

Key works about branch-cut-and-price (BCP)

- [Balinski and Quandt, 1964]
- [Laporte and Nobert, 1983]
- [Desrochers et al., 1992]
- [Lysgaard et al., 2004]
- [Fukasawa et al., 2006]
- [Baldacci et al., 2008]
- [Jepsen et al., 2008]
- [Baldacci et al., 2011b]
- [Pecin et al., 2017b]
- [Costa et al., 2019]

Best branch-and-cut algorithm

- Uses capacity, framed capacity, generalized capacity, strengthened comb, multistar, partial multistar, extended hypotour inequalities, and classical Gomory mixed integer cuts.

$$
\begin{aligned}
&\min \sum_{(i,j)\in A} c_{ij} x_{ij} \\
&s.t. \\
&x(\delta^{+}(i)) = x(\delta^{-}(i)) = 1 \quad (\mathrm{i} \in \mathrm{V}_c) && \text{(C1)} \\
&x(\delta^{+}(i)) \geq \lceil q(S)/Q \rceil \quad (\mathrm{S} \subseteq \mathrm{V}_c) && \text{(C2)} \\
&x_{ij} \in \{0,1\} \quad ((i,j) \in A) && \text{(C3)}
\end{aligned}
$$

- At present, the most promising solution technique appears to be *branch-and-cut*. They all based on the so-called *two-index formulation*.

Key works about branch-cut-and-price (BCP)

- [Balinski and Quandt, 1964]
- [Laporte and Nobert, 1983]
- [Desrochers et al., 1992]
- [Lysgaard et al., 2004]
- [Fukasawa et al., 2006]
- [Baldacci et al., 2008]
- [Jepsen et al., 2008]
- [Baldacci et al., 2011b]
- [Pecin et al., 2017b]
- [Costa et al., 2019]

Robust branch-cut-and-price

- Combines branch-and-cut and branch-and-price

- The best exact algorithms for CVRP have been based on either branch-and-cut or Lagrangean relaxation/column generation
- The resulting branch-and-cut-and-price algorithm can solve to optimality all instances from the literature with up to 135 vertices

Key works about branch-cut-and-price (BCP)

- [Balinski and Quandt, 1964]
- [Laporte and Nobert, 1983]
- [Desrochers et al., 1992]
- [Lysgaard et al., 2004]
- [Fukasawa et al., 2006]
- [Baldacci et al., 2008]
- [Jepsen et al., 2008]
- [Baldacci et al., 2011b]
- [Pecin et al., 2017b]
- [Costa et al., 2019]

- Enumeration technique
- Subset-row cuts
- Ng-route relaxation

Key works about branch-cut-and-price (BCP)

- [Balinski and Quandt, 1964]
- [Laporte and Nobert, 1983]
- [Desrochers et al., 1992]
- [Lysgaard et al., 2004]
- [Fukasawa et al., 2006]
- [Baldacci et al., 2008]
- [Jepsen et al., 2008]
- [Baldacci et al., 2011b]
- [Pecin et al., 2017b]
- [Costa et al., 2019]

Best branch-cut-and-price

- Limit-memory method
- The best performing exact algorithms for CVRP developed in the last 10 years are based in the combination of cut and column generation

How big a VRP problem can exact methods solve

- Measuring it by the size where instances can be consistently solved, We observe an increase *from 50 to 200* customers.[1] **(2014)**
- All the instances used for benchmarking exact algorithms, *with up to 199 customers*, were solved to optimality. [2] **(2017)**
- Sophisticated branch-cut-and-price (BCP) algorithms for some of the most classical VRP variants now solve many instances with up to *a few hundreds* of customers.[3] **(2020)**
- Now most instances of the most classic VRPs with up to 200 customers can be solved, some of them in a long time. More importantly, instances with *up to 100* customers can often be solved in less than 1 minute. [3] **(2020)**

[1] Toth, Paolo, and Daniele Vigo, eds. Vehicle routing: problems, methods, and applications. Society for Industrial and Applied Mathematics, 2014.
[2] Pecin, Diego, et al. "Improved branch-cut-and-price for capacitated vehicle routing." Mathematical Programming Computation 9.1 (2017): 61-100.
[3] Pessoa, Artur, et al. "A generic exact solver for vehicle routing and related problems." Mathematical Programming 183.1 (2020): 483-523.

How big a VRP problem can (hybrid) heuristics (probably) solve

- It is safe to say that current metaheuristics are capable of producing high quality solutions for instances with *up to 500* customers.[1] **(2014)**
- Two heuristics are tested on newly proposed instances with *100-1000 customers*. [2] **(2017)**
- Two-stage tabu search used on VRPTW with *100-1000 customers*. [3] **(1999)**
- Local search method tested on new instances with *3000-30000 customers*. [4] **(2019)**

[1] Toth, Paolo, and Daniele Vigo, eds. Vehicle routing: problems, methods, and applications. Society for Industrial and Applied Mathematics, 2014.
[2] Uchoa, Eduardo, et al. "New benchmark instances for the capacitated vehicle routing problem." European Journal of Operational Research 257.3 (2017): 845-858.

[3] Gehring, Hermann, and Jörg Homberger. "A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows." Proceedings of EUROGEN99. Vol. 2. Springer Berlin, 1999.
[4] Arnold, Florian, Michel Gendreau, and Kenneth Sörensen. "Efficiently solving very large-scale routing problems." Computers & Operations Research 107 (2019): 32-42.

Time cost comparison[1]

Exact method (BCP)

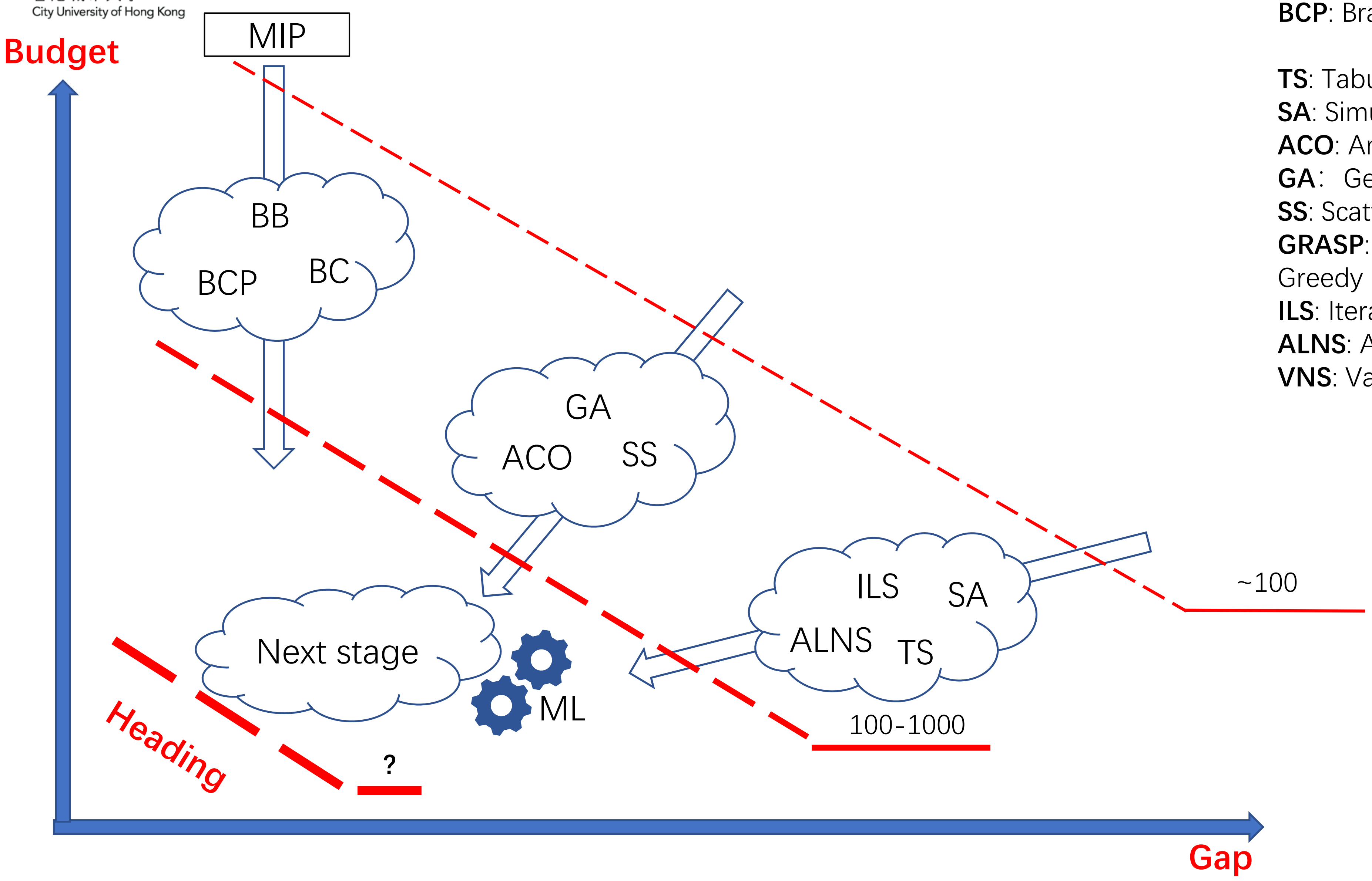
Table 5
BCP median times for each configuration.

	Hours	Days	Ratio		Hours	Days	Ratio
n				Demand			
100	0.26	0.01	0.03	U	0.19	0.01	0.02
125	0.2	0.01	0.02	1–10	4.41	0.18	0.45
150	1.36	0.06	0.14	5–10	9.73	0.41	1
175	2.99	0.12	0.31	1–100	99.51	4.15	10.23
200	9.73	0.41	1	50–100	183.01	7.63	18.81
225	40.85	1.7	4.2	Q	>312	>13	>32.07
250	34.77	1.45	3.57	SL	286.5	11.94	29.45
Dep				*r*			
C	1.42	0.06	0.15	[3,5]	1.37	0.06	0.14
E	53.15	2.21	5.46	[6,8]	2.03	0.08	0.21
R	9.73	0.41	1	[9,11]	9.73	0.41	1
				[12,14]	24.14	1.01	2.48
Cust				[15,16]	86.67	3.61	8.91
R	9.92	0.41	1.02	[17,18]	139.9	5.83	14.38
C	7.1	0.3	0.73	[21,23]	143.32	5.97	14.73
RC	9.73	0.41	1				

Heuristics

n	mins	hours
100-200	<10	/
200-500	10-60	<1
500-1000	60-800	1-13.3

[1] Uchoa, Eduardo, et al. "New benchmark instances for the capacitated vehicle routing problem." European Journal of Operational Research 257.3 (2017): 845-858.



MIP: Mixed integer programming
BB: Branch-and-Bound
BC: Branch-and-Cut
BCP: Branch-and-Cut-and-Price

TS: Tabu Search
SA: Simulated Anneal
ACO: Ant Colony Optimization
GA: Generic Algorithm
SS: Scatter Search
GRASP:
Greedy randomized adaptive search procedure
ILS: Iterated Local Search
ALNS: Adaptive Large Neighbourhood Search
VNS: Variable Neighbourhood Search

香港城市大學
City University of Hong Kong
電腦科學系
Department of
Computer Science

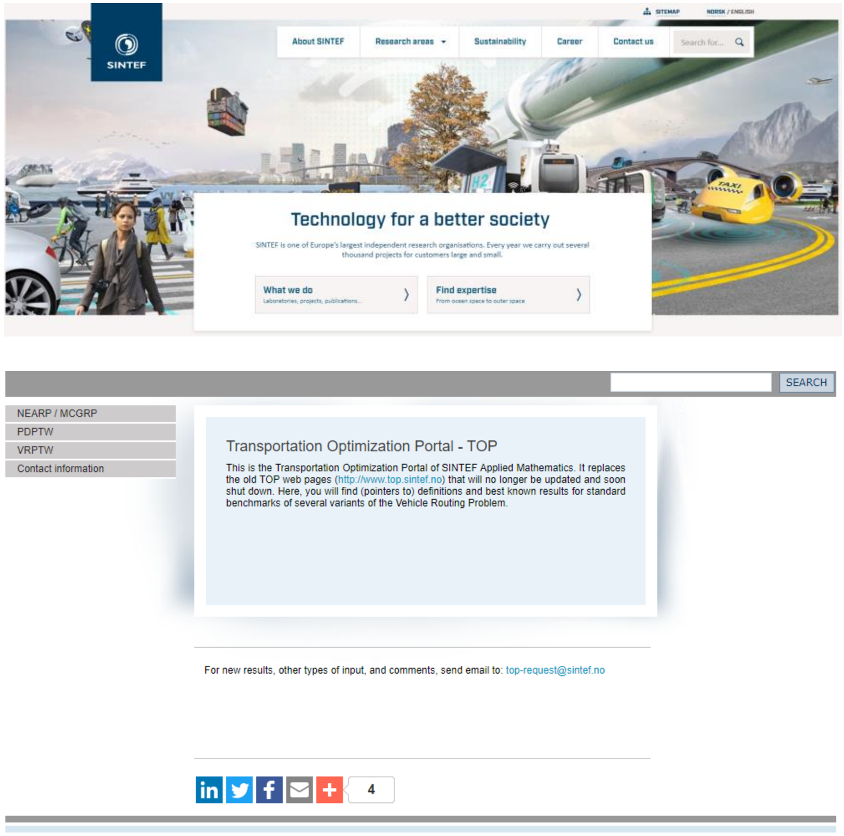
https://baijiahao.baidu.com/s?id=1685417905897714864&wfr=spider&for=pc

ALNS, Tabu, GLS

https://blog.csdn.net/yunqiinsight/article/details/89178260

ALNS,ML

Best known solution of SINTEF Benchmark



HUAWEI

lrc1_10_1	82	49111.78	SB	30-jan-19
lrc1_10_2	71	45547.38	HW	30-Nov-20
lrc1_10_3	53	35620.73	HW	30-Nov-20
lrc1_10_4	40	27213.93	HW	30-Nov-20
lrc1_10_5	72	50323.04	HW	26-Feb-21
lrc1_10_6	67	45115.22	HW	30-Nov-20
lrc1_10_7	60	41560.52	HW	23-Dec-20
lrc1_10_8	55	41063.60	HW	15-Jan-21
lrc1_10_9	53	39182.15	HW	30-Nov-20
lrc1_10_10	47	36552.46	HW	23-Dec-20
lrc2_10_1	22	34463.46	SCR	05-Nov-19
lrc2_10_2	19	38619.13	HW	10-Dec-20
lrc2_10_3	16	27218.08	HW	23-Dec-20
lrc2_10_4	11	23220.38	HW	26-Feb-21
lrc2_10_5	16	40848.54	HW	30-Nov-20
lrc2_10_6	17	30910.65	SCR	05-Nov-19
lrc2_10_7	15	33275.24	SCR	31-Dec-19
lrc2_10_8	-	-	-	-
lrc2_10_9	-	-	-	-
lrc2_10_10	11	29100.28	HW	23-Dec-20

CAINIAO 菜鸟

emapa

rc1_10_1	90	45830.62	CAINIAO	Dec-19
rc1_10_2	90	43718.84	SCR	Apr-19
rc1_10_3	90	42163.46	SCR	Apr-19
rc1_10_4	90	41397.81	SCR	Feb-20
rc1_10_5	90	45069.37	SCR	Jun-19
rc1_10_6	90	44944.95	SCR	Sep-19
rc1_10_7	90	44457.79	SCR	Feb-20
rc1_10_8	90	43956.91	SCR	Jan-20
rc1_10_9	90	43899.45	SCR	Apr-19
rc1_10_10	90	43573.95	SCR	Jan-20
rc2_10_1	20	30276.27	CAINIAO	Nov-18
rc2_10_2	18	26104.09	SCR	Oct-18
rc2_10_3	18	19911.48	SCR	Jan-20
rc2_10_4	18	15693.28	CAINIAO	Feb-19
rc2_10_5	18	27067.04	SCR	Jul-19
rc2_10_6	18	26741.27	CAINIAO	27-sep-18
rc2_10_7	18	24999.66	SCR	Jan-20
rc2_10_8	18	23595.33	SCR	Mar-19
rc2_10_9	18	22943.42	SCR	Mar-19
rc2_10_10	18	21834.94	SCR	Jul-19

PDPTW

VRPTW

Outline

1. Roadmap
2. Exact Methods for VRP
3. Conclusion

Conclusion

1. Branch-cut-and-price (BCP) is one of the state-of-the-art exact method
2. Exact methods can solve many instances with up to a few hundreds of customers
3. Use hybrid approach and ML techniques

電腦科學系
Department of
Computer Science

香港城市大學
City University of Hong Kong

Exact Methods for VRP

Thanks!

- **Fei Liu**
- **April 7, 2021**
- **Email: fliu36-c@my.cityu.edu.hk**