Vehicle routing methodologies to support costing and pricing decisions

Wout Dullaert, ITMMA, University of Antwerp
Olli Bräysy, University of Jyväskylä
Frans Cruijssen, TNT Express
Bruno De Borger, University of Antwerp

---

1. Introduction

- The more rigid the customer, the harder to design a cost efficient route
- Pricing strategies
  - Going-rate pricing
  - Preceived value pricing
  - Markup or cost-plus pricing
- Even when prices in the industry are not cost-based, information on incremental costs remains essential
  - Price floor
  - Determine profitability
- Undesirable to have completely customer-specific prices
  - Transaction costs
  - Fairness issues
  - Development of price structures
- Area of application: heterogeneous vehicle routing problem with time windows (FSMVRPTW)

Fleet Size and Mix Vehicle Routing Problem (FSMVRP)

- different vehicle types with different capacities and acquisition costs
- Objective: find a fleet composition and a corresponding routing plan that minimizes the sum of routing and vehicle costs.
- Practical applications of FSMVRP
- Various models exist in the literature depending on
  - how the variable costs and fleet size are issued
  - whether there are limits on the number of vehicles of each type
  - Best known objective function (Liu & Shen 1999): Vehicle cost + ”En route time” (constant sum of service time is excluded in reporting)

---

2. Short literature review

- Shared costs: when part of the costs cannot be traced back to a single customer or a single shipment.
  - Common costs:
  - Joint costs:
    - Game-theory in cost-allocation
    - DEA
    - …
- In vehicle routing incremental costs of a customer depends on customer characteristics and on the other customers’ characteristics

- Campbell and Savelsbergh (2005, Trans. Sci.): home delivery problem
  - Whether or not to accept delivery request upon arrival
  - All accepted requests must be serviced
  - Which time slot maximizes expected total profits
- Insertion based heuristics in GRASP framework
  - All accepted delivery requests are inserted where they maximize profits (select among *k* best locations)
  - Check whether the delivery requests can be serviced in one of the time slots of the customer (different criteria to assess the profitability)
  - To estimate expected profits: compare best value of VRPTW with or without customer involved

---

- Computational testing on randomly generated problem instances
  - impact of o.a. varying the number of allowable time slots, revenue per request and time slot width
  - Stringent time windows have a significant impact on routing costs and suggest pricing slots as interesting research avenue

---

- Campbell and Savelsbergh (2006, Trans. Sci.)
  - Adjust Campbell and Savelsbergh (2005)
  - maximize total profit=total revenue – total costs – total incentives paid to affect the probabilities of the time slots.
  - assumptions made on customer behavior (e.g. likelihood that customers select a time slot and effect of incentives on customers' behavior)
  - Simulation runs to assess the impact of future costs based on simple insertions in GRASP setting

---

- Confessore et al. (2007,IJPR)
  - Solomon's sequential insertion heurist I1
  - Artificial 100 customer problem instances
  - 16 scenarios with varying percentage of customers with tight time windows and the time window width of the remaining more flexible customers as well

$$TC = a_0 + a_1 \text{RelativeTWSize}$$

  - 78% to 97% of the variability in total cost for the 16 scenarios considered

- Our research objectives
  - Estimating the incremental cost of a customer
  - Identifying cost drivers

- Need for:
  - Powerful heuristic to calculate the cheapest solution for all customers
  - Different heuristics to estimate the solution after removing a customer (trade off computation time and solution quality)

---

3. A MSDA heuristic
(Bräysy et al. 2007)

- Multi-Start Deterministic Annealing (MSDA)
- 3 phases, embedded in restart loop
- Phase 1: Initial solution
- Phase 2: Route elimination
- Phase 3: Iterative improvement
  - 4 local search operators
  - Variable Neighborhood Descent until local optimum
  - Threshold Accepting until iteration limit, or no improvement limit
- First accept

- Adaptive memory of good arcs, utilized upon restart

Phase 1: generation of initial solutions

- Based on Savings (Clarke & Wright 1964)
- Savings based on total cost
- Merging route R1 into R2, all insertion points in R2 are tried
- Probabilistic insertion in one of the 3 best improving points
- Each route initialized with smallest possible vehicle type
- Greedy upgrade of vehicle type when needed

- Upon restart, some arcs from the final solution kept

Phase 2: route elimination

- Based on simple insertions, procedure ELIM
- All routes considered for depletion, in random order
- Customers tried in decreasing order of criticality

$$\varsigma_i = \frac{\eta(d_i)}{\eta(b_i - a_i)} + \eta(c_{0i})$$

- Best feasible insertion point w.r.t. total cost
- Cutoff when insertion cost exceeds elimination savings
- ELIM is run until quiescence

Phase 3: iterative improvement

- 4 local search operators iterated, First Accept
- ICROSS
  - Cross-exchange with reversal of segments
  - Heterogeneous fleet
  - Limited segment length
- IOPT: Or-opt extended with segment reversal
- ELIM: As in Phase 2 (every second iteration)
- SPLIT: All possible splits (every third iteration)

- Route sequence shuffled before each iteration
- Iterate until
  - local optimum, or no improvement over given # iterations
- Threshold Accepting
  - all moves except SPLIT
  - until iteration limit

- The procedure is restarted a given number of times
- Adaptive memory of arcs appearing in elite solutions
- New initial solution, start with current
  - Remove arcs not present in x % of the best solutions (e.g. 70 %)
  - Random removal of remaining arcs (e.g. 50 % probability)

MSDA - Four settings

- MSDA Quick
  - 200 iterations, 2 restarts, 1 run
- MSDA Medium-1
  - 1.000 iterations, 2 restarts, 1 run
- MSDA Medium-3
  - 1.000 iterations, 2 restarts, 3 runs
- MSDA Best
  - 10.000 iterations, 2 restarts, 3 runs

- Remaining parameter values are identical

Summary of Results

- MSDA Quick (~ 6.5 CPU seconds)
  - Outperformed on several individual instances
  - 1.6 % better than Dell'Amico et al., 13 times faster
  - 7.4 % better than Liu & Shen, 18 % slower
- MSDA Medium-1 (~ 70 CPU seconds)
  - 3.6 % better than Dell'Amico et al., 140 best known, 134 new
- MSDA Medium-3 (~ 210 CPU seconds)
  - 157 best known, 151 new
  - on average 3.9 % better than Dell'Amico et al., 2.5 times slower
  - on average 9.6 % better than Liu & Shen, 13 times slower
- MSDA Best (~ 660 CPU seconds)
  - 167 best known, 165 new
  - 1.1 % performance improvement over MSDA Medium-3, 10 times slower
- Type 2 instances are improved the most

8 approaches to estimating the incremental cost

1. Full re-optimization:
   - Construct new solution from scratch
   - Use MSDA for 1000 iterations
2. DA500: MSDA for 500 iterations
   - Maintain current structure after removing customer
   - MSDA for 500 iterations
3. DA100: MSDA for 100 iterations
   - Current solution structure + MSDA for 100 iterations
4. Local optimization
   - Current solution structure
   - MSDA for maximum 1000 iterations
   - No threshold for stage 3 local search: search ends when local optimum is found

---

5. Single route optimization
   - Consider using smaller vehicle after removing customer from route
   - Try IOPT to reduce distance in the route
6. Close re-optimization I
   - Similar to Local optimization (4)
   - Look for improvements within route from which customer was removed + from the neighboring routes
   - Distance limited adjusted during search, only consider customers within current distance limit
   - Use of 2 of 4 local search operators (ICROSS and IOPT)

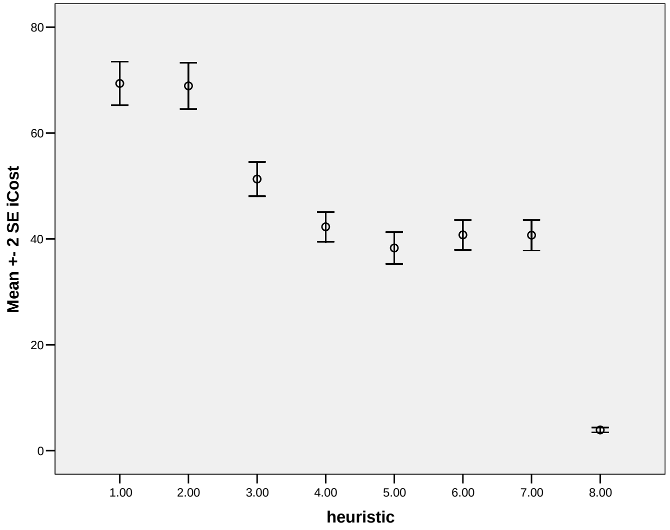
---

7. Close re-optimization II
   - Similar to close re-optimization II
   - Set of routes is extended by the routes that are closest to the route in which an improvement was obtained in the previous step
8. Simple removal
   - Relink route without changing the sequence of customers

Computational testing

- Liu and Shen (1999) benchmark:
  - 56 problem instances
  - R, C, RC subsets & 1 and 2 subsets
  - 100 customers per problem instance
- Restrict hardest problem instances: C103, C204, R104, R209, RC202, RC101
- Java JDK (5.0), AMD Athlon 2600+ (512 MB RAM) computer
- Short run and long run cost estimates: the higher, the more powerful the cost estimator

Short run average cost estimates



Short run costs: Tamhane T2 test

	1	2	3	4	5	6	7	8
1		-0.4557909	-18.067116	-27.084713	-31.076081	-28.603898	-28.666428	-65.446822
2	0.4557909		-17.611325	-26.628922	-30.62029	-28.148108	-28.210637	-64.991031
3	18.067116	17.611325		-9.0175973	-13.008965	-10.536783	-10.599313	-47.379706
4	27.084713	26.628922	9.0175973		-3.9913676	-1.5191854	-1.5817153	-38.362109
5	31.076081	30.62029	13.008965	3.9913676		2.4721822	2.4096524	-34.370742
6	28.603898	28.148108	10.536783	1.5191854	-2.4721822		-0.0625298	-36.842924
7	28.666428	28.210637	10.599313	1.5817153	-2.4096524	0.0625298		-36.780394
8	65.446822	64.991031	47.379706	38.362109	34.370742	36.842924	36.780394	

Hit rates and time consumption

	short run costs			Long run costs (cost structure A)		
	overall best	best of 7	total CPU	overall best	best of 7	total CPU
1	0.5533		176.32	0.6117		341.70
2	0.4533	0.9917	92.95	0.3883	0.9983	196.43
3	0.0467	0.1617	18.59	0.0317	0.1150	39.10
4	0.0100	0.0650	1.14	0.0083	0.0433	3.03
5	0.0033	0.0200	0.01418	0.0000	0.0017	0.01457
6	0.0050	0.0317	0.44900	0.0033	0.0150	0.63700
7	0.0033	0.0350	0.48500	0.0083	0.0417	1.54500
8	0.0000	0.0050	0.00093	0.0000	0.0017	0.00162

Explaining incremental costs

- Examining cost structure for
  - Pricing purposes
  - Examining profitability of customers
  - Estimating cost of scheduling inflexibility of customers

- cost drivers to variation in DA500 cost estimates:
  - size of customer demand,
  - distance from the depot,
  - width of the service time window
  - Relative with of the service time window
  - number of customers located within 5, 10, 20, 30 and 50% of the maximum distance in the problem (3 implementations).

Regression results

Approximating cost high quality cost estimators

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig	Correlations			Collinearity Statistics	
		B	Std. Error	Beta			Zero-order	Partial	Part	Tolerance	VIF
1	(Constant)	18.000	1.298		13.871	.000					
	Single_route	1.330	.025	.912	54.266	.000	.912	.912	.912	1.000	1.000
2	(Constant)	14.619	1.413		10.348	.000					
	Single_route	.777	.105	.533	7.432	.000	.912	.291	.122	.052	19.089
	close_route1	.602	.111	.389	5.424	.000	.908	.217	.089	.052	19.089

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.912(a)	.831	.831	21.96072	.831	2944.792	1	598	.000
2	.916(b)	.839	.839	21.45679	.008	29.418	1	597	.000

a Predictors: (Constant), single_route
b Predictors: (Constant), single_route, close_route1

Conclusions

- 8 different approximations for the incremental cost of a customer.
  - approaches clearly differ with respect to solution quality and CPU time requirements.
  - DA500: most powerful approach next to a time consuming full re-optimization approach.
- even a wide set of possible cost drivers cannot sufficiently model the incremental cost of a customer.
- approximating incremental costs either by DA500 or by a regression equation based on even faster cost estimators is a more fruitful approach to estimate incremental costs in real-life routing problems.

- Toevoegen slides ideeen Bruno.

Backup slides

Table 2. Comparison between different approaches for the Liu and Shen (1999) variant

Data set	Liu & Shen 99 total cost	Dullaert et al. 02 en route time	Dell'Amico et al. 06 total cost	MSDA Quick en route time	MSDA Quick total cost	MSDA Medium en route time	MSDA Medium total cost	MSDA Best en route time	MSDA Best total cost
R1A	4398	1548.53	4180.83	1581.04	4199.36	1551.86	4154.36	**1535.47**	**4131.31**
R1B	2054	1557.38	1927.57	1321.07	1942.40	1285.95	1910.11	**1284.21**	**1898.88**
R1C	1700	1557.85	1615.44	1283.39	1611.98	1255.53	1591.12	**1252.25**	**1579.17**
C1A	8007	1166.09	7229.02	1508.15	7208.15	1451.72	7151.72	**1441.15**	**7141.15**
C1B	2485	1126.01	2384.77	1082.66	2447.11	977.69	2373.25	**978.82**	**2365.49**
C1C	1705	1155.45	1629.70	937.33	1685.11	857.92	1629.59	**853.50**	**1621.83**
RC1A	5184	1665.04	5117.96	1640.86	5053.36	1584.56	4997.06	**1539.78**	**4948.53**
RC1B	2235	1680.55	2163.51	1445.23	2180.98	1409.70	2140.95	**1393.85**	**2126.60**
RC1C	1849	1689.92	1784.51	1413.83	1791.83	1382.47	1765.72	**1376.92**	**1758.29**
R2A	3809	1426.52	3568.97	1194.07	3471.34	1114.93	3392.20	**1060.70**	**3310.70**
R2B	1797	1431.49	1727.04	1110.23	1604.77	1066.82	1535.00	**1037.19**	**1495.37**
R2C	1513	1419.81	1436.22	1108.43	1358.43	1028.13	1279.03	**1018.56**	**1257.65**
C2A	6717	821.38	6267.75	935.71	6135.71	775.62	5975.62	**772.38**	**5797.38**
C2B	1970	821.38	1897.62	813.13	1873.13	719.37	1769.37	**711.08**	**1756.08**
C2C	1288	811.16	1276.29	772.04	1322.04	691.61	1234.11	**673.86**	**1223.86**
RC2A	5273	1800.82	4752.95	1610.79	4498.29	1524.37	4430.62	**1449.12**	**4399.12**
RC2B	2324	1741.97	2156.11	1377.39	2027.39	1274.21	1925.46	**1242.95**	**1899.20**
RC2C	1978	1741.75	1828.95	1332.76	1674.01	1242.37	1583.62	**1224.69**	**1562.19**
Average	3074.11	1406.20	2892.24	1251.51	2847.27	1183.63	2780.40	**1164.77**	**2749.17**
Computer	P 16.5	N/A	P 85.0	AMD 2600	AMD 2600	A 210.8 2600	A 210.8 2600	1975.2 2600	1975.2 2600
Runs	3	1	1	1	1	3	3	3	3
Average CPU seconds per instance	163.4	N/A	849.67	6.49	6.49	70.29	70.29	658.40	658.40

MSDA Medium-1 vs Dell'Amico et al.

	% impr.	# Better	# Ties	# Worse
R1A	0.43	6	0	6
R1B	0.30	8	0	4
R1C	1.08	12	0	0
R1	**0.60**	**26**	**0**	**10**
C1A	0.70	6	0	3
C1B	-0.39	3	0	6
C1C	-0.20	1	6	2
C1	**0.04**	**10**	**6**	**11**
RC1A	2.18	8	0	0
RC1B	0.76	8	0	0
RC1C	0.53	6	0	2
RC1	**1.16**	**22**	**0**	**2**
R2A	3.00	9	0	2
R2B	10.39	10	0	1
R2C	9.59	11	0	0
R2	**7.66**	**30**	**0**	**3**
C2A	0.70	8	0	0
C2B	5.80	8	0	0
C2C	2.13	7	0	1
C2	**2.88**	**23**	**0**	**1**
RC2A	6.58	7	0	1
RC2B	10.01	8	0	0
RC2C	11.98	8	0	0
RC2	**9.53**	**23**	**0**	**1**
All	**3.58**	**134**	**6**	**28**

---

number of iterations, $n_{improve} = 1000$;

number of initial solutions, $n_{init} = 2$;

maximum threshold, $t_{\max} = 0.04$;

maximum segment length in ICROSS and IOPT, $l_s = 3$;

The threshold change step, $\Delta t = 0.001$;

restart limit $\bar{n} = 40$ iterations;

record arc frequencies from $n_{\lim it} = 5$ iteration.

Probabilities savings heuristic

Type I: $p_1 = 0.5$, $p_2 = 0.25$, $p_3 = 0.25$

Type II: $b = 0.7$ and $p_d = 0.5$.

	Cost structure A			Cost structure B			Cost structure C		
Problem	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
R1	4162.92	4163.68	4163.95	1921.81	1917.98	1915.63	1598.07	1595.52	1596.19
C1	7178.72	7168.17	7160.72	2394.05	2388.28	2400.89	1632.94	1633.19	1632.80
RC1	5006.36	5009.01	5009.34	2147.06	2157.91	2152.21	1775.12	1775.80	1781.62
R2	3461.98	3461.59	3482.59	1547.63	1567.70	1563.86	1298.54	1304.94	1284.21
C2	6006.45	6002.23	5987.81	1787.51	1790.97	1784.68	1249.10	1242.89	1247.21
RC2	4440.07	4456.28	4450.65	1940.30	1943.26	1947.37	1609.76	1608.29	1603.53
CPU/ s	89.90	91.79	90.21	62.32	59.57	61.01	54.02	53.61	54.54

MSDA Best: best result over three runs using above parameter setting
MSDA Quick: one run using one initial solution and 200 iterations, other parameter values remaining the same.

Data set	Liu & Shen (99) total cost	Dullaert et al. (02) schedule time	Dell'Amico et al. (06) total cost	MSDA Quick schedule time	MSDA Quick total cost	MSDA Best schedule time	MSDA Best total cost	MSDA Best distance	MSDA Best total cost (with distance)
R1A	4398	1548.53	4180.83	1581.04	4199.36	1551.86	4154.36	1466.84	4071.01
R1B	2054	1557.38	1927.57	1321.07	1942.40	1285.95	1910.11	1245.87	1855.37
R1C	1700	1557.85	1615.44	1283.39	1611.98	1255.53	1591.12	1218.59	1541.09
C1A	8007	1166.09	7229.02	1508.15	7208.15	1451.72	7151.72	1388.27	7088.27
C1B	2485	1126.01	2384.77	1082.66	2447.11	977.69	2373.25	952.17	2336.62
C1C	1705	1155.45	1629.70	937.33	1685.11	857.92	1629.59	838.66	1616.44
RC1A	5184	1665.04	5117.96	1640.86	5053.36	1584.56	4997.06	1557.21	4958.46
RC1B	2235	1680.55	2163.51	1445.23	2180.98	1409.70	2140.95	1391.38	2125.63
RC1C	1849	1689.92	1784.51	1413.83	1791.83	1382.47	1765.72	1363.97	1743.47
R2A	3809	1426.52	3568.97	1194.07	3471.34	1114.93	3392.20	946.64	3196.64
R2B	1797	1431.49	1727.04	1110.23	1604.77	1066.82	1535.00	942.95	1398.41
R2C	1513	1419.81	1436.22	1108.43	1358.43	1028.13	1279.03	924.53	1157.26
C2A	6717	821.38	6267.75	935.71	6135.71	775.62	5975.62	691.55	5691.55
C2B	1970	821.38	1897.62	813.13	1873.13	719.37	1769.37	678.89	1698.89
C2C	1288	811.16	1276.29	772.04	1322.04	691.61	1234.11	656.16	1186.16
RC2A	5273	1800.82	4752.95	1610.79	4498.29	1524.37	4430.62	1247.81	4247.81
RC2B	2324	1741.97	2156.11	1377.39	2027.39	1274.21	1925.46	1062.24	1708.49
RC2C	1978	1741.75	1828.95	1332.76	1674.01	1242.37	1583.62	1040.17	1376.42
Average	3074.11	1406.20	2892.24	1251.51	2847.27	1183.63	2780.40	1096.77	2679.48
Computer	P 233	N/A	P 600	AMD 2600	AMD 2600	AMD 2600	AMD 2600	AMD 2600	AMD 2600
Average CPU sec	494	N/A	849.67	6.49	6.49	70.29	70.29	71.39	71.39

Conclusions

- MSDA offers high quality solutions within reasonable computation times
- 157 best-known solutions to the 168 test problems.
- introduced a new variant of the objective function that seems equally industrially relevant: namely the sum of total distance and fixed vehicle costs.