The Cost of Scheduling Customers in Routing Problems

Wout Dullaert
Institute of Transport and Maritime Management Antwerp,
University of Antwerp, Keizerstraat 64, 2000 Antwerp, Belgium, wout.dullaert@ua.ac.be

Olli Bräysy
Agora Innoroad Laboratory, Agora Center,P.O. Box 35, FI-40014 University of Jyväskylä, Finland,, Olli.Braysy@jyu.fi

Amir Salehipour
Department of Applied Economics, University of Antwerp, Antwerp, Belgium, amir.salehipour@ua.ac.be

The cost of servicing customers in routing problems tends to differ according to the characteristics of their service. In this paper, several methods are developed for determining the incremental cost of a customer in a heterogeneous routing problem with time windows and identifying its cost drivers.

Keywords: vehicle routing, incremental costs, scheduling

1. Introduction

Both for less-than-truckload (LTL) and full-truckload (FTL) routing and scheduling, customers tend to differ a lot with respect to the flexibility they give to the dispatcher to design routes. The more flexibility they give, the more cost efficient the routes the dispatcher can design. In practice, the cost-driving effect of scheduling inflexibility has been recognized for a longer time. E.g., express delivery companies charge higher rates for customers requiring a faster delivery. In most cases, reliable cost estimates of the various sources of scheduling inflexibility are unavailable.

Correct estimates of the incremental cost of customers or customer types are nevertheless essential for pricing routing services and accepting new customers. The additional cost of routing a customer acts as a price-floor: the freight rate should at least cover the additional cost of servicing that customer. This information remains valuable, even if prices in the industry are not cost-based. If prices are aimed at capturing customers' willingness to pay (e.g., perceived value pricing) or if prices are based on competitors' price levels (e.g., going-rate pricing), the incremental cost of servicing a customer is vital for determining the contribution or profitability of servicing each customer.

In this paper, several methods for determining the incremental cost of a customer in a vehicle routing problem with heterogeneous vehicles in which service at customers is restricted to prespecified hard time windows. The Fleet Size and Mix Vehicle Routing Problem (FSMVRP) is a Vehicle Routing Problem (VRP, see Toth and Vigo [17]) where the homogeneous fleet assumption of the traditional VRP has been lifted. A number of vehicle types with different capacities and acquisition costs are given. The objective is to find a fleet composition and a corresponding routing plan that minimizes the sum of routing and vehicle costs. For reviews on the FSMVRP, we refer the reader to Salhi and Rand [14], Osman and Salhi [13], and Lee et al. [10].

Section 2 presents 8 different approximation methods for estimating the incremental cost of a customer. They are extensively tested on problem instances for heterogeneous routing problems in Section 3. This section also looks at factors influencing the incremental costs of customers. Section 4 concludes the paper.

2. Approximations for incremental costs in routing

To determine the exact incremental cost of routing a customer two routing problems have to be solved to optimality: once with and once without the customer involved. The difference yields the additional cost generated by that customer. Exact solution algorithms for routing and scheduling are only capable of solving relatively small problem instances (Cordeau et al. [8]). For real-life applications, one must resort to heuristic procedures to approximate the optimal solution (without quality guarantee) within a reasonable amount of computation time. For the determination of a customer's incremental cost, these computation time considerations are even more an issue given the large number of computations involved.

Re-solving an entire routing problem for each customer is a computationally expensive procedure to determine the incremental routing costs. For a small routing problem of 100 customers, 100 routing problems of 99 customers have to be solved to determine the incremental cost of each customer. In real-life applications, the number of customers to be routed daily is a lot larger. For strategic decisions, company activity probably needs to be simulated of a large number of days to obtain reliable cost estimates. The number of routing problems that needs to be solved to obtain the incremental cost of different categories of customers soon becomes unmanageable, even for the fast, high-quality metaheuristics available today. Therefore more efficient ways of estimating the incremental routing cost of customers are suggested in this paper.

In this paper, we evaluate different approaches for determining the incremental cost. Full and partial re-optimization approaches are compared to naïve re-optimization procedures that simply remove a customer from the route without changing the sequence of the other customers in the route. Partial re-optimization procedures, reschedule only a part of the original routing problem. Clearly, the route containing the customer under consideration needs to be rescheduled as the customer is temporarily removed. But also the customers of a number of adjacent routes should be included in the analysis. Once the number of routes to be rescheduled is determined, local search heuristics can be used to re-optimize the routes. The re-optimization procedure that we applied here is the same as for the full re-optimization process.

The routing costs of the initial routing problem with all customers involved acts as a benchmark for all cost approximations. It should therefore be solved with the best possible solution method available. The recently developed multi-start deterministic annealing (MSDA) heuristic for the FSMVRPTW (Bräysy et al., [7]) obtained 167 best known solutions (of which 165 new best solutions) for the 168 problem instances from the literature. The heuristic is also time efficient, making it well-suited to calculate the incremental cost of each customer. The metaheuristic comprises three phases. In the first phase high quality initial solutions are generated by means of a savings-based heuristic combining diversification strategies with learning mechanisms. In phase two an attempt is made to reduce the number of routes in the initial solution with a new local search procedure and in phase three the solution from phase two is further improved by a set of four local search operators that are embedded in a deterministic annealing framework to guide the improvement process. For more details, the reader is referred to Bräysy et al. [7].

From a managerial point of view vehicle costs can be ignored in the so-called 'short run', when capacity cannot be adjusted. In the long run, by definition, all inputs of the production process are variable and therefore vehicle costs should be part of the objective function. The heuristic of Bräysy et al. [7] was adapted to be able to minimize both objective functions. By using a number of new implementation strategies and efficient (time window) feasibility checks, computation time is small compared to the solution quality obtained. We studied the following 8 metrics:

(1) Full re-optimization: In full re-optimization each customer is removed from the set of customers and a new solution is build from scratch. The total routing costs of the full set minus the total routing costs of the reduced set of customers, probably yields the best possible cost estimate of the incremental cost of that particular customer.

(2) DA500 and (3) DA100: For DA500 and DA100, the structure of the initial routing problem is maintained after removing the customer under consideration. After removing the customer, the deterministic annealing phase of the MSDA heuristic (phase 3) is executed for respectively 500 or 100 iterations.

(4) Local optimization: In the original implementation of the final improvement stage of the MSDA heuristic (phase 3), a threshold is used to allow for a controlled escape from local optima.

For the local minimum cost estimate, this threshold is not used as the search ends as soon as a local optimum is found. Except for this modification, the local minimum cost approximation is identical to the original improvement phase of the MSDA heuristic for the FSMVRPTW.

(5) Single route optimization: After removing a customer, Single Route Optimization checks whether the route can be serviced by a smaller vehicle. If so, possible savings in distance and vehicle cost are recorded. Then IOPT is applied to this tour only and additional distance savings are recorded. The IOPT intra-tour operator is a generalization of Or-opt (Or [12]). It considers segments of any length and also includes moves where the segment is reversed before it is relocated (Bräysy [1]). The reported value is the sum of the savings generated by the possible use of a smaller vehicle and the distances savings obtained by IOPT.

(6) Close re-optimization I: This approximation method is similar to the local optimization method (4) except the fact that we look for improvements within the route from which the customer was removed and its neighboring routes only within a distance limit that is adjusted during the search. As described in Bräysy et al. [7], the distance limit for the initial solution is set to the distance of the farthest customer from the depot to represent some kind of potential maximum arc length in a solution, multiplied by a uniform number r (between 0 and 1). During the search information is gathered on the maximum distances that have still made improvements possible. The maximum distance that has still enabled improvement is used as the new reference and multiplied by *r* to get the new distance limit.
For Close Re-optimization I, only customers within the current distance limit are taken into consideration. Of the four possible local search operators from stage 3 of the MSDA heuristic, the route splitter is not used and route elimination is only used in the beginning of the search. The search is mainly based on the IOPT and ICROSS operators. Cross-exchanges (Taillard et al. [16]) relocate or exchange segments of consecutive customers from different routes while preserving orientation. The ICROSS neighborhood (Bräysy et al. [2]) used here extends the neighborhood by also including moves where the current segment is reversed before insertion and allowing for adjustments of vehicle types.
To increase the efficiency of the improvement phase, ICROSS and IOPT do not consider a route pair or a single route if no improvement was found last time and the routes have not changed since. So if customer 1 was removed from route 1, and there are e.g., 3 routes close to it (e.g., 3,4,5) , only route pairs (1,3), (1,4) and (1,5) are evaluated by ICROSS and IOPT.

(7) Close Re-optimization II: Close Re-optimization II is similar to Close Re-optimization I. The only difference is that set of routes considered for re-optimization is adjusted during the search. In the first stage, only routes that are located within the distance limit are considered for re-optimization. In the next stage, the set is extended by routes that are closed to routes for which an improvement was obtained in the previous step. This process is repeated for as long as new improvements can be found in the newly defined sets of routes. So e.g., in the above example, if an improvement was found for the route pair (1,3), also combinations with routes close to route 3 will be considered etc.
(8) Simple removal: Simple removal is clearly the most naïve cost estimate. It simply consists of re-linking the current route after removing the customer under consideration, without changing the sequence of the customers. Only cost differences related to distance savings are recorded.

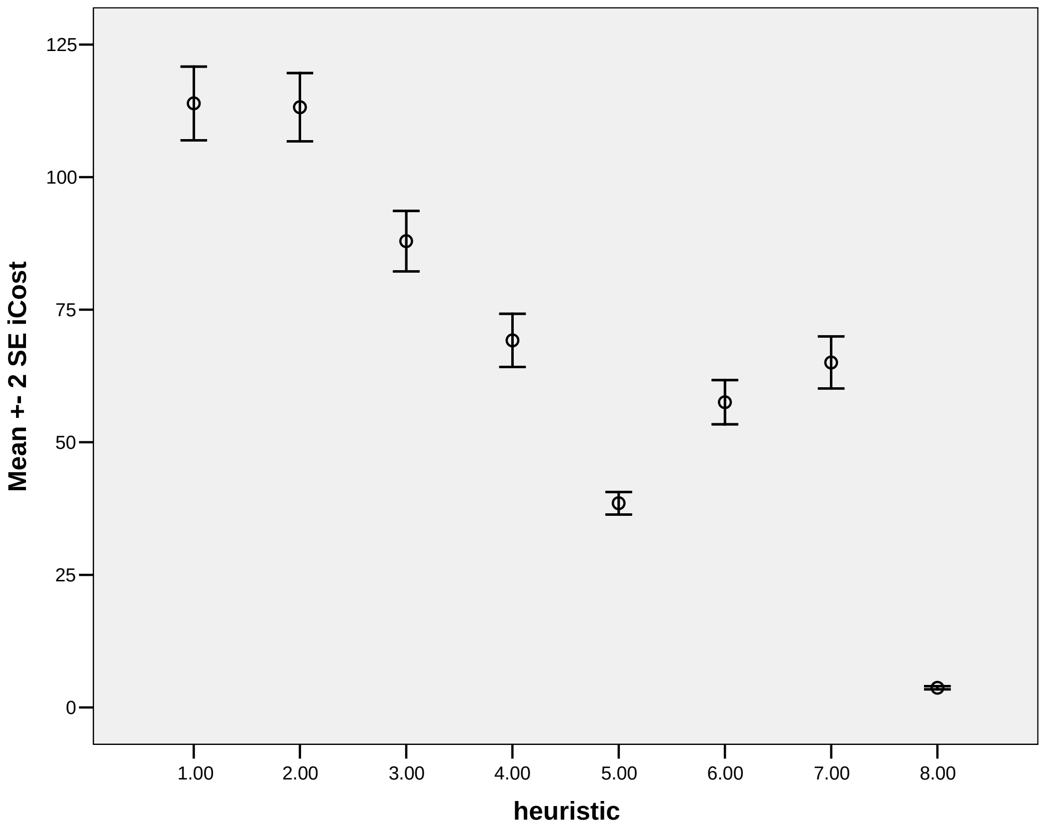
3. Computational testing

The cost approximations were implemented in Java (JDK 5.0) and tested on an AMD Athlon 2600+ (512 MB RAM) computer. The computational experiments were performed on the benchmark instances proposed by Liu and Shen [11], derived from the well-known VRPTW instances of Solomon [15]. Solomon’s problem sets for the VRPTW consist of 56 instances of 100 customers with randomly generated coordinates (set R), clustered coordinates (set C) or both (semi-clustered RC set). Subsets R1, C1 and RC1 have a short scheduling horizon and small vehicle capacities. By contrast, subsets R2, C2 and RC2 may have routes with a larger number of customers. For each six subsets Liu and Shen introduced several vehicle types with different capacities and costs. In addition, three different vehicle cost structures A, B and C were suggested, resulting in total to 168 problem instances. Here cost structure A refers to largest vehicle costs and C to the smallest.

3.1. Long-run incremental costs

Due to space limitations, we here only report on long-run incremental cost estimates. In the long-run, capacity can be adjusted and as a consequence, both vehicle and distance related costs should be reflected in the incremental cost estimate of a customer. Of the three different cost structures for vehicles used in Liu and Shen [11], we consider cost structures A (largest vehicle costs) and C (smallest vehicle costs). The same 6 problem instances are used to test the cost approximation methods, resulting in a test bed of 1200 customers for which incremental costs have to be estimated.

Figure 1: Long-run cost estimates



Full re-optimization (1) and DA500 (2) give the highest estimates for the incremental cost of a customer. Average performance is clearly lower from the third approach (DA100) onwards. As for the estimation of short run incremental costs, Figure 1 seems to suggest that the variance is not equal for the different cost estimators. This is confirmed by the standard deviation and standard error statistics in Table 6 and the formal Levene test statistic (289.466, Sig 0.000) rejecting the null hypothesis that the group variances are equal.
Given that the groups are of equal size, ANOVA is used to reject the hypothesis that average incremental cost estimates are equal across heuristics used (F = 227.578, Sig. 0.000). Using the post

hoc Tamhane T2 test (Table 2) the following differences between group means for the heuristics were found to be significant at the 5% level.

Table 1: Multiple comparisons for long run costs (A and C) based on Tamhane T2

	1	2	3	4	5	6	7	8
1		-0.7209535	-25.981485	-44.695939	-75.398092	-56.347825	-48.861196	-110.20222
2	0.7209535		-25.260531	-43.974986	-74.677138	-55.626871	-48.140243	-109.48126
3	25.981485	25.260531		-18.714455	-49.416607	-30.36634	-22.879711	-84.220731
4	44.695939	43.974986	18.714455		-30.702152	-11.651885	-4.1652568	-65.506277
5	75.398092	74.677138	49.416607	30.702152		19.050267	26.536896	-34.804124
6	56.347825	55.626871	30.36634	11.651885	-19.050267		7.4866283	-53.854392
7	48.861196	48.140243	22.879711	4.1652568	-26.536896	-7.4866283		-61.34102
8	110.20222	109.48126	84.220731	65.506277	34.804124	53.854392	61.34102	

Table 1 indicates that the full optimization (1) and DA500 (2) cost approximation have a similar performance. The group means of DA100 (3), single route optimization (5) and simple removal 8) are statistically different from all other cost estimators. The difference between local minimum (4) and close route optimization II (7) was not considered to be significant. The same goes for the difference between close route optimization I (6) and II (7), the difference between local minimum (4) and close route optimization I (6) being different at the 5% significance level.

For large datasets, statistical tests often have the drawback that null hypotheses are too easily rejected and due to the large number of observations practically non-significant variables are declared as statistically significant. This and the fact that the ability to save on vehicle costs matters most when vehicle costs are significant as they are in practice, we also report results on the A cost structure only. For the customers in the six problem instances for cost structure A, the null hypothesis of equal variances was rejected (Levene statistic 322.541, Sig 0.000). Because the equality of means across groups was rejected (F = 160.765, Sig 0.000), Table 2 list the differences between group means for the heuristics identified by the Tamhane T2 test at the 5% level. Contrary to the previous case, the difference between the performance of close re-optimization I and II is no longer considered to be significant.

Table 2: Multiple comparisons for long run costs (cost structure A) based on Tamhane T2

	1	2	3	4	5	6	7	8
1		-9.781975	-44.267551	-73.090543	-125.78218	-91.385472	-78.344565	-160.96156
2	9.781975		-34.485576	-63.308568	-116.00021	-81.603497	-68.56259	-151.17958
3	44.267551	34.485576		-28.822993	-81.51463	-47.117921	-34.077015	-116.69401
4	73.090543	63.308568	28.822993		-52.691637	-18.294928	-5.2540218	-87.871015
5	125.78218	116.00021	81.51463	52.691637		34.396709	47.437615	-35.179378
6	91.385472	81.603497	47.117921	18.294928	-34.396709		13.040906	-69.576086
7	78.344565	68.56259	34.077015	5.2540218	-47.437615	-13.040906		-82.616993
8	160.96156	151.17958	116.69401	87.871015	35.179378	69.576086	82.616993	

3.2. Cost drivers and approximating cost estimators

Table 3 illustrates that DA500 provides the best cost estimates if one ignores full re-optimization as a viable approach to determining incremental costs.

Table 3: Hit rates and computing times for short and long costs

	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

Both for the short and long run costs (cost structure A), DA500 generates the best possible cost estimates in more than 99% of the cases and within acceptable computation times (99.17% and 99.83% respectively).

Although high quality cost estimators have significant business value, logistics practitioners will also be interested in unraveling the underlying cost structure. By identify the main cost drivers of servicing customers, the cost of servicing a (new) customer could be easily estimated based on the customers characteristics without having to calculate any specific cost estimate. Moreover, information on cost drivers could be used to design price tariffs.

Eight different cost drivers were therefore considered for explaining the variation in the short-run cost estimates of DA500: the size of customer demand, the distance from the depot, the width of the service time window and the number of customers located within 5, 10, 20, 30 and 50% of the maximum distance in the problem. Stepwise linear regression (entry probability 5%, removal probability 10%) was used to explain the variation in the DA500 cost estimates. The models were unable to explain more than 50% of the variation in the DA500 incremental costs estimates and its parameter estimates sometimes had wrong signs. Explaining or predicting incremental costs based on cost drivers therefore seems to be of little value, although this conclusion is based on theoretical problem instances only. It may well be that this approach proves useful in real-life routing problems.

Table 3 illustrates that computation times are significantly higher for the first two incremental cost estimators (CPU times in seconds for all customers in the problem set). It would therefore be beneficial if one could obtain equally good cost estimates by identifying a strong relationship between the cost estimates obtained by the most powerful and time consuming cost approximation methods and the faster ones. Using stepwise regression (entry probability 5%, removal probability 10%) the following models were obtained for the dependent variable DA500 for estimating short cost run costs (see Tables 4 and 5).

Table 4: Model coefficients

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

Table 5: Model summary

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

Although explaining the variation in the cost estimates obtained by DA500 by means of a constant and the cost estimates of Single Route Optimization and Close Re-optimization I (model 2), obtained a higher adjusted R square, it cannot be preferred over Model 1. Model 2 suffers from a high degree of collinearity (Tolerance level < 0.10) due to the fact that the two explaining variables are highly correlated. The regression equation DA500 = 18.00 + 1.330 x cost estimate of Single Route Optimization (5) contains highly significant (from zero) coefficients and is capable of explaining 83.10% of the variation in the cost estimates of DA500, which can make it an interesting tool for on average obtaining higher quality cost estimates within a fraction of the computation cost (0.01418 versus 92.95). This difference is even larger when modeling the long run costs. For the DA500 long run cost estimates for cost structure A and the Single Route Optimization cost estimator the following model was obtained : DA500 = 31.705 + 3.315 x Single Route Optimization. Both explanatory variables are significantly different from zero (t-values of 9.002 and 50.274 respectively) and the model has an adjusted R square of 0.808.

4. Conclusion

This paper offers eight different approximations for the incremental cost of a customer. Full and partial re-optimization approaches are compared to naïve re-optimization procedures that simply remove a customer from the route without changing the sequence of the other customers in the route. The approaches clearly differ with respect to solution quality and CPU time requirements. Both for short and long run costs, DA500, an approximation based on the FSMVRPTW heuristic from Bräysy et al. [7] was considered to be the most powerful approach next to a time consuming full re-optimization approach.
Computation testing on theoretical benchmarks indicated that even a wide set of possible cost drivers cannot sufficiently model the incremental cost of a customer. Less than 50% of the variation in incremental cost estimates could be explained by cost drivers related to time window size, customer proximity and demand size. We therefore believe that approximating incremental costs either by DA500 or by a regression equation based on even faster cost estimators (e.g., Single Route Optimization) is a more fruitful approach to estimate incremental costs in real-life routing problems.

Acknowledgement

This work was partially supported by the EDGE project of the Research Council of Norway, the KAKS Foundation and the Research Foundation - Flanders (Belgium). This support is gratefully acknowledged. The authors would also like to thank Bruno De Borger, Peter Goos and Piet Rietveld for their constructive comments on an earlier version of this paper. The usual disclaimer applies.

References

[1] O. Bräysy (2003), A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows, *INFORMS Journal on Computing* **15**(4), 347 – 368.

[2] O. Bräysy, G. Hasle and W. Dullaert (2004), A multi-start local search algorithm for the vehicle routing problem with time windows, *European Journal of Operational Research* **159**, 586 – 605.

[3] O. Bräysy and M. Gendreau (2005), Vehicle routing problem with time windows, Part I: route construction and local search algorithms, *Transportation Science* **39**, 104 – 118.

[4] O. Bräysy and M. Gendreau (2005), Vehicle routing problem with time windows, Part II: metaheuristics, *Transportation Science* **39**, 119 – 139.

[5] O. Bräysy, M. Gendreau, G. Hasle and A. Løkketangen (2006a), A survey of heuristics for the vehicle routing problem, part I: basic problems and supply side extensions, Working paper, SINTEF ICT, Norway.

[6] O. Bräysy, M. Gendreau, G. Hasle and A. Løkketangen (2006b), A survey of heuristics for the vehicle routing problem, Part II: Demand Side Extensions, Working paper, SINTEF ICT, Norway.

[7] O. Bräysy, W. Dullaert, G. Hasle, D. Mester and M. Gendreau (2007), An Effective Multi-restart Deterministic Annealing Metaheuristic for the Fleet Size and Mix Vehicle Routing Problem with Time Windows, Paper submitted to Transportation Science.

[8] J. F. Cordeau, G. Laporte and A. Mercier (2001), A unified tabu search heuristic for vehicle routing problems with time windows, *Journal of the Operational Research Society*, **52**, 928 – 936.

[9] W. Dullaert, O. Bräysy, and G. Van de Weyer (2005), Scheduling flexibility in vehicle routing. In: Witlox, F., Dullaert, W. and B. Vernimmen (Eds.). *Proceedings of the BIVEC-GIBET Research Day*, Part I, 339 – 350.

[10] Y. H. Lee, Kim J. I., Kang K. H. and Kim K. H. (2006), A heuristic for vehicle fleet mix problem using tabu search and set partitioning, Working paper, Yonsei University, Seoul, Korea.

[11] F. H. Liu and S. Y. Shen (1999), The fleet size and mix vehicle routing problem with time windows, *Journal of the Operational Research Society* **50**, 721-732.

[12] I. Or (1976), Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. Ph.D. thesis, Northwestern University, Evanston, IL.

[12] I. H. Osman and Salhi S. (1996), Local search strategies for the vehicle fleet mix problem. V.J. Rayward-Smith, I.H. Osman, C.R. Reeves, G.D. Smith, eds. *Modern heuristic search methods*. Chichester, John Wiley & Sons, 131 – 153.

[13] S. Salhi and G. Rand (1993), Incorporating vehicle routing into the vehicle fleet composition problem, *European Journal of Operational Research* **66**, 313 – 360.

[14] M. M. Solomon (1987), Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, **35** 254 – 265.

[15] E. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin (1997), A tabu search heuristic for the vehicle routing problem with soft time windows, *Transportation Science* **31**, 170 – 186.

[16] P. Toth and D. Vigo (Eds) (2001), *The Vehicle Routing Problem.* SIAM monographs on discrete mathematics and applications. Society for Industrial and Applied Mathematics, Philadelphia, USA.