On solving the time window assignment vehicle routing problem via iterated local search

Lucas Burahem Martins
Advisor: Prof. DSc. Mayron César de Oliveira Moreira
Co-advisor: Prof. DSc. Manuel Iori
Giorgio Zucchi

14/08/2020

UNIVERSIDADE FEDERAL DE LAVRAS

Contents

Introduction

- Vehicle routing is a class of problems that appears in several combinatorial optimization studies due to their practical relevance.
  - Mainly in the areas of retail and transport [Toth e Vigo 2014].
- [Spliet e Desaulniers 2015] introduced the *Time Window Assignment Vehicle Routing Problem* (TWAVRP).
  - The exogenous time windows are represented by the arrival and departure limits of a customer.
  - Each endogenous time window with a fixed-width, must be associated with the exogenous time window of the client.

Introduction

- The TWAVRP faced in this work is part of a research whose focus is to give an efficient and accurate solution for a routing problem faced by an Italian company (Coopservice) providing logistics services in several distribution fields.

Introduction

- Our purpose is to help the company to minimize the actual delivery time and the total cost of the routing service.
- We decided to start our research by first looking at the combinatorial aspect of the TWAVRP, with the aim of focusing later on its application to the company case study.

Introduction

- Coopservice context:
  - Each vehicle leaves a depot and must visit a set of hospitals.
  - The hospital staff should be at the delivery place when the vehicle arrives.
  - Each hospital has a particular time window.
  - Each hospital requests products that respect the vehicle capacity.
- Over a set of scenarios Ω, the challenge is to build a schedule subject to the technical constraints, minimizing costs and maximizing time window robustness.

Introduction

- We propose an algorithm that:
  - Generates a set of routes by invoking an *Iterated Local Search* (ILS) metaheuristic.
  - Selects the most appropriate routes through an auxiliary mathematical formulation.

General Objective

Is there a heuristic strategy that can efficiently solve the TWAVRP as defined by [Dalmeijer e Spliet 2018, Spliet e Gabor 2014]?

Literature review

- The approached problem has characteristics that resemble:
  1. Pharmaceutical Vehicle Routing Problem (Pharmaceutical VRP) [Magalhães e Sousa 2006];
  2. Vehicle Routing Problem with Time Windows (VRPTW) [Desrochers, Desrosiers e Solomon 1992];
  3. Time Window Assignment Vehicle Routing Problem (TWAVRP) [Spliet e Gabor 2014];

Methodology: Proposed heuristic

- The proposed heuristic has two successive phases.
  1. Generate a pool of feasible routes;
  2. Selects a subset of routes having minimum cost.

Algorithm 1: Main algorithm

Input: I (instance)
Output: $(s, f(s))$ (solution, and its objective function)
$P \leftarrow \emptyset$; ▹ Empty pool of routes
foreach $\omega \in \Omega$ **do**
    $P \leftarrow P \cup ILS(I_\omega, \alpha, n_{iter})$; ▹ Generating the set of routes for each scenario
$s \leftarrow RSM(P, I)$; ▹ Route Selector Model (RSM)
return $(s, f(s))$;

Methodology: Iterated Local Search (ILS)

- ILS algorithm is a metaheuristic method to generate a sequence of solutions to a problem iteratively. These solutions are obtained through iterative applications of improvement methods in each solution [Stützle e Ruiz 2018].
  - Initial Solution
  - Local Search (LS)
  - Perturbation
  - Acceptance criterion

Methodology: Constructive Heuristic

Algorithm 2: Constructive Heuristic (CH)

Input: I (data set), H_ω (set of all available clients for a data set I on scenario ω)
Output: s (feasible solution)
$s \leftarrow \emptyset$;
$\tilde{\mathscr{H}} \leftarrow sort(H_\omega)$; ▷ sort clients in non-descending order of earliest exogenous time window
while $\tilde{\mathscr{H}} \neq \emptyset$ **do**
  $\mathscr{R} \leftarrow \emptyset$;
  foreach $j \in \tilde{\mathscr{H}}$ **do**
    $\mathscr{R} \leftarrow \mathscr{R} \cup \{j\}$;
    if $infeasible(\mathscr{R}) = $ ***true*** **then**
      $\mathscr{R} \leftarrow \mathscr{R} \setminus \{j\}$;
    else
      $\tilde{\mathscr{H}} \leftarrow \tilde{\mathscr{H}} \setminus \{j\}$;
  $s \leftarrow s \cup \mathscr{R}$;
return s

Methodology: Local Search (ILS)

- The proposed LS method will be composed of 6 elementary neighborhood movements:
  - N1 *Relocate intra-route*
  - N2 *Swap intra-route*
  - N3 *2-opt*
  - N4 *Relocate inter-route*
  - N5 *Swap inter-route*
  - N6 *Cross inter-route*

Methodology: Local Search (ILS)

Algorithm 3: Local Search method (LS)

Input: s (feasible solution)
Output: s^* (best feasible solution found)
$s^* \leftarrow s$;
foreach $N \in NL(s^*)$ **do** ▹ *$NL(s^*)$: list of inter-neighborhoods of solution s^**
  foreach $s' \in N$ **do**
    if $f(s') < f(s^*)$ ***and*** *feasible*(s') = ***true*** **then**
      $s^* \leftarrow s'$;
      foreach $N \in NI(s^*)$ **do** ▹ *$NI(s^*)$: list of intra-neighborhoods of solution s'*
        foreach $\tilde{s} \in N$ **do**
          if $f(\tilde{s}) < f(s^*)$ ***and*** *feasible*$(\tilde{s})$ = ***true*** **then**
            $s^* \leftarrow \tilde{s}$;
return s^*

Methodology: Perturbation

- Starting from a solution s^*, the Perturbation method invokes a list of $NL(s^*)$ of possible neighborhood moves according to all neighborhood moves (N1, N2, N3, N4, N5, and N6). A percentage α of neighborhoods in $NL(s^*)$ is randomly chosen and applied to s^*.

Methodology: Iterated Local Search (ILS)

Algorithm 4: Iterated Local Search (ILS)

Input: H_ω (data set), α (perturbation factor), n_{iter} (number of iterations)
Output: $\mathscr{P}$ (set of feasible solutions found)
$s^* \leftarrow \emptyset$; ▹ Best solution found so far (take $f(s^*) = +\infty$)
$s \leftarrow CH(H, H_\omega)$; ▹ H_ω: set of available customers of data set H
$s_{ls} \leftarrow LS(s)$;
$\mathscr{P} \leftarrow s_{ls} \cup s$; ▹ Initializing the set of feasible solutions
$s^* \leftarrow s_{ls}$;
$count \leftarrow 0$
while $count \neq n_{iter}$ **do**
    $s' \leftarrow Perturbation(s^*, \alpha)$;
    $s_{ls} \leftarrow LS(s')$;
    $\mathscr{P} \leftarrow \mathscr{P} \cup s' \cup s_{ls}$;
    if $f(s') < f(s^*)$ **then**
        $s^* \leftarrow s'$;
        $count \leftarrow 0$;
    else
        $count \leftarrow count + 1$;
return $\mathscr{P}$;

Methodology: Route Selector Model (RSM)

- The ILS algorithm generates a set R_ω of feasible routes for each scenario $\omega \in \Omega$.
- Note that all routes in R_ω respect for the TWAVRP:
  - Capacity of the vehicles;
  - Time-windows of the customers;
- The MILP aim is to choose the most appropriate subset of routes from R_ω, assigning an endogenous time window to each client, overall scenarios.

Methodology: Route Selector Model (RSM)

- Take from R_ω:
  1. f_{jr}^{ω} as the starting time of service on client j on the route r in scenario ω;
  2. c_r^{ω} as the cost to choose a route $r \in R_\omega$ in scenario ω;
  3. x_{jr}^{ω} as a binary parameter equal to one if client j belongs to route $r \in R_\omega$ in scenario ω, 0 otherwise.
- Customer $j \in H$ must be delivered at time window $[e_j, l_j]$.
- Consider u_r^{ω} as a binary variable equal to one if route $r \in R_\omega$ is selected, 0 otherwise.
- y_i as a continuous variable that measures the starting time of the endogenous time window of customer $i \in H$.
- w_i gives the time window width of customer i.

Methodology: Route Selector Model (RSM)

$$\min \sum_{\omega \in \Omega} p_\omega c_r^\omega u_r^\omega \quad (1)$$

subject to

$$\sum_{r \in R_\omega} x_{jr}^\omega u_r^\omega = 1 \qquad \forall j \in H, \omega \in \Omega \quad (2)$$

$$\sum_{r \in R_\omega} f_{jr}^\omega x_{jr}^\omega u_r^\omega \geq y_j \qquad \forall j \in H, \omega \in \Omega \quad (3)$$

$$\sum_{r \in R_\omega} f_{jr}^\omega x_{jr}^\omega u_r^\omega \leq y_j + w_i \qquad \forall j \in H, \omega \in \Omega \quad (4)$$

$$y_j \in [e_j, l_j - w_j] \qquad \forall j \in H, \omega \in \Omega \quad (5)$$

$$u_r^\omega \in \{0, 1\} \qquad \forall \omega \in \Omega, r \in R_\omega. \quad (6)$$

Computational experiments: TWAVRP Instances

- Each instance considers a different combination of:
  - Number of customers
  - Vehicle capacity
  - Demand for each scenario
  - Probability of each scenario
  - Exogenous time windows
  - Size of endogenous time windows
  - Travel costs
  - Travel times
- The instance set comprises ninety instances divided into two classes: small instances and large ones.

Computational experiments: Experiments

- The experiments compare our ILS based-algorithm with the *Branch-and-Cut* (B&C) proposed by [Dalmeijer e Spliet 2018]
- Algorithm 4 was executed five times on each instance.
    1. This number was tuned through preliminary tests in which we obtained a good trade-off between quality and computational effort.
- n_{iter} and α were tuned by *Irace* package [López-Ibáñez et al. 2016].
- we generated 200 training instances by using the instance generator proposed by [Dalmeijer e Spliet 2018].
- The values returned by the *Irace* package at the end of this test were $n_{iter} = 100$ and $\alpha = 0.35$.

Computational experiments: Experiment 1

Table 1: Average results aggregated by number of customers (10 instances per line, 5 ILS executions per instance)

Instance	CPU time (seconds)			Gaps	
N. customers	B&C	ILS	ILS+RSM	Gap* (%)	Gap(%)
10	0.1	4.50 ± 0.29	6.61 ± 0.56	0.34 ± 1.00	0.41 ± 1.02
15	4.5	16.50 ± 1.17	26.25 ± 1.86	0.00 ± 0.18	0.11 ± 0.25
20	2.2	39.06 ± 2.01	80.30 ± 7.49	0.02 ± 0.05	0.06 ± 0.10
25	12.4	68.48 ± 2.03	153.29 ± 18.56	0.06 ± 0.14	0.27 ± 0.78
30	544.0	107.27 ± 3.40	284.38 ± 12.62	0.04 ± 0.10	0.28 ± 0.39
35	1,531.7	161.59 ± 9.48	501.77 ± 97.94	0.02 ± 0.13	0.29 ± 0.42
40	3,252.0	224.33 ± 6.11	749.92 ± 41.11	0.10 ± 0.52	0.72 ± 0.73
45	3,600.0	289.34 ± 28.78	990.15 ± 172.79	-0.69 ± 0.83	-0.18 ± 1.61
50	3,600.0	372.98 ± 24.41	1,743.16 ± 261.71	-1.89 ± 0.12	-1.62 ± 1.31

Computational experiments: Experiment 1

Table 2: Results for instances with 45-50 customers (best UB values appear in bold)

Instance		B&C by [Dalmeijer e Spliet 2018]		ILS + RSM	
#	N. customers	LB	UB	Best UB	Avg UB
71	45	49.52	51.78	**51.22**	**51.41**
72	45	50.73	**52.13**	51.86	52.94
73	45	41.50	**41.70**	41.95	42.24
74	45	47.25	**47.84**	47.96	48.16
75	45	48.77	49.86	**49.47**	50.02
76	45	48.38	52.09	**49.90**	**50.03**
77	45	50.09	**51.18**	**51.18**	51.25
78	45	52.02	53.95	**53.35**	**53.74**
79	45	47.45	**48.21**	48.27	48.69
80	45	49.57	**50.57**	50.61	50.78
81	50	56.81	58.85	**58.16**	**58.29**
82	50	51.50	53.20	**52.98**	**53.03**
83	50	57.45	60.67	**58.77**	**58.89**
84	50	52.31	56.38	**54.09**	**54.23**
85	50	53.74	56.07	**55.06**	**55.26**
86	50	51.68	54.76	**53.02**	**53.16**
87	50	52.47	54.14	**53.81**	**53.87**
88	50	54.82	56.91	**56.27**	**56.36**
89	50	59.23	61.51	**60.32**	**60.62**
90	50	57.68	59.55	**58.95**	**59.23**

Conclusion

- We studied the *Time Windows Assignment Vehicle Routing Problem* (TWAVRP).
- We compared the results of our algorithm (ILS+RSM) with the Branch-and-Cut proposed by [Dalmeijer e Spliet 2018].
- The ILS+RSM presented competitive results, concerning both solution quality and computational effort, in particular for the larger size instances involving 45 and 50 customers.

Conclusion

- Future avenues concern:
  1. incorporating new complicating constraints deriving from the real-world case study in the metaheuristic;
  2. testing other neighborhood-based metaheuristics as generators of routes;
  3. testing multiple calls to the RSM with different pools of routes.

References

- CAMPELO, P. et al. Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector. European Journal of Operational Research, v. 273, n. 1, p. 131–145, 2019.

DALMEIJER, K.; SPLIET, R. A branch-and-cut algorithm for the time window assignment vehicle routing problem.Computers Operational Research, v. 89, p. 140–152, 2018.

DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. Operations Research, v. 40, n. 2, p. 342–354, 1992.

SPLIET, R.; GABOR, A. F. The time window assignment vehicle routing problem. Transportation Science, v. 49, n. 4, p. 721–731, 2015.

TA S, D.; JABALI, O.; Van Woensel, T. A Vehicle Routing Problem with Flexible Time Windows. Computers Operations Research , v. 52, p. 39–54, 2014.