LNS based Matheuristic for the Dial-a-Ride Problem

Oscar Tellez[1], Samuel Vercraene[1], Fabien Lehuédé[2], Olivier Péton[2],Thibaud Monteiro[1].

[1]University of Lyon, INSA Lyon, DISP Laboratory EA 4570

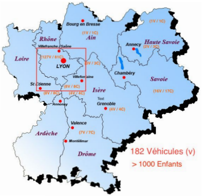
{oscar.tellez, samuel.vercraene,thibaud.monteiro}@insa-lyon.fr

[2]École des Mines Nantes, IRCCyN Laboratory

{fabien.lehuede, olivier.peton}@mines-nantes.fr

November 3, 2016

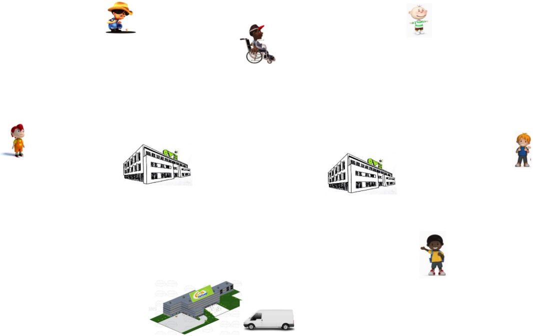
Daily transportation service for disabled people by *GIHP Service Adapté*



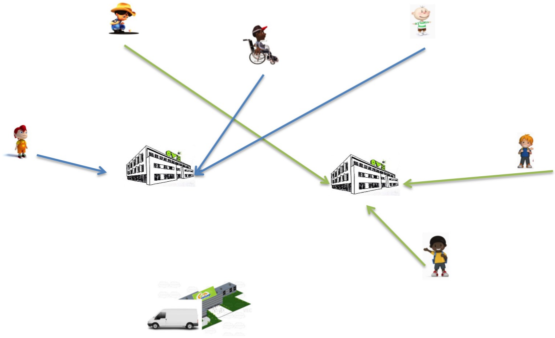
Rhône Alpes

Medical-Social Establishments	Families Children	Transporter vehicles
x160	**x1000**	**X182**

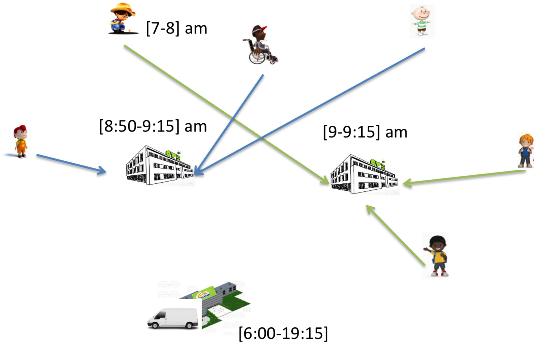
Dial-a-Ride Problem (DARP)



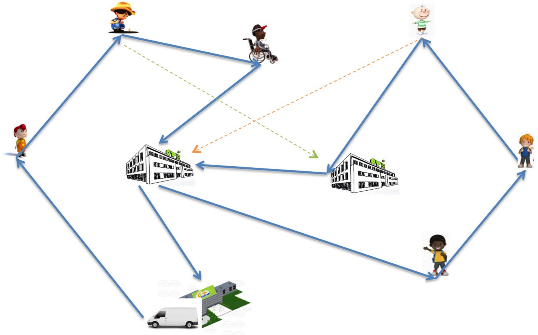
Dial-a-Ride Problem (DARP)



Dial-a-Ride Problem (DARP)



Dial-a-Ride Problem (DARP)



Solution Methods for DARP

Auteur	Exact	Aprox	Instances	Solution methods
Gschwind and Irnich [2014]	X		C	Branch & Cut & Price
Ropke and Cordeau [2009]	X		C/R	Branch & Cut & Price
Braekers et al. [2014]	X		C	DA
Cordeau and Laporte [2003]		X	C	Tabu Search
Ropke and Pisinger [2006]		X	C/R	ALNS
Parragh et al. [2010]		X	C	VNS
Jain and Van Hentenryck [2011]		X	C	LNS-FFPA
Qu and Bard [2012]		X	PDP	GRASP
Masson et al. [2014]		X	C	ALNS
Braekers et al. [2014]		X	C	DA
Chassaing et al. [2016]		X	C/R	ELS
Parragh and Schmid [2013]	X	X	C	Hybrid Column Generation and LNS
Ritzinger et al. [2014]	X	X	C	Dynamic Programming LNS
Grangier et al. [2016]	X	X	VRP-CD	LNS-SPP

Instances: (C) Cordeau (R) Ropke & Pisinger.

Algorithm 1: Matheuristic LNS-SCP

Input: Problem data, Ω^- *set of destroy operators*, Ω^+ *set of repair operators*
Output: Best found solution s^*

s : *Current solution*, s' : *Temporary solution*
Pool of routes $\mathcal{J} \leftarrow \emptyset$
Generate initial solution s
while *Stopping criterion not reached* **do**
  $s' \leftarrow s$
  Destroy quantity: select a number Φ of requests to remove from s'
  Operator selection: destroy operator $M^- \in \Omega^-$, repair operator $M^+ \in \Omega^+$
  Destroy: Apply M^- to remove Φ requests of s'
  Repair: Apply M^+ to reinsert requests
  $\mathcal{J} \leftarrow \mathcal{J} \cup routes(s')$
  if *acceptance criterion is met* **then**
    $s \leftarrow s'$
  if *cost of* s' *is better than cost of* $s^\star$ **then**
    $s^* \leftarrow s'$
  if *SCP criterion is reached* **then**
    $s' \leftarrow$ Solve Set Covering Problem($\mathcal{J}$)
    update $s^* \leftarrow s'$ if s' is cheaper
    Perform pool management
return s^*

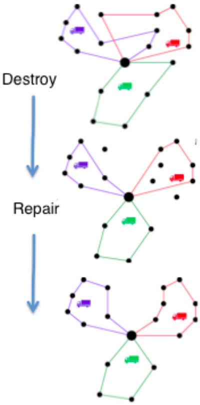


Figure: LNS

LNS Operators

The ALNS framework from Pisinger and Ropke [2007] suggests the following operators:

Destroy Operators

- Random Removal
- Historical Node-Pair Removal
- Historical Request-Pair Removal
- Worse Removal
- Route Removal (cluster)
- Distance Removal
- Time Removal
- Proximity Removal

Repair Operators

- Cheapest Insertion
- K-regret

LNS Operators

When combined with SCP for the DARP, experiments suggest to use:

Destroy Operators

- Random Removal
- Historical Node-Pair Removal
- Historical Request-Pair Removal
- Worse Removal
- Route Removal (cluster)
- Distance Removal
- Time Removal
- Proximity Removal

Repair Operators

- Cheapest Insertion
- K-regret

Set Covering Problem (SCP)

$\mathcal{J} = \{1, \ldots, |\mathcal{J}|\}$ be the set of routes in the pool,

$v_t \in \mathbb{R}^+$ be the cost of route $t \in \mathcal{J}$,

$R_{rt} \in \{0, 1\}$ be 1 if request $r \in \mathcal{R}$ is served by route $t \in \mathcal{J}$ and 0 otherwise,

$y_t = 1$ if route $t \in \mathcal{J}$ is used in the solution and 0 otherwise.

$$Objective := \min \sum_{t \in \mathcal{J}} v_t y_t \tag{1}$$

s.t

$$\sum_{t \in \mathcal{J}} R_{rt} y_t \geq 1 \qquad \forall r \in \mathcal{R}, \tag{2}$$

$$y_t \in \{0, 1\}. \tag{3}$$

Set Covering Problem (SCP)

Memory mechanism: Non-dominated routes are memorized at every LNS iteration.

Definition (Route dominance: route $\mathscr{A}$ dominates route $\mathscr{B}$)

if $\mathscr{A}$ visits the same set of nodes as $\mathscr{A}$ at a lower cost.

SCP Solving:

- The SCP is solved every η iterations
- MILP solver is used (time limit 3 secs, initialized with s^*)
- Current solution is updated only if SCP improves it.
- At the end, duplicates request are removed from SCP solution.

Pool management: If SCP solution was not optimal within the time limit, pool is cleared and filled again with the routes of s^* .

Comparison Performance: LNS-SCP vs ALNS vs LNS

		LNS-SCP		ALNS		LNS	
Instance	BKS	$\%GAP_{min}$	$\%GAP_{Avg}$	$\%GAP_{min}$	$\%GAP_{Avg}$	$\%GAP_{min}$	$\%GAP_{Avg}$
R1a	190.02	0.00	0.00	0.00	0.00	0.00	0.00
R2a	301.34	0.00	0.00	0.00	0.00	0.00	0.26
R3a	532	0.00	0.13	0.32	0.57	0.59	0.85
R4a	570.25	0.61	1.13	3.12	3.48	3.50	3.82
R5a	626.93	7.42	8.28	6.15	7.87	7.88	8.23
R6a	785.26	5.27	6.75	6.04	6.65	5.98	7.05
R7a	291.71	0.00	0.00	0.00	0.00	0.00	0.00
R8a	487.84	1.44	1.87	1.36	2.52	2.52	3.24
R9a	658.31	4.49	5.66	4.40	6.14	5.05	7.12
R10a	851.82	8.92	9.48	9.28	10.05	8.95	10.03
R1b	164.46	0.00	0.00	0.00	0.00	0.00	0.00
R2b	295.66	0.10	0.19	0.43	0.50	0.80	1.13
R3b	484.83	0.87	1.21	2.14	2.57	2.59	3.07
R4b	529.33	0.58	2.60	4.39	5.59	6.29	6.87
R5b	577.29	4.25	4.90	5.26	5.78	5.32	6.05
R6b	730.67	7.46	8.95	7.83	8.82	8.08	8.56
R7b	248.21	0.00	0.00	0.00	0.00	0.00	0.00
R8b	**458.73**	**0.66**	**1.52**	**2.51**	**3.05**	**3.01**	**3.49**
R9b	593.49	1.29	3.82	6.68	7.24	5.52	7.02
R10b	785.68	9.04	11.48	11.10	12.03	10.37	11.92
Average		2.62	3.40	3.55	4.14	3.82	4.44

Instances from Cordeau and Laporte [2003]. 5 mins evaluation over 5 runs.
LNS-SCP gains an small performance.

Comparison efficiency: LNS-SCP vs ALNS vs LNS

LNS-SCP gains a lot of efficiency!

	LNS-SCP		ALNS		LNS	
Instance*	t(sec)	t%	t(sec)	t%	t(sec)	t%
R1a	1.98	26%	7.69	100%	6.91	90%
R2a	51.84	29%	102.62	58%	177.19	100%
R3a	236.59	100%	194.49	82%	123.26	52%
R7a	16.75	63%	26.08	98%	26.52	100%
R1b	5.95	41%	14.45	100%	3.52	24%
R2b	107.80	49%	192.66	88%	218.60	100%
R7b	8.55	16%	31.58	59%	53.09	100%
Average		46%		84%		81%

t: Average time to find the BKS. Test in 5 mins over 5 runs.
* Instances where all three algorithms found a BKS.

Benchmarking: 5 mins test

LNS-SCP is competitive with dedicated algorithms!

		Jain and Van Hentenryck [2011]		Parragh and Schmid [2013]		Masson et al. [2014]		Ours	
		LNS-FFPA (5 runs)		HLNS (5 runs)		ALNS (5 runs)		LNS-SCP (5 runs)	
Instance	New BKS	Avg	% GAP_{Avg}	Avg	$\%GAP_{Avg}$	Avg	$\%GAP_{Avg}$	Avg	$\%GAP_{Avg}$
R1a	190.02	190.77	0.40	190.02	0.00	190.02	0.00	190.02	0.00
R2a	301.34	304.45	1.03	303.63	0.76	301.98	0.21	301.34	0.00
R3a	532.00	547.15	2.85	536.77	0.90	536.74	0.89	532.70	0.13
R4a	570.25	595.05	4.35	581.41	1.96	588.37	3.18	576.72	1.13
R5a	626.93	662.56	5.68	643.72	2.68	655.02	4.48	678.83	8.28
R6a	785.26	832.74	6.05	837.01	6.59	824.21	4.96	838.29	6.75
R7a	291.71	292.86	0.39	291.93	0.08	291.71	0.00	291.71	0.00
R8a	487.84	505.15	3.55	498.44	2.17	500.30	2.55	496.96	1.87
R9a	658.31	711.60	8.09	697.22	5.91	691.45	5.03	695.57	5.66
R10a	851.82	911.18	6.97	908.12	6.61	903.35	6.05	932.54	9.48
R1b	164.46	164.46	0.00	164.46	0.00	164.46	0.00	164.46	0.00
R2b	295.66	301.67	2.03	297.99	0.79	300.58	1.66	296.24	0.19
R3b	484.83	504.69	4.10	495.02	2.10	499.26	2.98	490.68	1.21
R4b	529.33	566.48	7.02	540.76	2.16	554.96	4.84	543.08	2.60
R5b	575.64	610.33	6.03	607.17	5.48	604.28	4.98	605.55	5.20
R6b	730.67	785.13	7.45	772.21	5.69	780.71	6.85	796.08	8.95
R7b	248.21	248.31	0.04	248.21	0.00	248.21	0.00	248.21	0.00
R8b	458.73	477.75	4.15	473.19	3.15	472.35	2.97	465.71	1.52
R9b	593.49	633.51	6.74	620.39	4.53	617.22	4.00	616.16	3.82
R10b	784.30	857.95	9.39	874.36	11.48	844.48	7.67	875.88	11.68
Avg		535.19	4.32	529.10	3.15	528.48	3.17	531.84	3.42

Instances from Cordeau and Laporte [2003] in 5 mins over 5 runs.

Benchmarking: 50000 iterations

LNS-SCP is competitive with state of the art algorithms + 2 new BKS!

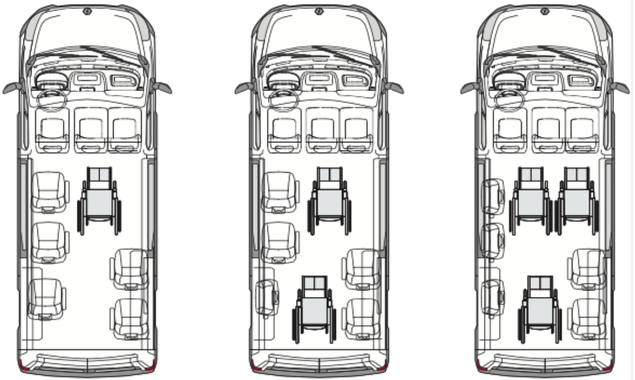
Paper		Parragh [2011]			Masson et al. [2014]			Braekers et al. [2014]			Chassaing et al. [2016]			Ours		
BKS_{new}		GAP_{Avg}	GAP_{min}	T_{min}	GAP_{Avg}	GAP_{min}	T_{min}	GAP_{Avg}	GAP_{min}	T_{min}	GAP_{Avg}	GAP_{min}	T_{min}	GAP_{Avg}	GAP_{min}	T_{min}
R1a	190.019	0.00	0.00	0.54	0.00	0.00	2.09	0.00	0.00	0.28	0.00	0.00	0.25	0.00	0.00	0.04
R2a	301.336	0.39	0.00	2.76	0.13	0.00	6.06	0.00	0.00	0.70	0.00	0.00	1.25	0.00	0.00	0.86
R3a	531.995	1.17	0.62	5.11	0.29	0.00	13.17	0.29	0.02	0.81	0.35	0.08	2.30	0.25	0.00	3.70
R4a	570.25	1.05	0.15	16.29	1.63	0.56	30.47	1.80	1.21	1.24	0.74	0.05	7.37	0.31	0.24	13.86
R5a	**626.93**	**1.70**	**0.41**	**26.70**	**1.47**	**0.71**	**44.00**	**0.82**	**0.46**	**1.49**	**1.70**	**0.62**	**12.07**	**6.70**	**6.34**	**26.63**
R6a	785.26	1.92	0.46	48.48	1.61	0.78	64.96	1.96	1.59	1.78	1.38	0.96	21.92	2.24	0.41	35.72
R7a	291.71	0.29	0.00	1.02	0.29	0.00	3.45	0.18	0.18	0.38	0.43	0.00	0.47	0.00	0.00	0.13
R8a	487.84	1.51	0.84	5.92	1.67	0.65	14.76	0.65	0.64	0.81	1.09	0.77	2.68	0.77	0.41	6.48
R9a	658.31	2.70	0.48	24.89	1.69	0.79	34.43	1.27	0.66	1.20	3.50	2.21	11.25	1.28	0.52	20.21
R10a	851.82	3.18	2.41	47.17	1.69	1.22	67.75	1.06	0.25	1.91	1.04	0.65	21.33	3.63	1.97	41.53
R1b	164.46	0.97	0.00	0.61	0.64	0.00	2.19	0.00	0.00	0.40	0.00	0.00	0.28	0.00	0.00	0.15
R2b	295.66	1.09	0.10	3.00	0.34	0.10	9.17	0.14	0.01	0.86	0.02	0.00	1.37	0.23	0.00	2.87
R3b	484.825	1.33	0.00	8.19	1.44	0.45	23.82	1.07	0.78	1.27	1.21	0.86	3.70	0.50	0.00	10.58
R4b	529.33	2.24	1.04	22.58	1.18	0.68	51.26	2.20	1.07	1.95	0.50	0.33	10.20	1.18	0.67	21.00
R5b	***575.637***	**2.24**	**1.80**	**44.09**	**1.33**	**0.19**	**91.54**	**1.22**	**0.72**	**2.59**	**0.51**	**0.20**	**19.93**	**0.97**	**0.00**	**36.59**
R6b	730.67	1.78	1.00	71.50	2.47	1.51	128.36	2.26	1.76	3.01	0.81	0.08	32.32	4.71	2.92	48.50
R7b	248.21	0.00	0.00	1.30	0.00	0.00	5.06	0.45	0.45	0.57	0.00	0.00	0.58	0.00	0.00	0.30
R8b	458.73	2.51	1.08	9.54	1.44	0.58	23.63	0.80	0.66	1.35	0.80	0.54	4.32	0.72	0.00	8.10
R9b	593.49	2.15	0.00	27.49	1.88	1.13	68.13	1.20	0.80	2.44	0.68	0.32	12.43	1.24	0.50	27.55
R10b	***784.303***	3.45	2.36	69.57	1.78	1.27	118.18	2.06	1.20	2.71	2.33	1.39	31.45	3.32	0.00	56.86
Avg		**1.58**	**0.64**	**21.84**	**1.15**	**0.53**	**40.12**	**0.97**	**0.62**	**1.39**	**0.85**	**0.45**	**9.87**	**1.40**	**0.70**	**18.08**
Nb. Best		**7**			**6**			**3**			**6**			**7**		

Benchmarking instances Cordeau and Laporte [2003] over 5 runs.

Contributions

- LNS-SCP framework for the DARP.
- Selection of LNS fundamental Operators.
- The SCP component increases the (A)LNS efficiency and performance.
- In the 5 mins benchmark provides competitive results for dedicated algorithms.
- In the 50000 iterations benchmark provides 2 new BKS and stands competitive against state of the art algorithms.

Perspectives: reconfigurable vehicles capacity



7 🚹 + 1 ♿ =8 | 6 🚹 + 2 ♿ =8 | 4 🚹 + 3 ♿ =7

Perspectives: regularity issues

- **Master Problem:** determine the set of daily solutions so that regularity is maximized.
- **Sub-problems:** Transportation plan for every mid-day.

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
Matin	M1	M2	M3	M4	M5
Soir	S1	S2	S3	S4	S5

LNS based Matheuristic for the Dial-a-Ride Problem

Oscar Tellez[1], Samuel Vercraene[1], Fabien Lehuédé[2], Olivier Péton[2],Thibaud Monteiro[1].

[1]University of Lyon, INSA Lyon, DISP Laboratory EA 4570

{oscar.tellez, samuel.vercraene,thibaud.monteiro}@insa-lyon.fr

[2]École des Mines Nantes, IRCCyN Laboratory

{fabien.lehuede, olivier.peton}@mines-nantes.fr

November 3, 2016

Timo Gschwind and Stefan Irnich. Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. *Transportation Science*, 49(2):335–354, 2014.

Stefan Ropke and Jean-François Cordeau. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3):267–286, 2009.

Kris Braekers, An Caris, and Gerrit K. Janssens. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166–186, September 2014. ISSN 01912615. doi: 10.1016/j.trb.2014.05.007. URL http://linkinghub.elsevier.com/retrieve/pii/S0191261514000800.

Jean-François Cordeau and Gilbert Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Trans. Res. Part B: Methodol.*, 37(6):579–594, July 2003.

Stefan Ropke and David Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4):455–472, November 2006.

ISSN 0041-1655, 1526-5447. doi: 10.1287/trsc.1050.0135. URL http://pubsonline.informs.org/doi/abs/10.1287/trsc.1050.0135.

Sophie N Parragh, Karl F Doerner, and Richard F Hartl. Variable neighborhood search for the dial-a-ride problem. *Comput. Oper. Res.*, 37(6):1129–1138, June 2010.

Siddhartha Jain and Pascal Van Hentenryck. Large neighborhood search for Dial-a-Ride problems. In *Principles and Practice of Constraint Programming – CP 2011*, Lecture Notes in Computer Science, pages 400–413. Springer Berlin Heidelberg, 2011.

Yuan Qu and Jonathan F Bard. A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439–2456, 2012.

Renaud Masson, Fabien Lehuédé, and Olivier Péton. The Dial-A-Ride problem with transfers. *Comput. Oper. Res.*, 41(0):12–23, January 2014.

M. Chassaing, C. Duhamel, and P. Lacomme. An ELS-based approach with dynamic probabilities management in local search for the Dial-A-Ride Problem. *Engineering Applications of Artificial Intelligence*, 48:119–133, February 2016. ISSN 09521976. doi:

10.1016/j.engappai.2015.10.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0952197615002171.

Sophie N Parragh and Verena Schmid. Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Comput. Oper. Res.*, 40(1):490–497, January 2013.

Ulrike Ritzinger, Jakob Puchinger, and Richard F Hartl. Dynamic programming based metaheuristics for the dial-a-ride problem. *Ann. Oper. Res.*, pages 1–18, 18 May 2014.

Philippe Grangier, Michel Gendreau, Fabien Lehuédé, and Louis-Martin Rousseau. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. Technical report, CIRRELT, 2016.

David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, August 2007. ISSN 03050548. doi: 10.1016/j.cor.2005.09.012. URL http://linkinghub.elsevier.com/retrieve/pii/S0305054805003023.

Sophie N Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transp. Res. Part C: Emerg. Technol.*, 19(5):912–930, August 2011.