The Time Window Assignment Vehicle Routing Problem

Remy Spliet, Adriana F. Gabor

Erasmus University Rotterdam

June 13, 2012

Problem Description

Consider a distribution network of one depot and multiple customers:

Problem Description

Consider a distribution network of one depot and multiple customers:

Stage 1: Demand is uncertain, assign time windows

Problem Description

Consider a distribution network of one depot and multiple customers:

Stage 1: Demand is uncertain, assign time windows

Stage 2: Demand is known, design routes adhering to the time windows.

Problem Definition

- n customers.
- Scenarios of demand Ω with each scenario $\omega \in \Omega$ occurring with probability p_ω.
- Demand d^ω for each $\omega \in \Omega$.
- Vehicle capacity Q.
- Travel costs c_{ij} and travel time t_{ij} for going from customer i to j.
- Exogenous time windows $[s, e]$.
- Endogenous time window widths w.

Problem Definition

- n customers.
- Scenarios of demand Ω with each scenario $\omega \in \Omega$ occurring with probability p_ω.
- Demand d^ω for each $\omega \in \Omega$.
- Vehicle capacity Q.
- Travel costs c_{ij} and travel time t_{ij} for going from customer i to j.
- Exogenous time windows $[s, e]$.
- Endogenous time window widths w.

Time Window Assignment Vehicle Routing Problem:
Assign endogenous time windows to each customer such that the expected costs of visiting them on the day of delivery are minimized.

TWAVRP model

- Let $R(\omega)$ be the set of feasible routes for scenario ω.

TWAVRP model

- Let $R(\omega)$ be the set of feasible routes for scenario ω.
- Let c_r be the costs of route r.

TWAVRP model

- Let $R(\omega)$ be the set of feasible routes for scenario ω.
- Let c_r be the costs of route r.
- Let a_r^v be 1 if location v is visited on route r.

TWAVRP model

- Let $R(\omega)$ be the set of feasible routes for scenario ω.
- Let c_r be the costs of route r.
- Let a_r^v be 1 if location v is visited on route r.
- Let t_r^v be the time of arrival at location v on route r, and 0 if location v is not visited on route r.

TWAVRP model

- Let $R(\omega)$ be the set of feasible routes for scenario ω.
- Let c_r be the costs of route r.
- Let a_r^v be 1 if location v is visited on route r.
- Let t_r^v be the time of arrival at location v on route r, and 0 if location v is not visited on route r.

Let $y_i \in [s_i, e_i - w_i]$ be the start of the time window for customer i.

TWAVRP model

- Let $R(\omega)$ be the set of feasible routes for scenario ω.
- Let c_r be the costs of route r.
- Let a_r^v be 1 if location v is visited on route r.
- Let t_r^v be the time of arrival at location v on route r, and 0 if location v is not visited on route r.

Let $y_i \in [s_i, e_i - w_i]$ be the start of the time window for customer i.
Let x_r^ω indicate whether route r is used for scenario ω.

TWAVRP model

$$\min \sum_{\omega \in \Omega} p_\omega \sum_{r \in R(\omega)} c_r x_r^\omega$$

$$\sum_{r \in R(\omega)} a_r^v x_r^\omega = 1 \qquad v = 1, \ldots, n, \forall \omega \in \Omega$$

$$\sum_{r \in R(\omega)} t_r^v x_r^\omega \geq y_v \qquad v = 1, \ldots, n, \forall \omega \in \Omega$$

$$\sum_{r \in R(\omega)} t_r^v x_r^\omega \leq y_v + w_v \qquad v = 1, \ldots, n, \forall \omega \in \Omega$$

$$x^\omega \in \{0, 1\}^{|R(\omega)|} \qquad \forall \omega \in \Omega$$

$$y \in [s, e - w]$$

Solution Approach

Branch&Price.

Pricing Problem

The pricing problem is a shortest path problem with:

Pricing Problem

The pricing problem is a shortest path problem with:

- Positive costs for using an arc.

Pricing Problem

The pricing problem is a shortest path problem with:

- Positive costs for using an arc.
- Positive or negative costs for visiting a node.

Pricing Problem

The pricing problem is a shortest path problem with:

- Positive costs for using an arc.
- Positive or negative costs for visiting a node.
- Costs for visiting a node that are linear in time, with positive or negative coefficient.

Pricing Problem

The pricing problem is a shortest path problem with:

- Positive costs for using an arc.
- Positive or negative costs for visiting a node.
- Costs for visiting a node that are linear in time, with positive or negative coefficient.
- Time window constraints (exogenous).

Pricing Problem

The pricing problem is a shortest path problem with:

- Positive costs for using an arc.
- Positive or negative costs for visiting a node.
- Costs for visiting a node that are linear in time, with positive or negative coefficient.
- Time window constraints (exogenous).
- Capacity constraints.

Pricing Problem

The pricing problem is a shortest path problem with:

- Positive costs for using an arc.
- Positive or negative costs for visiting a node.
- Costs for visiting a node that are linear in time, with positive or negative coefficient.
- Time window constraints (exogenous).
- Capacity constraints.
- Elementarity constraint.

Pricing Problem

The pricing problem is a shortest path problem with:

- Positive costs for using an arc.
- Positive or negative costs for visiting a node.
- Costs for visiting a node that are linear in time, with positive or negative coefficient.
- Time window constraints (exogenous).
- Capacity constraints.
- Elementarity constraint.

The pricing problem can be solved separately for each scenario.

Pricing Problem

Relaxation:
Relax the elementarity condition.

Pricing Problem

Relaxation:
Relax the elementarity condition.

Basic Algorithm:
Ioachim et al. (1998) propose a labeling procedure to solve the shortest path problem with time window constraints and linear costs in time.

Pricing Problem

Relaxation:
Relax the elementarity condition.

Basic Algorithm:
Ioachim et al. (1998) propose a labeling procedure to solve the shortest path problem with time window constraints and linear costs in time.

Modifications:

Pricing Problem

Relaxation:
Relax the elementarity condition.

Basic Algorithm:
Ioachim et al. (1998) propose a labeling procedure to solve the shortest path problem with time window constraints and linear costs in time.

Modifications:

- Incorporate capacity constraints

Pricing Problem

Relaxation:
Relax the elementarity condition.

Basic Algorithm:
Ioachim et al. (1998) propose a labeling procedure to solve the shortest path problem with time window constraints and linear costs in time.

Modifications:

- Incorporate capacity constraints
- Apply 2-Cycle elimination (see for instance Irnich and Villeneuve (2003)).

Column Generation Algorithm

Version 1:
Solve the pricing problem for all scenarios.

Column Generation Algorithm

Version 1:
Solve the pricing problem for all scenarios.

Idea:
Do not solve the pricing problem for all scenarios.

Column Generation Algorithm

Version 1:
Solve the pricing problem for all scenarios.

Idea:
Do not solve the pricing problem for all scenarios.

Version 2:

- Solve the pricing problems iteratively.
- When a route for scenario ω is feasible and has negative reduced costs for scenario ω', add it to the model and do not solve the pricing problem for scenario ω'.

Branch&Price

Branch&Price is used to solve the TWAVRP:

Lower Bounds LP relaxation.

Upper Bounds Integer solutions to the LP relaxation.

Branching Special Ordered Subset (SOS) branching on arcs.

Computational Experiments

For the numerical experiments, several instances of the TWAVRP are generated for different numbers of customer locations:

Locations: Uniformly distributed in a square (5×5) area around the depot.

Travel Costs: Equal to the Euclidean distance.

Travel Time: Equal to the Euclidean distance.

Exogenous Time Windows: Three versions $[10, 16]$, $[9, 18]$ and $[7, 21]$ randomly assigned to customers at fixed frequency $\{0.1, 0.6, 0.3\}$ respectively. The depot has time window $[6, 22]$.

Time Window Width: 2.

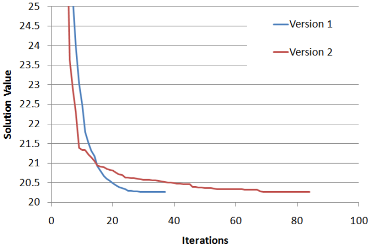
Demand Distribution: Normally distributed rounded up, with different randomly generated mean per customer and variance 1. The mean was generated using a normal distribution with mean 5 and variance 1 for each customer.

Vehicle Capacity: 30.

Scenario Distibution: Equal probabilities.

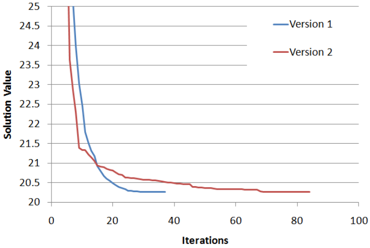
Computational Experiments: LP relaxations

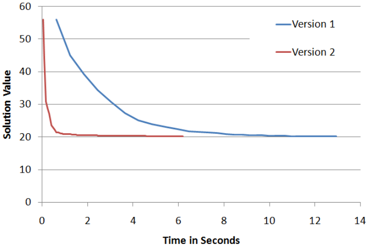
15 Customers using 10 scenarios



Computational Experiments: LP relaxations

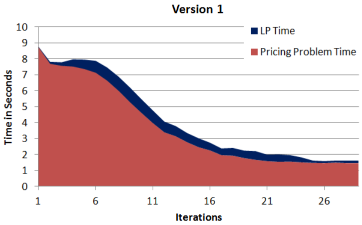
15 Customers using 10 scenarios





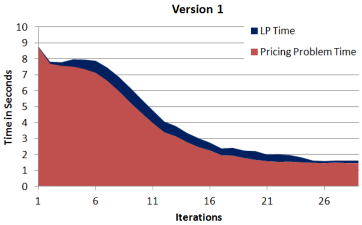
Computational Experiments: LP relaxations

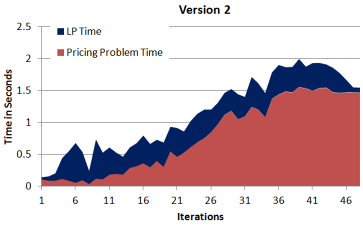
15 Customers using 100 scenarios



Computational Experiments: LP relaxations

15 Customers using 100 scenarios





Computational Experiments: LP relaxations

Table 1 : Averaged (over 25) computational results, 10 scenarios

	CGA Version 1			CGA Version 2		
		Time			Time	
#Cust.	Total	LP	Pricing	Total	LP	Pricing
10	6.462	0.148	6.311	2.418	0.273	2.138
15	23.394	0.380	23.001	9.512	0.392	9.112
25	128.808	1.978	126.789	50.438	1.134	49.260
35	396.004	6.844	389.043	160.925	2.698	158.118
50	1278.150	24.010	1253.810	554.119	7.547	546.243

Computational Experiments: LP relaxations

Table 1 : Averaged (over 25) computational results, 10 scenarios

	CGA Version 1			CGA Version 2		
		Time			Time	
#Cust.	Total	LP	Pricing	Total	LP	Pricing
10	6.462	0.148	6.311	2.418	0.273	2.138
15	23.394	0.380	23.001	9.512	0.392	9.112
25	128.808	1.978	126.789	50.438	1.134	49.260
35	396.004	6.844	389.043	160.925	2.698	158.118
50	1278.150	24.010	1253.810	554.119	7.547	546.243

Table 2 : Averaged (over 25) computational results, 15 customers

	CGA Version 1			CGA Version 2		
		Time			Time	
#Scenarios	Total	LP	Pricing	Total	LP	Pricing
2	4.643	0.117	4.520	2.992	0.119	2.870
5	12.224	0.153	12.065	5.451	0.196	5.247
10	26.132	0.412	25.708	9.730	0.367	9.351
25	68.393	2.006	66.360	24.422	1.446	22.952
50	140.090	6.287	133.754	49.311	4.367	44.896
100	296.201	16.405	279.699	144.515	15.169	129.254

Computational Experiments: Branch&Price

Table 3 : 3 Scenarios

Inst.	# Loc.	LP gap	Opt Found	5%	1%	Opt Proven	Gap
1	8	3.086	34.554	26.177	162.069	547.577	-
2	8	2.470	15.585	6.381	33.665	43.805	-
3	8	3.746	2.324	2.324	58.563	110.074	-
4	8	2.500	26.161	1.825	26.348	31.153	-
5	8	1.934	39.905	22.136	40.576	64.194	-
6	9	4.775	33934.1	393.027	28768.8	45170.1	-
7	9	4.259	18.782	18.798	97.453	399.751	-
8	9	5.053	11699.7	7325.59	28573.2	313155	-
9	9	-	-	-	-	-	1.15
10	9	0.776	2.106	2.106	2.106	34.227	-
11	10	0.153	1.466	0.842	0.842	2.839	-
12	10	4.940	5969.51	2111.48	9848.06	31036.7	-
13	10	-	-	-	-	-	17.87
14	10	8.223	13.697	296933	298143	298144	-
15	10	3.980	5791.97	241.317	4278.51	6593.7	-
16	12	-	-	-	-	-	13.59
17	12	2.442	312.391	5.944	315.121	2844.88	-
18	12	4.239	26212.8	397.208	31454.1	166823	-
19	12	4.914	74315.3	47.455	6239.62	91851.6	-
20	12	-	-	-	-	-	1.75

Conclusions

- Adding columns in multiple scenarios speeds up the column generation algorithm significantly.

Conclusions

- Adding columns in multiple scenarios speeds up the column generation algorithm significantly.
- The LP bound with 2-cycle elimination is pretty tight.

Conclusions

- Adding columns in multiple scenarios speeds up the column generation algorithm significantly.
- The LP bound with 2-cycle elimination is pretty tight.
- The optimal solution is found relatively early in the branching tree.

Conclusions

- Adding columns in multiple scenarios speeds up the column generation algorithm significantly.
- The LP bound with 2-cycle elimination is pretty tight.
- The optimal solution is found relatively early in the branching tree.
- The Branch&Price algorithm spends most time on closing a small gap.

References

- Ioachim, I., Gélinas, S., Soumis, F. and Desrosiers, J. 1998, 'A Dynamic Programming Algorithm for the Shortest Path Problem with Time Windows and Linear Node Costs', *Networks*, 31, pp. 193-204.
- Irnich, S. and Villeneuve, D. 2003,'The Shortest Path Problem with Resource Constraints and k-Cycle Elimination for $k \geq 3$', *Les Cahiers du GERAD*, 55.