A hybrid optimal method to the vehicle routing problem with time windows

Brian Kallehauge

Center for Traffic and Transport Research

Technical University of Denmark

ROUTE 2000
INTERNATIONAL WORKSHOP ON VEHICLE ROUTING

SKODSBORG, DENMARK - AUGUST 16-19, 2000

Outline

- Introduction
- Mathematical programming formulation of the VRPTW
- Shortest path decompositions
  - Dantzig-Wolfe decomposition
  - Lagrangian relaxation
- Cutting-plane algorithm with trust-region using several subgradients at each point
- The hybrid method
- Solomon's problem sets
- Results
- Conclusion
- Further developments

This presentation considers the VRPTW with hard time windows and identical vehicles where the objective is to minimize the total distance travelled.

To date, two equivalent decomposition methods have been developed for the VRPTW, namely Dantzig-Wolfe decomposition, proposed by Martin Desrochers, Jacques Desrosiers and Marius Solomon and published in 1992, and Lagrangian relaxation, proposed by Niklas Kohl and Oli B.G. Madsen, developed around 1993 but first published in 1997.

Both methods split the constraints into the same two sets, yielding the same subproblem, namely the elementary shortest path problem with time windows and capacity constraints. The subproblem is denoted a pricing problem in the Dantzig-Wolfe column generation context and the Lagrange problem in the Lagrangian relaxation context.

The master problems, on the other hand, are different but it is well known that they provide the same lower bound on the VRPTW in a branch and bound context.

In the Dantzig-Wolfe decomposition method the master problem is a set partitioning problem. The linear programming relaxation of the master problem is solved using the simplex method. The dual variables of the pricing problem are determined by the simplex multipliers.

In the Lagrangian relaxation method the master problem can be formulated as the minimization of a convex nonsmooth function, piecewise affine in the Lagrangian multipliers. This problem is denoted the dual problem. Methods from the field of nonsmooth optimization can therefore be applied to this problem. Several nonsmooth methods have already been applied to the dual problem, e.g. subgradient methods and bundle methods. Previous to this work these methods did not seem to be competitive compared to the Dantzig-Wolfe column generation method.

Jesper Larsen showed in his recent research on the parallelization of the branch and bound method that the computational time in the root node in Dantzig-Wolfe decomposition can be a serious problem. This is due to negative cost cycles in the subproblem, which is caused by large initial multipliers. In classical Dantzig-Wolfe column generation the multipliers are not directly controlled.

A way to control the multipliers is by considering a Lagrangian based method. In the Lagrangian relaxation based method one can choose small initial multipliers and control the step sizes to the optimal level. Although the dual solution is not controllable we believed that this would create easier instances of the subproblems and lead to faster multiplier convergence.

Formulation of the VRPTW

$$\min \sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk}$$

b.a.

$$\sum_{k\in V}\sum_{i\in N} x_{ijk} = 1 \quad \forall j \in C \tag{1}$$

$$\sum_{i\in C} d_i \sum_{j\in N} x_{ijk} \leqslant q \quad \forall k \in V \tag{2}$$

$$\sum_{j\in N} x_{0jk} = 1 \quad \forall k \in V \tag{3}$$

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0 \quad \forall h \in C, \forall k \in V \tag{4}$$

$$\sum_{i\in N} x_{i,n+1,k} = 1 \quad \forall k \in V \tag{5}$$

$$s_{ik} + t_{ij} - K(1 - x_{ijk}) \leqslant s_{jk} \quad \forall i \in N, \forall j \in N, \forall k \in V \tag{6}$$

$$a_i \leqslant s_{ik} \leqslant b_i \quad \forall i \in N, \forall k \in V \tag{7}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in N, \forall j \in N, \forall k \in V \tag{8}$$

This presentation considers a linear integer programming formulation of the VRPTW, which is well known from the literature. The first paper on an optimal method to the VRPTW by Kolen, Kaan and Trienekens in 1987 introduced the total distance travelled as the objective function and this has later been the convention in all the following optimal methods except a branch and cut method by Kontoravdis presented in 1997 where the primary objective is to minimize the number of vehicles.

In the work presented here we have discovered that due to the hard time windows the vehicles often wait before servicing a customer. For problems with wide time windows this means that in some cases the driver spends 75% of the total time waiting - in this formulation at no cost! This example is an indication of that the objective from the strictly geographical vehicle routing problem is generally not compatible when time constraints are introduced. There is therefore a need to introduce a more general objective function in the VRPTW such that it is no longer the total distance which is minimized but the total time, i.e. the sum of the driving time and the waiting time. This means that waiting costs are introduced in the shortest path subproblems.

The model can be characterized as a multicommodity network flow model with time windows and capacity constraints. In fact Tomlin already in 1966 proposed a Dantzig-Wolfe decomposition of the classical multicommodity problem where the subproblems were shortest path problems. The Dantzig-Wolfe decomposition of the VRPTW can therefore be viewed as a generalization of the method by Tomlin.

Shortest path decompostions

Dantzig-Wolfe decomposition

$$\min \sum_{p \in P'} c_p y_p$$
$$\sum a_{ip} y_P = 1 \quad \forall i \in C$$
$$\sum_{p \in P'} y_p = |V|$$
$$y_p \geqslant 0 \quad \forall p \in P'$$

- Approximate primal solution (aggregated convex combination of paths)
- Column-generation mechanism

Lagrangian relaxation

$$\Theta(\lambda) := \min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} - \sum_{j \in N} \lambda_j \left(\sum_{k \in V} \sum_{i \in N} x_{ijk} - 1 \right)$$

- $-\Theta$ is a convex piecewise affine function in $\lambda = (\lambda_1, \ldots, \lambda_n)$

Affine function. Let $f : X \mapsto R^n$. Then, f is affine if X is a convex set and $f(ax + (1-a)y) = af(x) + (1-a)f(y)$ for all x, y in X and a in $[0,1]$. Equivalently, f is affine if it is both convex and concave. Moreover, if $X = R^n$, f is the translation of a linear function: $ax + b$.

Nonsmooth optimization

The dual problem

$$\min\{-\Theta(\lambda) : \lambda \in \mathbb{R}^n\}$$

Subdifferential

The vector of constraint values

$s_j(\lambda) = \sum_{k \in V} \sum_{i \in N} x_{ijk} - 1$

is a subgradient for $-\Theta(\lambda)$ in λ.

- The subdifferential $\partial\Theta(\lambda)$ is the set of all subgradients at λ.
- General assumption in NSO: At every point we know the function value and one subgradient.
- In this application several paths are known at each point - *multiple pricing in column generation.*

Cutting-plane algorithm with trust-region using several subgradients at each point

- Kelley-Cheney-Goldstein cutting-plane algorithm for convex programs (1959-1960).

- Stabilizing principle from NSO - force the next iterate to be a priori in a box centered at the current point and having side lengths 2γ.

- More than one subgradient is available at each point.

  $$\min \theta$$
  $$\theta \geqslant \Theta(\lambda_i) + s(\lambda_i)^T(\lambda - \lambda_i) \quad \text{for } i = 0, \ldots, I$$
  $$\|\lambda\|_\infty \leqslant \gamma$$

- Linear program with $n+1$ variables and $2n+I$ constraints.

- Approximation of the dual function - the problem is that there are exponential many pieces.

- Update the trust-region according to how well the approximation fits the dual function.

- The cutting-plane program (row generation mechanism) is the dual LP to the DW decomposition master program but is stabilized to obtain faster convergence and initialized with small initial multipliers (easier subproblems).

- If the current iterate (multipliers) is better (closer to the solution), and the next iterate is worse (goes further awayfrom this solution); the algorithm is *unstable.*

The hybrid method

- Phase 1: Cutting-plane algorithm
  - The starting point $\lambda_0 = 0$.
  - Solving the dual problem using the cutting-plane algorithm.
- Phase 2: Dantzig-Wolfe algorithm (Jesper Larsen 1999)
  - Inserting the columns found in phase 1 in the Dantzig-Wolfe master LP.
  - Generation of subtour elimination constraints and 2-path cuts in the root node.
  - Branching on vehicles, then on arcs.
  - Column reduction to examine a larger part of the branch-and-bound tree.

Solomons problem sets

- geographical data
- number of customers serviced by a vehicle
- number of customers with restrictive time windows
- how restrictive the time windows are
- position of the time windows

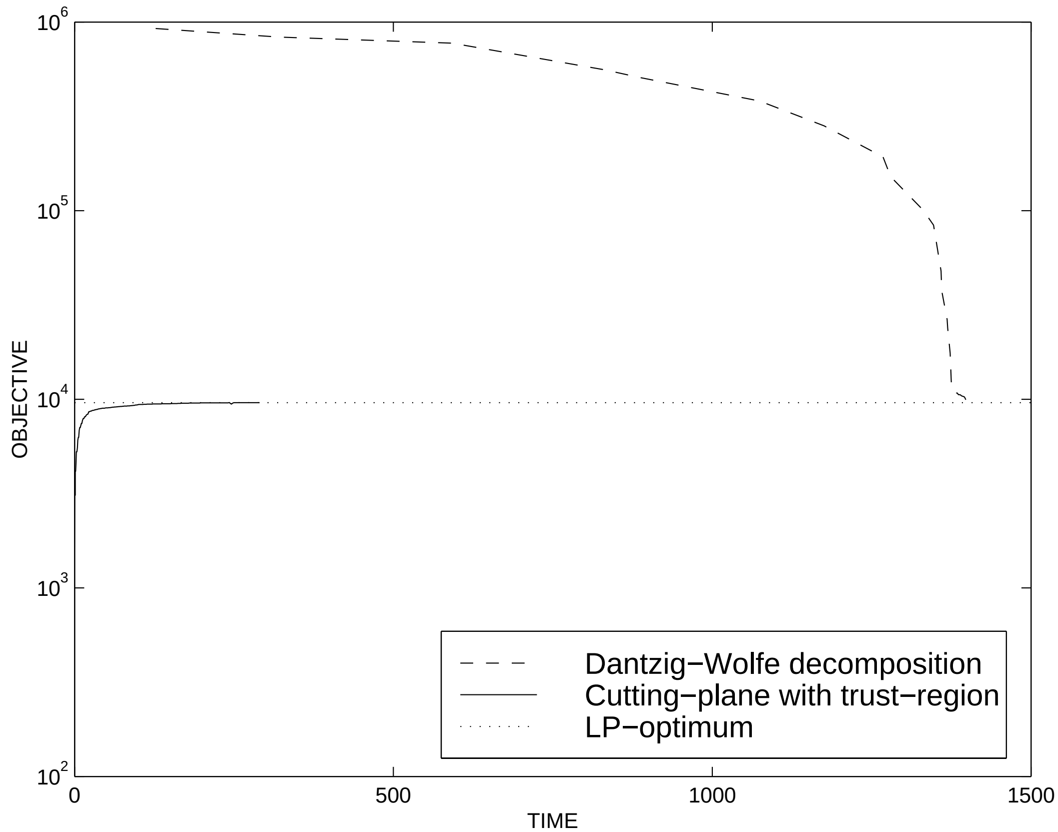
Problem set	**Num. of problems**	$[a_0, b_0]$	bt_i	q
R1	36	$[0, 230]$	10	200
C1	27	$[0, 1236]$	90	200
RC1	24	$[0, 240]$	10	700
Total	87			

Problem set	**Num. of problems**	$[a_0, b_0]$	bt_i	q
R2	33	$[0, 1000]$	10	1000
C2	24	$[0, 3390]$	90	200
RC2	24	$[0, 960]$	10	1000
Total	81			

Geographical data for Solomon's problem sets

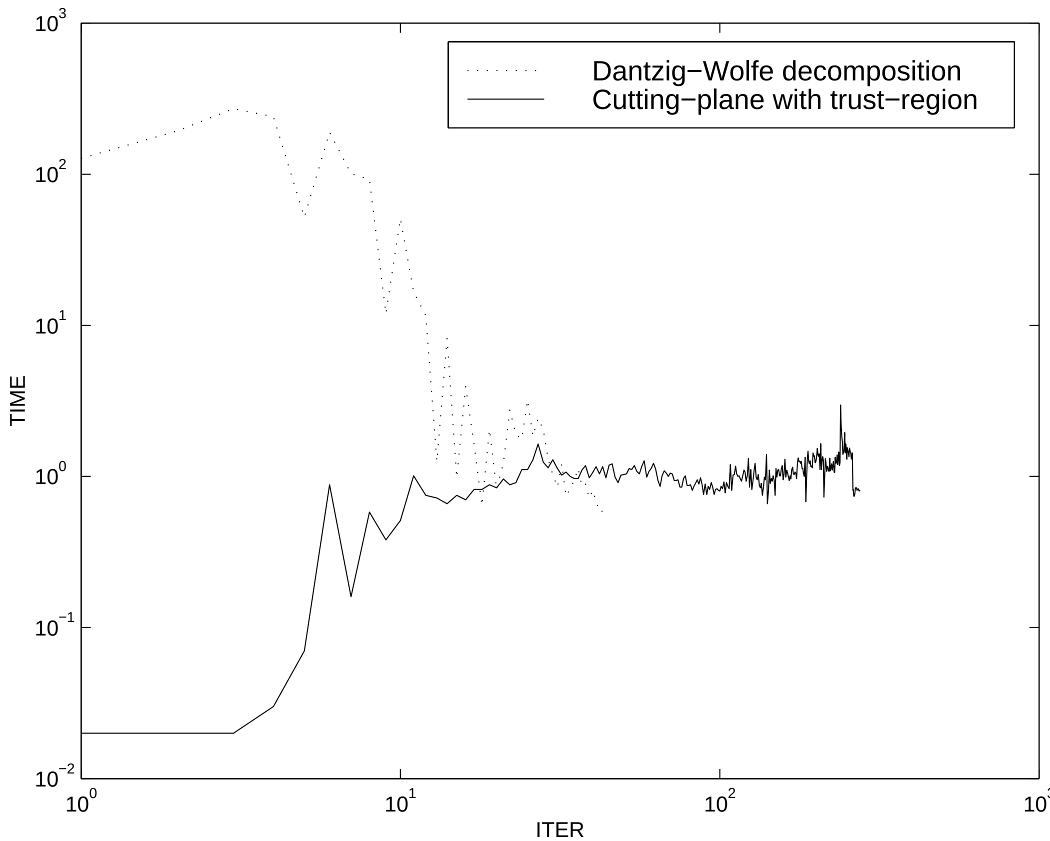
Cutting-plane with trust-region compared to Dantzig-Wolfe

The Dantzig-Wolfe primal lower bound and the Lagrangian dual lower bound for R104.100 is shown on the y-axis and the accumulated computational time is shown on the x-axis. (A logarithmic (base 10) scale is used for the y-axis.)



The difficulty of the subproblems in the cutting-plane method compared to the Dantzig-Wolfe method

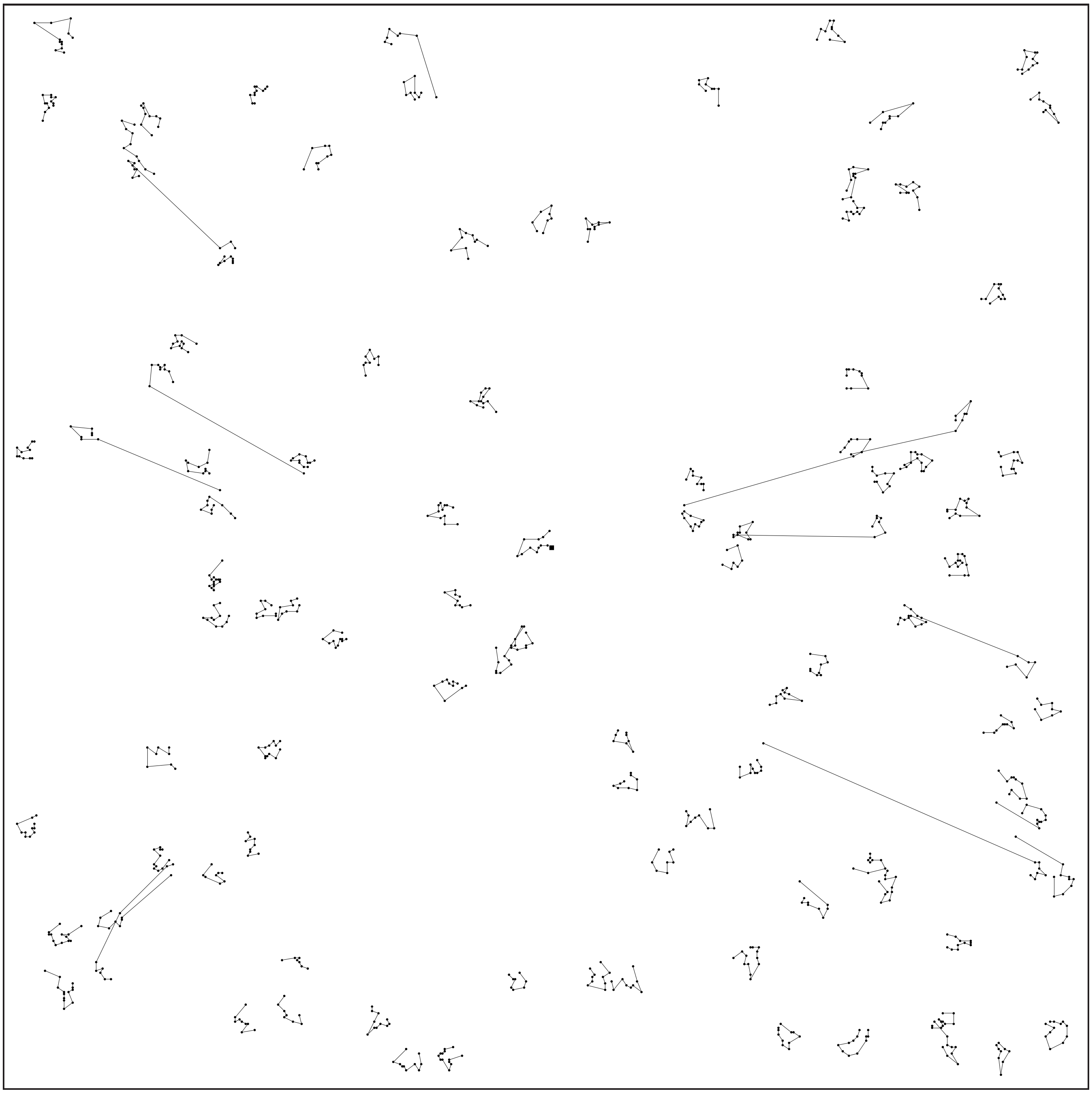
Number of seconds used in the shortest path subproblem (y-axis) versus the iteration (x-axis) for Dantzig-Wolfe and cutting-plane applied on R104.100. (A logarithmic (base 10) scale is used for the x-axis and y-axis.)



R1 problems with 100 customers

Problem	LB_{opt}	Cutting-plane iter	Cutting-plane time	Dantzig-Wolfe iter	Dantzig-Wolfe time
R101	1631.15	131	17.43	22	1.08
R102	1466.60	135	83.03	41	38.31
R103	1206.31	301	227.69	50	557.97
R104	949.50	276	288.77	44	1408.69
R105	1346.14	150	40.61	23	2.94
R106	1226.44	214	106.81	41	465.41
R107	1051.84	381	295.69	44	3930.36
R108	907.16	307	434.43	48	3419.23
R109	1130.59	228	88.99	36	44.39
R110	1048.48	223	131.40	28	245.07
R111	1032.03	276	200.93	39	466.79
R112	919.19	213	261.73	34	2843.32
Total		2835	2177.51	661	13423.56

An instance with 1000 customers - C110_1.1000 (Homberger)



C110_1.1000

```
---------- Statistics
This program ran on serv3 ().
Total execution time                      24836.17 seconds
         (Solving root 23245.11 seconds)
Time used in separation                      34.25 seconds
         Cuts generated            2
Accumulated time used in calls of SPPTWCC  870.12 seconds
Time used in largest single SPPTWCC call     9.41 seconds
Branching nodes examined 3 (Veh 0, Arc 1, TW 0)
        (hereof 0 where not feasible)
No of calls to SPPTW 292, Routes generated 53294
Max no of columns selected per SPPTW 200
No of multiple customers deleted explicitly 0
IP value 424448
RP value 424446.833
LP value 424444.000
-------------------------------------------------------
Total CPLEX optimize time 23872.30 Biggest 1000.05
Total branching time 23.49 Biggest 23.49
```

Conclusion

- The computational times for the dual problem (root node) has been reduced significantly by applying a cutting-plane method with trust-region, especially for the Solomon type 2 problems where many customers are serviced on each route.

- We have succeded in solving 14 previously unsolved type 2 problems to optimality on a single processor system compared to recent computational studies with up to 32 processors (Cook and Rich 1999).

- The solutions for the problems R201.100 RC201.100 are the first optimal solutions reported on these problem sets containing 100 customers.

- We have solved an extended C1 Solomon problem (Homberger) with 400 customers (C1_4_1.400) and an extended C1 problem with 1000 customers (C110_1.1000).

- The problem containing 1000 customers is the biggest VRPTW problem ever solved to optimality. The biggest problems so far solved have contained up to 200 customers.

Number of solved Solomon problems

		Type 1				Type 2				
Author	Method	R1	C1	RC1	I alt	R2	C2	RC2	Total	
Larsen (1999)	DW	29	27	18	74	6	8	3	17	91
Cook og Rich (1999)	DW	33	27	20	80	8	20	2	30	110
Previously solved problems		33	27	20	80	8	20	4	32	112
This project		33	27	20	80	14	23	9	46	126
Number of problems		36	27	24	87	33	24	24	81	168

Further developments

- Deal with the relaxation from ESPPTWCC to SPPTWCC - 2 approaches:
  - Use an algorithm to solve the ESPPTWCC. Irinia Dumitrescu and Natashia Boland (ISMP 2000) reported solution for ESPPCC (100 nodes, 4000 arcs).
  - Design branching and cutting strategies to eliminate cycles using the current SPPTWCC algorithm.
- Use methodologies from VRP in Solomon type 2 problems. K-tree relaxation.

Non-elementary paths

Natashia Boland (2000): What are the gaps between the E-LP and NE-LP?

In the LP-solution before insertion of cuts :

- Check how many customers that are repeated on a path, i.e. are included in some cycle, 3-cycle, 4-cycle etc.

- Check how many paths that are non-elementary, i.e. contain cycles.

- Measure the sum of the LP-variables representing non-elementary routes over the total sum of the variables, i.e. the fraction of flow on paths with cycles in LP-optimum.

Problem	NE customers	NE paths/all paths	NE flow/total flow
R101.100	0	0/26	0/19.5
R102.100	0	0/18	0/18.0
R103.100	8	7/39	1.22/14.06
R104.100*	49	41/78	4.18/10.16
R105.100	0	0/59	0/14.88
R106.100	8	9/74	1.37/13.00
R107.100	23	29/78	2.42/10.95
R108.100*	34	32/73	3.63/9.82
R109.100	14	14/78	1.70/12.23
R110.100	24	25/89	2.86/10.96
R111.100	17	20/91	1.87/11.43
R112.100*	41	36/86	2.79/9.49
R204.25*	19	4/4	1.25/1.25
RC201.25	3	1/4	0.50/3.00
RC202.25	15	4/6	1.08/1.83
RC203.25	25	7/7	1.17/1.17
RC204.25*	25	8/8	1.00/1.00
RC205.25	14	7/9	1.00/2.00
RC206.25	25	9/9	1.50/1.50
RC207.25	25	14/14	1.15/1.15
RC208.25*	25	12/12	1.00/1.00