The Split Delivery Vehicle Routing Problem:
Using Mixed Integer Programming within a Heuristic Framework

by

Si Chen, University of Maryland
Bruce Golden, University of Maryland
Edward Wasil, American University

Presented at the Lunteren Conference on the Mathematics of Operations Research
The Netherlands, January 2006

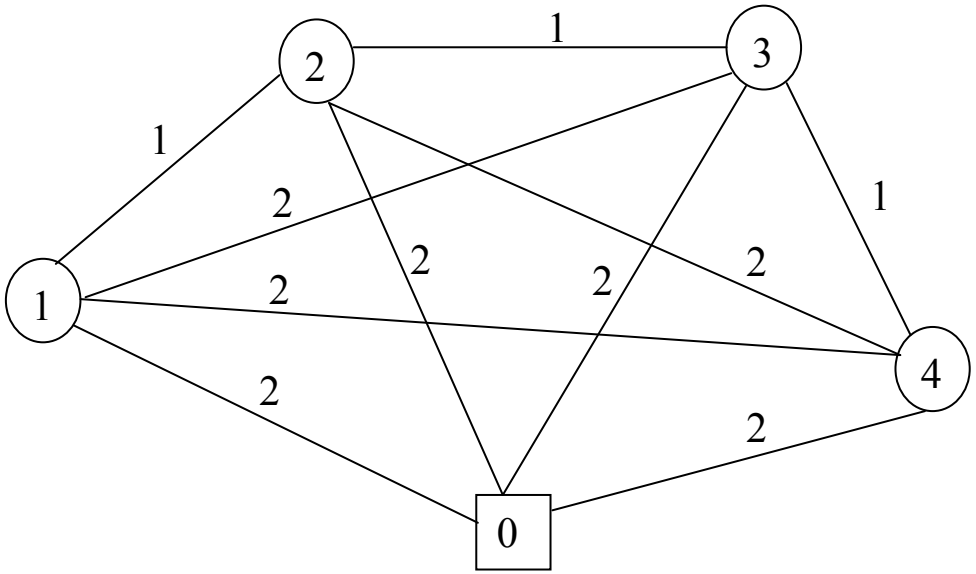
Introduction

- The split delivery vehicle routing problem (SDVRP)
  - A relaxation of traditional VRP
  - A customer's demand can be split between two or more vehicles
  - DT heuristic by Dror and Trudeau
  - SplitTabu heuristic by Archetti et al.
  - The potential exists to save vehicles and thus reduce distance traveled

Applications

- Livestock feed distribution (via trucks) to pens at a ranch
- Routing helicopters for crew exchange at off-shore locations
- Distributing bundles of newspapers in Korea
- Containerized sanitation pick-up/commercial collection

Example (from Archetti et al.)



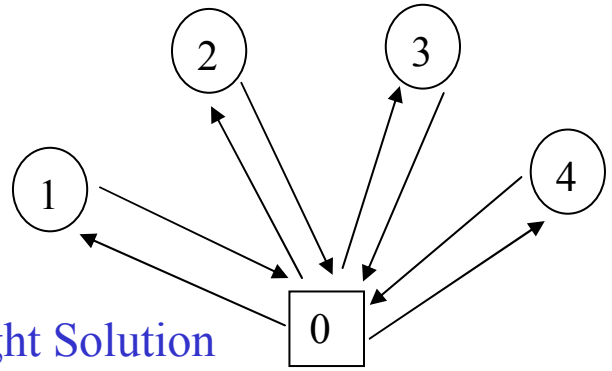
Node 0 is the depot

There are four customer nodes

Each customer has a demand of 3

Vehicle capacity is 4 units

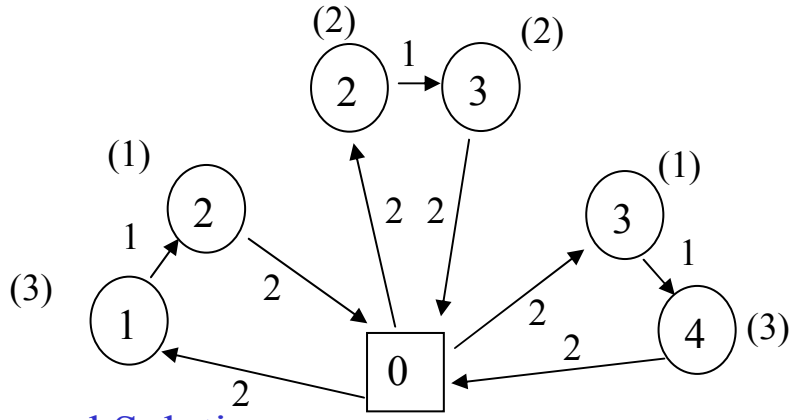
Distances are shown above



Clarke-Wright Solution

Distance = 4 x (2 + 2) = 16

4 vehicles are used



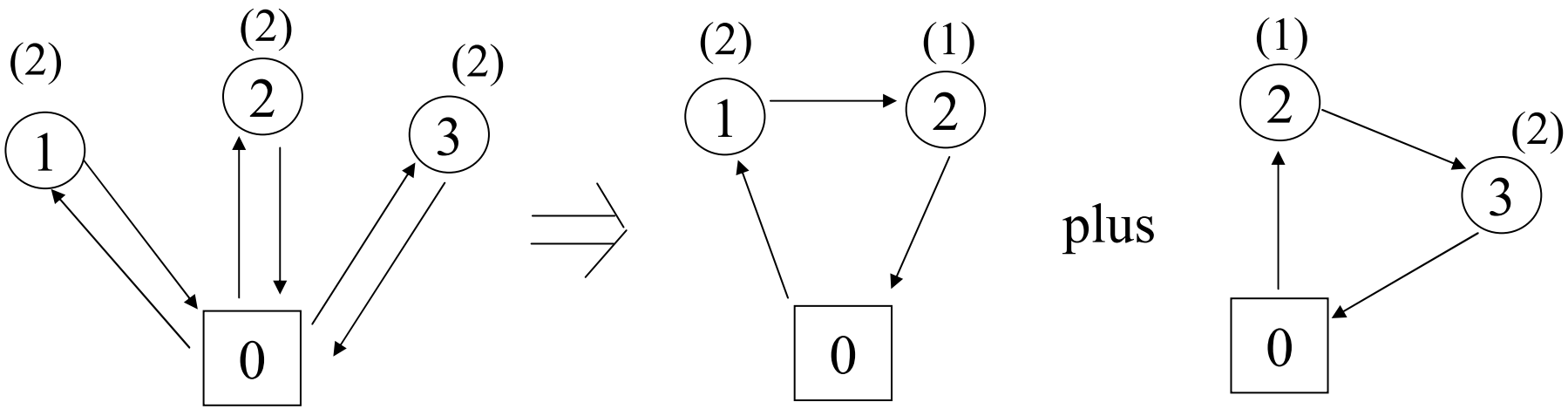
Improved Solution

Distance = 3 x 5 = 15

3 vehicles are used

DT Heuristic

- A k-split interchange splits the demand of customer *i* among k routes provided that
  - capacity constraints are obeyed, and
  - total distance is reduced

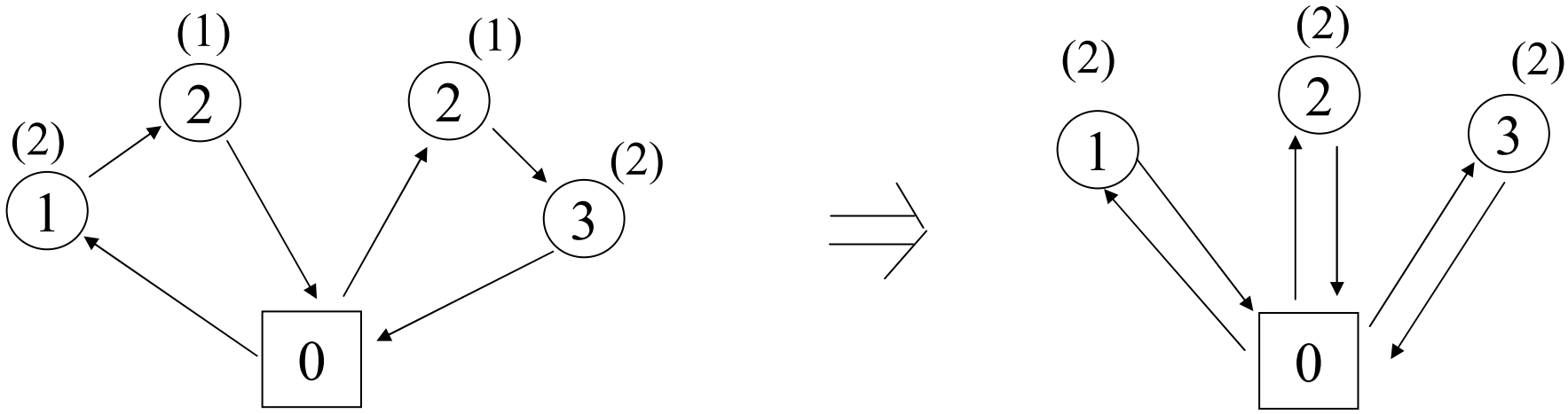


vehicle capacity = 3

Example of a 2-split interchange

DT Heuristic

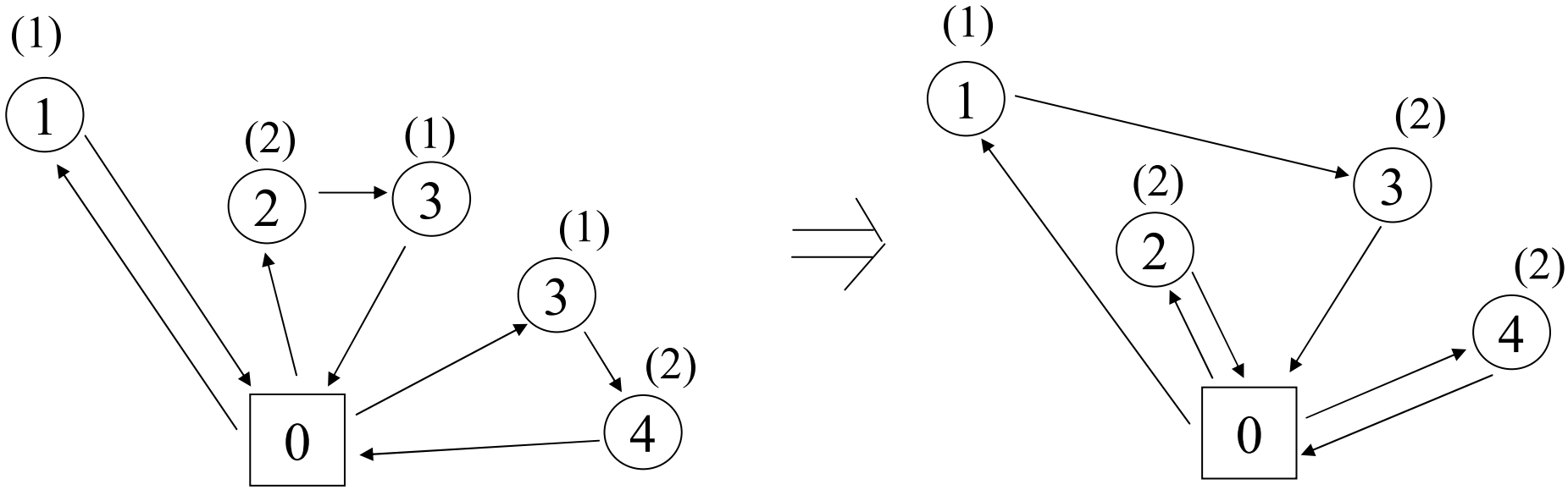
- Route addition serves as an inverse operation to a k-split interchange
  - Add a route to eliminate a split delivery
  - Do so if it reduces total distance



Example of a route addition

Tabu Search Moves (Archetti et al.)

- These moves can be more complex
  - Remove a customer from a set of routes and insert it into a new route or an existing route with enough residual capacity (see below)
  - Other insertion moves are also possible



vehicle capacity = 3

Computational Results for the DT Heuristic

- Based on a Euclidean problem with 75 nodes
- Vehicle capacity is 160 units
- Six demand scenarios

a)	[0.01 – 0.1]	d)	[0.1 – 0.9]
b)	[0.1 – 0.3]	e)	[0.3 – 0.7]
c)	[0.1 – 0.5]	f)	[0.7 – 0.9]

- Demand *i* in scenario [α – β] is generated randomly from a uniform distribution on the interval [160 α , 160 β]
- There are 30 instances per scenario

Computational Results for the DT Heuristic

Scenario	Average Distance Reduction from CW	Average Number of Vehicles Saved from CW	CPU Time (seconds)
[0.01 – 0.1]	0.12	0.06	28.8
[0.1 – 0.3]	1.50	0.61	33.6
[0.1 – 0.5]	2.48	1.83	46.9
[0.1 – 0.9]	6.30	6.47	86.1
[0.3 – 0.7]	9.01	8.37	86.0
[0.7 – 0.9]	11.24	14.07	222.2

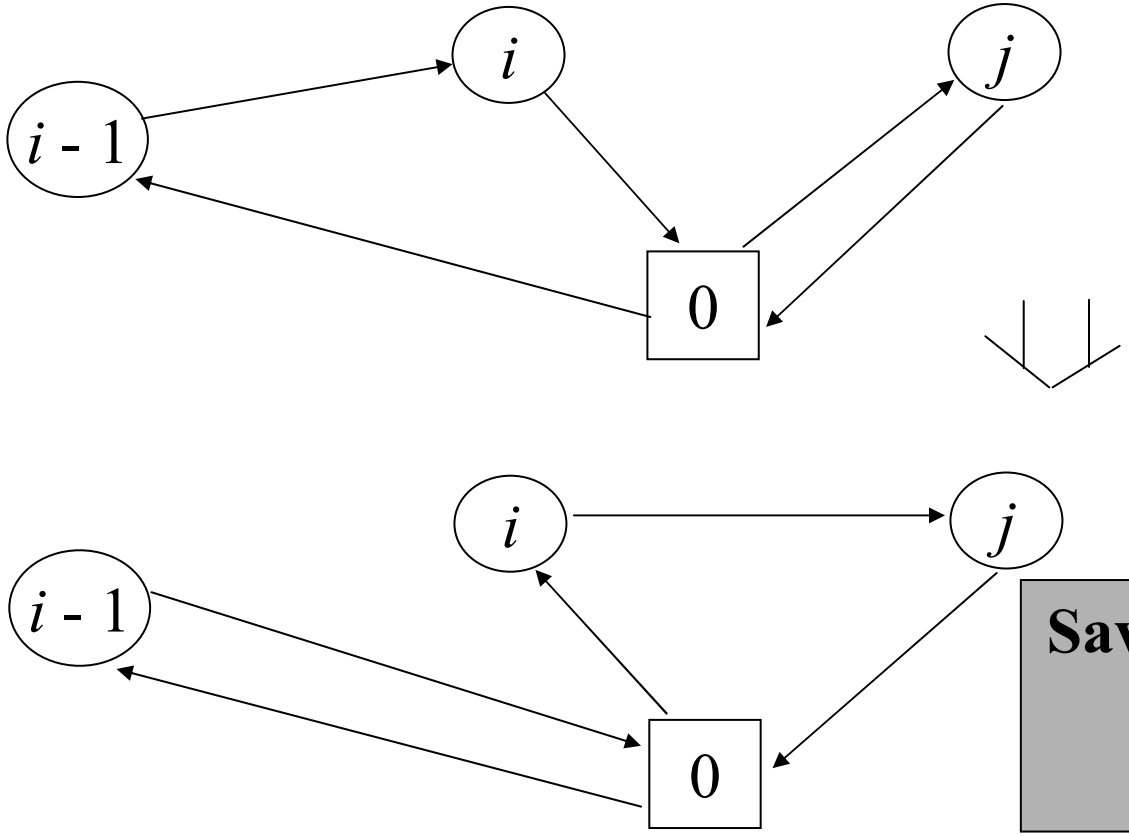
- Experiments run on a CYBER 173 with PASCAL Compiler
- The SDVRP has two components: routing and packing
- In the first two or three scenarios the greater emphasis is on routing
- The last three scenarios emphasize packing

A New MIP Approach

- Start with an initial solution (e.g., the CW solution)
- For each route in this solution, consider its one or two endpoints
- For each endpoint, we consider its l-closest endpoints to be its "neighbors" (l is a parameter)
- Each endpoint is allowed to reallocate its demand among its neighbors
- After this reallocation process, there are three possibilities for each endpoint (we assume symmetry here)
  - No change is made

A New MIP Approach

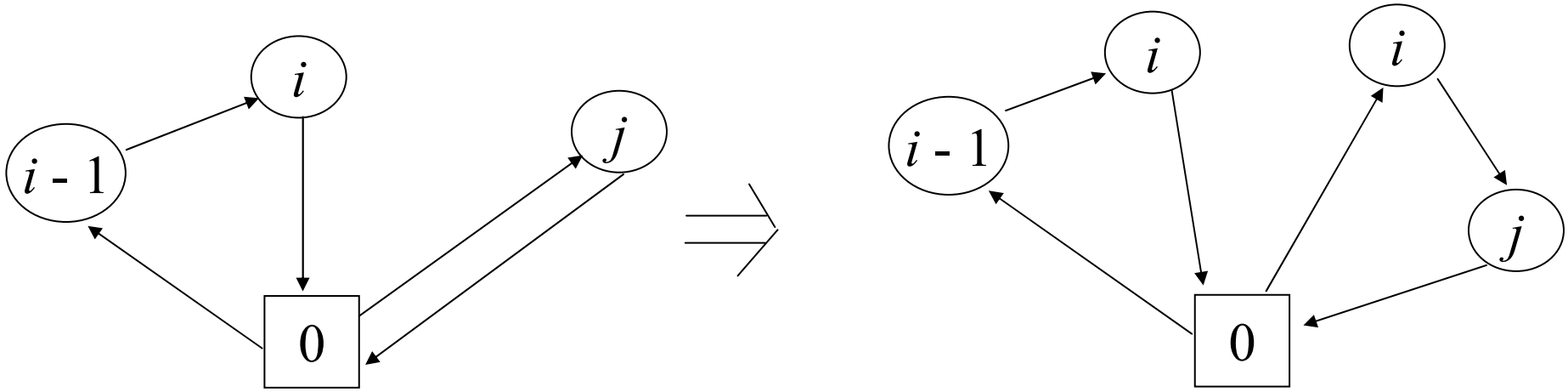
- The endpoint *i* is removed from its current route(s) and all of its demand is moved to another route or routes (see below for simple example)



$$\text{Savings} = (l_{i\text{-}1,\,i} + l_{i,\,0} - l_{i\text{-}1,\,0}) - (l_{0,\,i} + l_{i,\,j} - l_{0,\,j})$$

A New MIP Approach

- Part of the demand of endpoint *i* is moved to another route or routes



$$\textbf{Savings} = -(l_{0,i} + l_{i,j} - l_{0,j})$$

A New MIP Approach

- The objective is to find a set of reallocation operations that maximizes the total savings
- The Endpoint MIP formulation follows
- Definitions
  - i, j are endpoints of current routes
  - l_{ij} = length or distance between i and j
  - R_i = residual capacity on the route with i as an endpoint
  - D_i = demand of endpoint i carried on its route

A New MIP Approach

- Decision variables
  - d_{ij} = amount of endpoint i's demand moved before node j
  - $m_{ij} =$ 1 if i is inserted before j; 0 otherwise
  - $b_i =$ 1 if endpoint i's entire demand is removed from the route on which it was an endpoint; 0 otherwise

Endpoint MIP Constraints

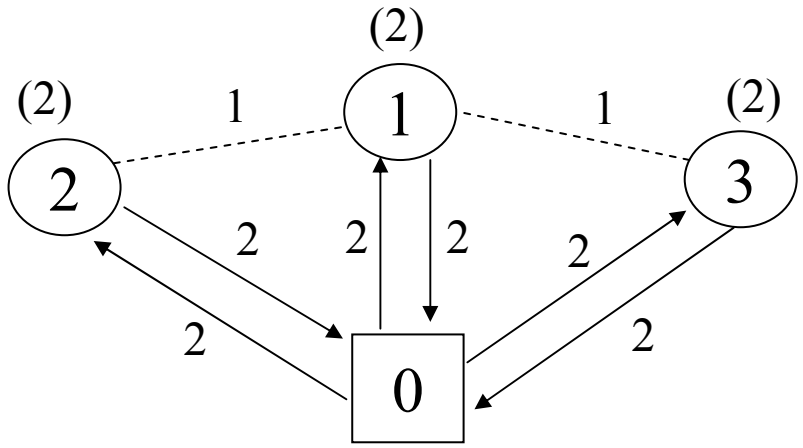
- The amount added to a route minus the amount taken away $\leq$ residual capacity
- The amount diverted from an endpoint on a route $\leq$ its demand on that route
- If an endpoint is removed from a route, all of its demand must be diverted to other routes
- If $d_{ij} > 0 \Rightarrow m_{ij} = 1$
- If an endpoint is removed from a route, no other endpoint can be inserted before it
- $d_{ij} \geq 0,\ b_i\ \&\ m_{ij} \in \{0, 1\}$

Objective Function via Small Example

Small Example

vehicle capacity = 3

total distance = 12



Maximize $2b_1\, l_{10} + 2b_2\, l_{20} + 2b_3\, l_{30} - m_{12}\,(l_{12} + l_{10} - l_{20}) - m_{13}\,(l_{13} + l_{10} - l_{30})$

$- m_{21}\,(l_{20} + l_{21} - l_{10}) - m_{23}\,(l_{20} + l_{23} - l_{30}) - m_{31}\,(l_{31} + l_{30} - l_{10})$

$- m_{32}\,(l_{32} + l_{30} - l_{20})$

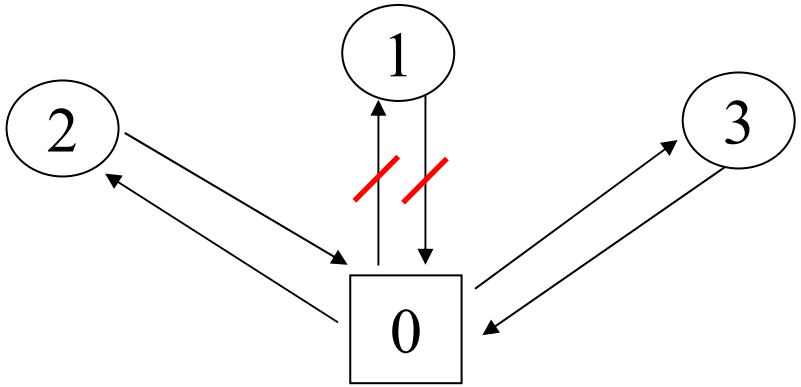
- Remember,

$m_{ij} =$ { 1 if i is inserted before j ; 0 otherwise

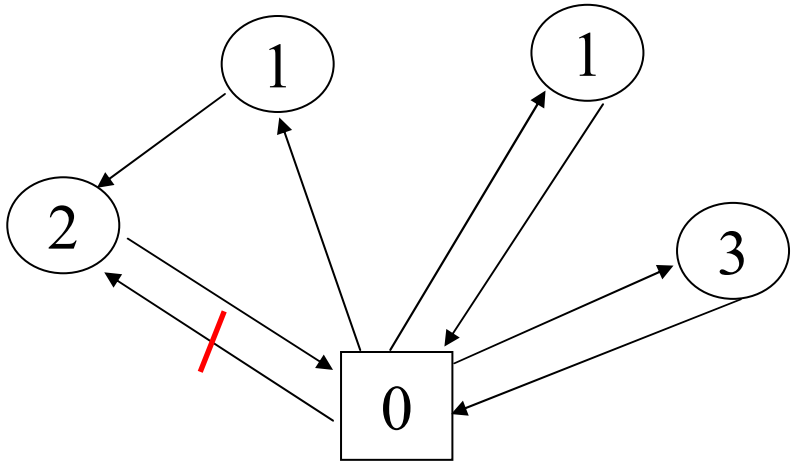
$b_i =$ { 1 if endpoint i's entire demand is removed from the route on which it was an endpoint ; 0 otherwise

Illustration of Objective Function

- If $b_1 = 1$, we save $2b_1 l_{10}$



- If $m_{12} = 1$, we save $-m_{12}(l_{12} + l_{10} - l_{20})$



Full Formulation for Small Example

$$\begin{aligned}\max \quad & 2b_1\, l_{10} + 2b_2\, l_{20} + 2b_3\, l_{30} - m_{12}\,(l_{12} + l_{10} - l_{20}) - m_{13}\,(l_{13} + l_{10} - l_{30}) \\ & - m_{21}\,(l_{20} + l_{21} - l_{10}) - m_{23}\,(l_{20} + l_{23} - l_{30}) - m_{31}\,(l_{31} + l_{30} - l_{10}) \\ & - m_{32}\,(l_{32} + l_{30} - l_{20})\end{aligned}$$

s.t.

$$\begin{aligned} d_{21} + d_{31} - d_{12} - d_{13} &\le \mathrm{R}_1 & \qquad d_{12} + d_{13} &\le \mathrm{D}_1 \\ d_{12} + d_{32} - d_{21} - d_{23} &\le \mathrm{R}_2 & \qquad d_{21} + d_{23} &\le \mathrm{D}_2 \\ d_{13} + d_{23} - d_{31} - d_{32} &\le \mathrm{R}_3 & \qquad d_{31} + d_{32} &\le \mathrm{D}_3 \end{aligned}$$

$$\begin{aligned} d_{12} + d_{13} &\ge \mathrm{D}_1\, b_1 & \qquad \mathrm{D}_1\, m_{12} &\ge d_{12} & \qquad \mathrm{D}_2\, m_{23} &\ge d_{23} \\ d_{21} + d_{23} &\ge \mathrm{D}_2\, b_2 & \qquad \mathrm{D}_1\, m_{13} &\ge d_{13} & \qquad \mathrm{D}_3\, m_{31} &\ge d_{31} \\ d_{31} + d_{32} &\ge \mathrm{D}_3\, b_3 & \qquad \mathrm{D}_2\, m_{21} &\ge d_{21} & \qquad \mathrm{D}_3\, m_{32} &\ge d_{32} \end{aligned}$$

Full Formulation for Small Example

- Continuation of constraints

$$1 - b_1 \geq m_{21} + m_{31}$$

$$1 - b_2 \geq m_{32} + m_{12}$$

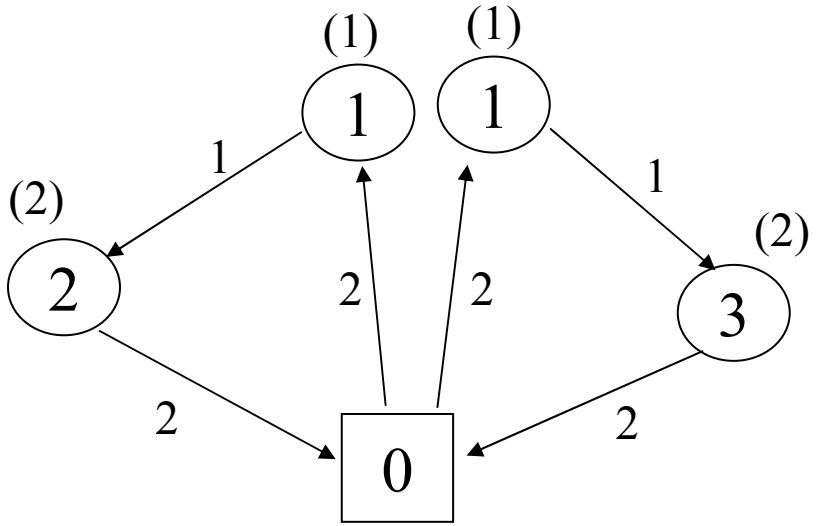
$$1 - b_3 \geq m_{23} + m_{13}$$

$$d_{ij} \geq 0$$

$$b_i \;\&\; m_{ij} \in \{0, 1\}$$

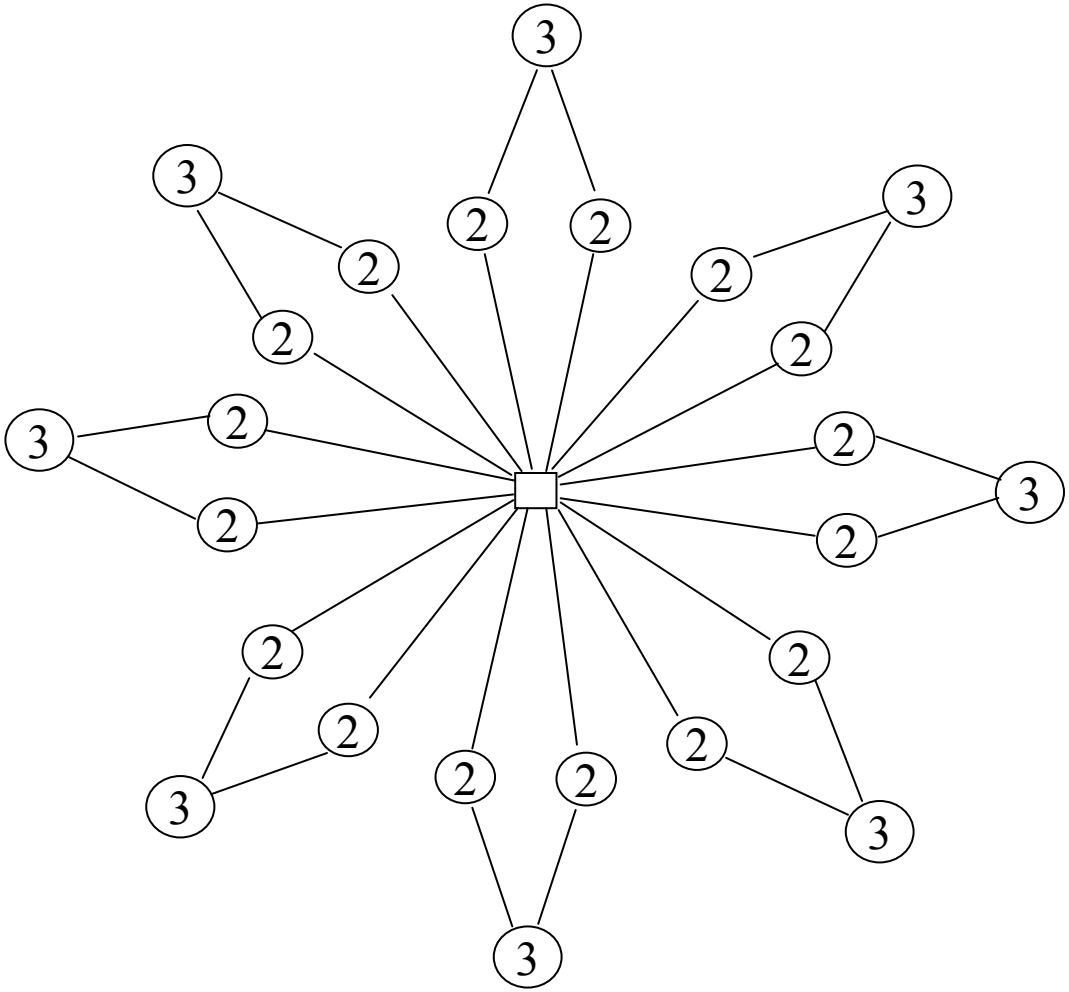
- MIP solution

  total distance = 10



Limitation of Endpoint MIP

- Not all feasible solutions can be reached, capacity = 8



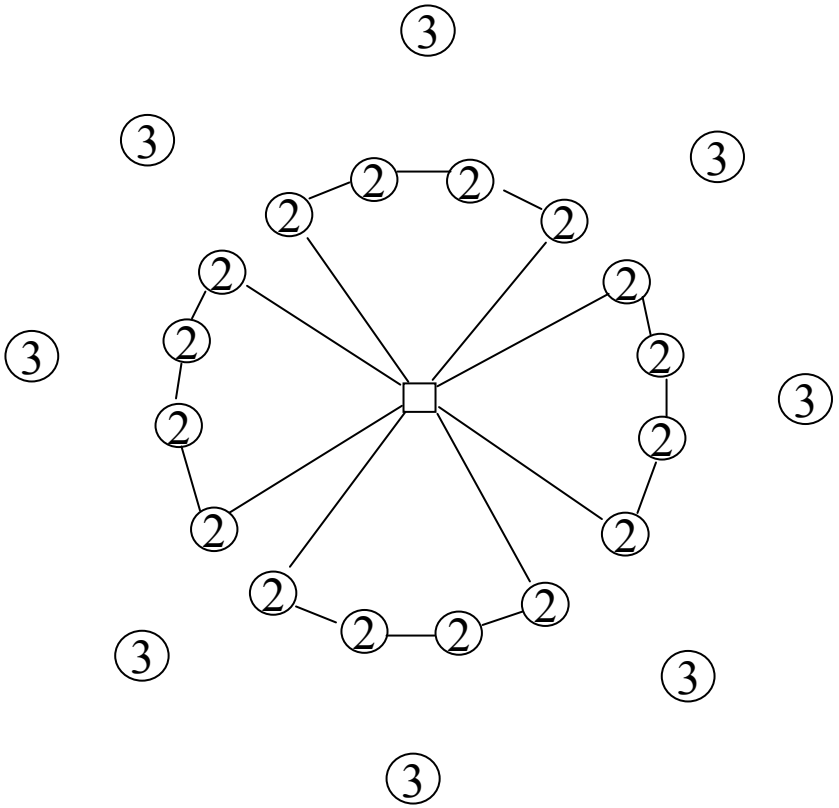
Initial solution

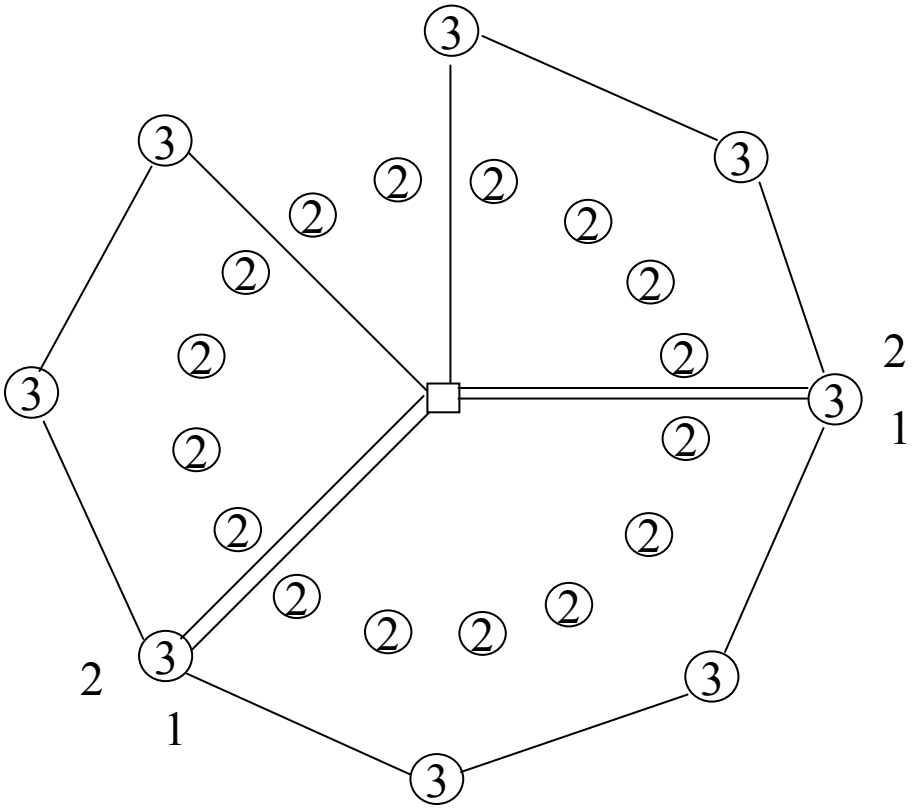
8 vehicles

Limitation of Endpoint MIP

- Improved solution

  7 vehicles





Computational Experiments

- Our experiments involved problems with 51, 76, 101, 121, 151, and 200 nodes (including the depot)
- Each problem set contains 30 instances for each of the six scenarios
- We also test on some benchmark problems (old and new)
- The Clarke-Wright solution is our starting solution
- Experiments are run on a Pentium 4 PC with 1.69 GHz and 512 MB RAM
- We use Cplex 7.0 with Visual C^{++} (Version 6.0)
- For post-optimization, we use the RTR travel algorithm of Li et al. (2005)
- The MIP is run twice

Comparison of the Two Heuristics

- We compare the MIP approach with the best tabu search variant from Archetti et al. (SplitTabu-DT)
- Archetti et al. use a Pentium 4 PC with 0.8 GHz and 256 MB RAM
- The comparison is approximate since SplitTabu-DT was only tested on one instance for each scenario in each problem set
- In contrast, we run 30 instances for each scenario in each problem set

Computational Comparison

Scenario	CW	MIP	MIP + PP	Time	Tabu	Time	% Better
[0.01 – 0.1]	49,955	49,755	45,721	1.9	46,376	4.8	1.41
[0.1 - 0.3]	78,776	76,129	72,356	3.4	76,140	21.8	4.97
[0.1 - 0.5]	105,456	99,117	94,385	14.7	100,867	28.2	6.43
[0.1 - 0.9]	156,833	145,124	140,833	55.4	146,992	60.8	4.19
[0.3 - 0.7]	164,461	143,482	140,867	47.9	149,690	48.6	5.89
[0.7 - 0.9]	240,196	206,841	205,600	135.4	216,521	106.0	5.04

- 51 nodes, capacity = 160

Computational Comparison

Scenario	CW	MIP	MIP + PP	Time	Tabu	Time	% Better
[0.01 – 0.1]	65,347	64,441	59,824	25.8	60,524	13.0	1.15
[0.1 - 0.3]	113,007	110,876	108,109	57.0	109,532	45.4	1.30
[0.1 - 0.5]	150,607	144,006	139,352	214.0	144,362	123.0	3.47
[0.1 - 0.9]	231,871	210,267	205,654	401.1	212,443	193.0	3.20
[0.3 - 0.7]	242,045	213,073	211,260	509.6	216,051	129.0	2.22
[0.7 - 0.9]	363,026	310,322	306,719	811.0	318,064	869.0	3.57

- 76 nodes, capacity = 140

Computational Comparison

Scenario	CW	MIP	MIP + PP	Time	Tabu	Time	% Better
[0.01 – 0.1]	69,755	67,240	65,143	53.9	64,874	57.8	-0.42
[0.1 - 0.3]	151,851	148,479	141,432	126.5	146,201	146.0	3.26
[0.1 - 0.5]	215,267	207,735	197,333	287.6	202,999	292.8	2.79
[0.1 - 0.9]	366,489	322,146	316,221	251.2	310,153	259.6	-1.96
[0.3 - 0.7]	381,021	317,790	313,455	716.5	303,802	777.8	-3.18
[0.7 - 0.9]	577,014	4,800,264	477,912	1024.3	486,779	1004.4	1.82

- 101 nodes, capacity = 200

Computational Comparison

Scenario	CW	MIP	MIP + PP	Time	Tabu	Time	% Better
[0.01 – 0.1]	102,583	100,353	98,516	36.4	108,470	42.4	9.18
[0.1 - 0.3]	291,471	285,080	256,889	136.4	291,871	142.6	11.99
[0.1 - 0.5]	425,153	400,437	368,705	220.7	420,612	268.0	12.34
[0.1 - 0.9]	751,617	632,431	607,913	722.8	658,397	877.8	7.67
[0.3 - 0.7]	756,257	625,696	612,395	605.4	663,955	658.6	7.77
[0.7 - 0.9]	1,223,830	899,813	894,178	725.4	1,030,408	1825.6	13.22

- 121 nodes, capacity = 200

Computational Comparison

Scenario	CW	MIP	MIP + PP	Time	Tabu	Time	% Better
[0.01 – 0.1]	96,249	94,420	87,515	107.8	89,095	172.8	1.77
[0.1 - 0.3]	199,700	194,325	184,495	308.0	191,825	393.2	3.82
[0.1 - 0.5]	281,996	266,581	253,292	630.5	263,271	739.2	3.79
[0.1 - 0.9]	465,426	410,578	394,537	2,220.0	390,972	2,278.0	-0.91
[0.3 - 0.7]	474,211	413,049	401,173	3,028.3	403,970	3,008.0	0.69
[0.7 - 0.9]	735,924	598,738	595,034	10,038.8	619,636	10,223.0	3.97

- 151 nodes, capacity = 200

Computational Comparison

Scenario	CW	MIP	MIP + PP	Time	Tabu	Time	% Better
[0.01 – 0.1]	114,303	114,245	104,019	413.4	105,627	525.8	1.52
[0.1 - 0.3]	250,659	242,205	225,865	618.5	238,415	754.8	5.26
[0.1 - 0.5]	359,392	340,604	320,256	1,775.7	328,447	2,668.0	2.49
[0.1 - 0.9]	592,915	521,442	509,460	3,038.1	485,383	3,297.2	-4.96
[0.3 - 0.7]	615,268	524,037	508,807	3,035.7	510,284	3,565.6	0.29
[0.7 - 0.9]	960,678	734,454	720,703	12,542.3	794,463	21,849.0	9.28

- 200 nodes, capacity = 200

Some Observations From Statistics

- If Tabu and MIP + PP are equally good with respect to quality of solution, then Tabu would beat the median MIP + PP result about half the time
- Using a binomial distribution with n = 36 and p = ½, we can test the null hypothesis that they are equally good
- Using a significance level of 0.01, the Tabu results should beat the median MIP + PP results in at least 10 out of 36 cases
- Since Tabu wins in only 5 out of 36 cases, the null hypothesis is clearly rejected

Old Benchmark Problems

- Archetti et al. also solve seven deterministic problems
- These problems resemble scenario 1 or scenario 2 in that demands are small relative to vehicle capacity
- We compare Tabu with MIP + PP on the next page
- MIP + PP clearly outperforms Tabu

Computational Comparison

# nodes	MIP + PP	Time	TABU	TABU RT	% Savings Over TABU
51	5,246,111	1.8	5,335,535	13.2	1.68
76	8,401,822	4.0	8,495,410	35.8	1.10
101	8,195,575	3.7	8,356,191	57.6	1.92
121	10,419,937	10.0	10,698,369	389.0	2.60
151	13,073,981	18.1	13,428,515	386.4	2.64
200	10,431,774	5.6	10,560,148	38.4	1.22
101	8,229,911	5.8	8,253,184	49.0	0.28

Parameters in Endpoint MIP

- Endpoint MIP has two parameters
  - the number of neighbor endpoints
  - a time limit
- We set these parameters in order to approximately equalize the running times of Tabu and MIP + PP

Other Issues

- There are few SDVRP benchmark problems in the literature
- We've created an additional 21 benchmark problems ranging in size from 8 to 288 nodes
- Using symmetry and geometry, we can visualize a near-optimal solution for each of these 21 problems

Computational Results on New Benchmark Problems

Distance

A	B	n	Time (sec)	Imp. Over CW	% above Near-Optimal	Vehicles Saved
4	2	8	0.7	4.88	0	2
4	4	16	54.4	10.70	0.86	3
4	10	40	503.2	16.43	1.02	8
4	12	48	504.1	17.63	1.42	10
4	20	80	507.1	19.29	2.11	17
8	2	16	67.3	10.28	0	4
8	4	32	502.7	12.62	0.53	7
8	10	80	508.3	16.85	0.51	16
8	12	96	514.5	17.48	1.85	20
8	20	160	548.6	20.64	0.40	32
12	2	24	500.0	12.35	0	6
12	4	48	504.3	10.55	5.01	11

Computational Results on New Benchmark Problems

Distance

A	B	n	Time (sec)	Imp. Over CW	% above Near-Optimal	Vehicles Saved
12	10	120	521.7	16.49	0.94	25
12	12	144	542.3	18.41	0.80	29
12	20	240	563.0	19.82	1.43	44
16	2	32	508.3	13.42	0	8
16	4	64	500.0	13.80	2.74	14
16	10	160	541.6	17.00	1.58	35
16	12	192	554.2	17.43	2.06	40
72	2	144	514.7	20.16	2.00	34
72	4	288	651.0	20.20	1.96	65

Conclusions and Future Work

- MIP + PP seems to work well
- It consistently outperforms Tabu
- The approach of using MIP to improve a feasible solution could be applied to other combinatorial optimization problems
- The robustness of MIP + PP needs to be tested more comprehensively
- Parameter values need to be determined automatically
- It would be nice to know if the visualized “near-optimal” solutions are optimal