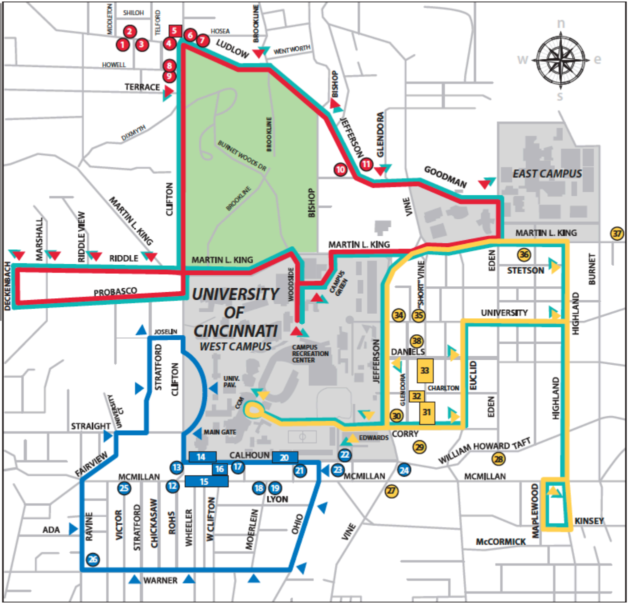
The Demand Weighted Vehicle Routing Problem

Jeffrey D.Camm, Saravanan Kuppusamy, Michael J.Magazine
University of Cincinnati
Kipp Martin
University of Chicago

November 14, 2011

Outline

Motivation


UNIVERSITY OF CINCINNATI
WEST CAMPUS
EAST CAMPUS
LUDLOW
GOODMAN
MARTIN L. KING
CLIFTON
BISHOP
JEFFERSON
GLENDORA
VINE
RIDDLE
PROBASCO
TERRACE
HOWELL
SHILOH
HOSEA
WENTWORTH
BROOKLINE
BURNET WOODS DR
DIXMYTH
MARSHALL
RIDDLE VIEW
DECKENBACH
WOODSIDE
CAMPUS GREEN
CAMPUS RECREATION CENTER
UNIV. PAV.
MAIN GATE
CALHOUN
MCMILLAN
LYON
STRATFORD
JOSELIN
STRAIGHT
FAIRVIEW
ADA
RAVINE
VICTOR
CHICKASAW
ROHS
WHEELER
W CLIFTON
MOERLEIN
OHIO
WARNER
EDWARDS
CORRY
DANIELS
CHARLTON
EUCLID
EDEN
STETSON
UNIVERSITY
HIGHLAND
BURNET
WILLIAM HOWARD TAFT
MAPLEWOOD
KINSEY
McCORMICK
"SHORT" VINE

Motivation

We consider a specific application and then describe the more generic problem.

1. Study current Bearcat Transportation System
   - North Route – day
   - East Route – day
   - Southwest Route – day
2. Make recommendations:
   - determine an assignment of stops/pickups to a route
   - determine the order of the stops/pickups – i.e. the bus route

Motivation

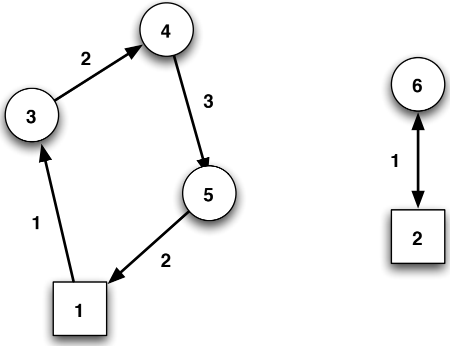
Criteria or metrics for route evaluation:

1. Students left waiting
2. Length of route
   - shorter routes imply shorter ride times for students
   - shorter routes also imply more frequent pickups and therefore less wait time at stops
   - *shorter routes can reduce students left waiting due to demand smoothing*
3. Even better – **demand-weighted length of route** – longer routes ideally have fewer students

Motivation

Route Evaluation – Example 1

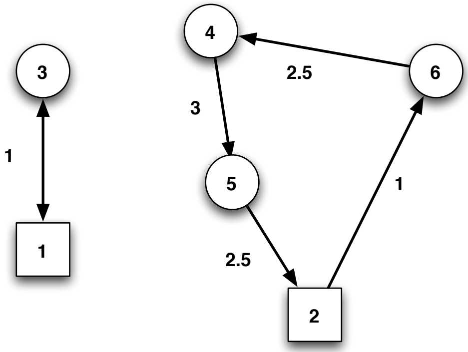
- two routes with hubs at nodes 1 and 2
- nodes 3, 4, 5, and 6 have student demands of 7, 1, 1, and 1, respectively



Motivation

Route Evaluation – Example 2

- two routes with hubs at nodes 1 and 2
- nodes 3, 4, 5, and 6 have student demands of 7, 1, 1, and 1, respectively



Motivation

	Solution 1		
	Route 1	Route 2	Total
Miles	8	2	10
Number of Students	9	1	10
Distance $\times$ Demand	72	2	74

	Solution 2		
	Route 1	Route 2	Total
Miles	2	9	11
Number of Students	7	3	10
Distance $\times$ Demand	14	27	41

In the real Bearcat case we can shorten both *total distance* and *demand-weighted distance.*

Generic Model

For any path $i_1, i_2, \ldots, i_m$, let

$$\Lambda(i_1, i_2, \ldots, i_m) = c_{(i_1,i_2)} + \cdots + c_{(i_{m-1},i_m)}$$

denote the length of the path and

$$\Omega(i_1, i_2, \ldots, i_m) = d_{i_1} + \cdots + d_{i_m}$$

the total demand associated with the nodes on the path.

The path is a *simple path* if it is a path and no nodes are repeated. A *simple cycle* is a simple path except that the first and last node are identical.

Generic Model

The Demand Weighted Vehicle Routing Problem (DWVRP) is to find:

1. A simple cycle (route) $k, i_{k_1}, i_{k_2}, \ldots, k$, for each hub $k = 1, \ldots, K$ such that every demand node $\{K+1, \ldots, n\}$ is assigned to exactly one cycle, and
2. $$\sum_{k=1}^{K} \Lambda(k, i_{k_1}, i_{k_2}, \ldots, k)\Omega(k, i_{k_1}, i_{k_2}, \ldots, k) \quad (1)$$

   is minimized. We assume that at least one node from the set $\{K+1, \ldots, n\}$ is assigned to each route, i.e. there are no empty routes.

Generic Model

Variables:

x_{kij} is 1 if arc (i,j) is assigned to the hub k tour, 0 if not
z_{ki} is 1 if node i is assigned to the hub k tour, 0 if not

Assign each node to a route:

$$\sum_{k \in H} z_{ki} = 1, \; \forall i \in N$$
$$z_{kk} = 1, \; \forall k \in H$$

Standard assignment constraints:

$$\sum_{j \neq i} x_{kij} = z_{ki}, \; \forall i \in N, \; \forall k \in H$$
$$\sum_{j \neq i} x_{kji} = z_{ki}, \; \forall i \in N, \; \forall k \in H$$

Cannot exceed vehicle capacity:

$$\sum_{i} d_{ik} z_{ki} \leq Q_k, \; \forall k \in H$$

Generic Model

Tour breaking constraints:

$$X_T = \{x_{kij} \mid \sum_{\substack{i \in M \\ j \in M}} x_{kij} \quad \leq \quad |M| - 1,\ \forall M \subset \overline{H},\ \ k \in H\} \tag{2}$$

Minimize demanded weighted distance:

$$\min \sum_{k \in H} \left((\sum_{i \in \overline{H}} d_i z_{ki})(\sum_{(i,j)} c_{ij} x_{kij}) \right) \tag{3}$$

Constraints are linear, but objective function has terms $d_i c_{ij} z_{ki} x_{ij}$.

We have a **nonconvex quadratic integer programming problem.**

Solution Approaches

1. Solve with a nonlinear integer programming code such as COIN-OR **Bonmin** and not worry about local optima.

2. Quadratic convex reformulation techniques. Hammer and Rubin (1970) – adding the term $M(x_i^2 - x_i)$ for sufficiently large M makes the model convex. Billionnet, Elloumi, and Plateau (2009) have improved on the continuous relaxation but require solving a semidefinite optimization problem. Use COIN-OR **Csdp**.

3. Global optimization techniques. Could use the COIN-OR **Couenne**. The problem is that the relaxation used gives very bad lower bounds.

4. Give a linear programming reformulation in **auxiliary variables.**

Solution Approaches – Use Auxiliary Variables

Key Idea: Apply dynamic programming algorithm (Christofides, Mingozzi, and Toth (1981)) to an easier, approximate problem.

In this relaxation we still minimize the demand-weighted distance.

$$\sum_{k=1}^{K} \Lambda(k, i_{k_1}, i_{k_2}, \ldots, k)\Omega(k, i_{k_1}, i_{k_2}, \ldots, k) \tag{4}$$

However, we relax requirements that 1) every node be assigned to a route, and that 2) the routes are simple cycles. Instead we minimize (4) subject to the constraint

$$\sum_{k=1}^{K} \Omega(k, i_{k_1}, i_{k_2}, \ldots, k) = \sum_{k=K+1}^{n} d_k \tag{5}$$

which says that the sum of the demands on each route must equal the total demand in the system.

Solution Approaches – Use Auxiliary Variables

Rather than actually solve the dynamic program, do the following:

- Write a linear program that is "equivalent" to the dynamic programming Bellman equations.
- Vertices of this linear program correspond to solutions of the relaxed dynamic programming problem.
- "Link" the dynamic programming variables with the x_{kij} variables through a linear transformation ($x = Tz$)
- Add the assignment constraints

We end up with a linear integer program that solves the original problem. Not only that, but the linear relaxation is very tight!

Solution Approaches

We started with:

$$
\text{Model 1} \qquad \begin{aligned} \min f(x) & \\ Ax &\geq b \\ x &\geq 0 \end{aligned}
$$

and by using dynamic programming, ended up with a reformulaiton:

$$
\text{Model 2} \qquad \begin{aligned} \min c^\top z & \\ Bx + Dz &\geq d \\ x, z &\geq 0 \end{aligned}
$$

How do we compare the quality of the formulations?

Solution Approaches

1. First try to compare over a common variable space. Define

$$P_x = \{x \mid Ax \geq b,\ x \geq 0\}.$$

P_x is the feasible region of Model 1. Next define

$$Q_x = \{x \mid \exists\, z \ni z \geq 0,\ Bx + Dz \geq d,\ x \geq 0\}$$

Q_x is the **projection** of the feasible region of Model 2 into the variable space of Model 1.

$$P_x \ ?\ Q_x$$

2. How do we compare objective functions values? If there exists $(\overline{x}, \overline{z})$ such that $\overline{x} \in P_x \cap Q_x$ what is

$$f(\overline{x}) \ ?\ c^\top \overline{z}$$

Solving the Auxiliary Variable Model

Here is the model we want to solve using COIN-OR software:

$$\text{Model 2} \qquad \begin{aligned} \min c^\top z & \\ Bx + Dz &\geq d \\ x, z &\geq 0 \end{aligned}$$

Direct Approach: Use COIN-OR **Cbc** or **SYMPHONY**. Problem – millions of variables, constraints, and nonzeros. Could not even solve the LP relaxation.

Column Generation Approach: Try COIN-OR **Dip** – there is a problem, Dip wants to apply column generation to

$$z_{IP} = \min\{c^\top x \mid A'x \geq b',\ A''x \geq b'',\ x \in \mathbb{Z}^n\}$$

and solve the related problem z_D, defined by

$$z_D = \min\{c^\top x \mid A'x \geq b',\ x \in \mathcal{P},\ x \in \mathbb{R}^n\}.$$

$$\mathcal{P} = \mathrm{conv}(\{x \in \mathbb{Z}^n \mid A'x \geq b'\})$$

Solving the Auxiliary Variable Model

c_{kp} = demand-weighted cost of cycle p (distance times demand) on route k

n_{kpi} = number of times cycle p on route k is incident to node i

P_k = number of cylces from hub k to node k

$$
\begin{aligned}
& \min \sum_{k=1}^{K} \sum_{p=1}^{P_k} c_{kp}\theta_{kp} \\
FM(\theta) \qquad & \sum_{k=1}^{K} \sum_{p=1}^{P_k} n_{kpi}\theta_{kp} = 1, \quad i = K+1, \ldots, n \\
& \sum_{p=1}^{P_k} \theta_{kp} = 1, \quad k = 1, \ldots, K \\
& \theta_{kp} \in \{0,1\}, \quad k = 1, \ldots, K, \quad p = 1, \ldots, P_k
\end{aligned}
$$

Solving the Auxiliary Variable Model

Summary of COIN-OR software usage:

- Use **Cbc** to solve to find an initial feasible solution by solving a generalized assignment problem
- Use **Clp** to solve the restricted master each time a new column enters (problem with using a warm start with **Osi**)
- Use **Cbc** to solve the restricted master for an integer solution to give an upper bound
- Use **Clp** (use **ClpNetworkMatrix**) for the separation problem that finds violated subtour constraints
- Use **Clp** for the separation problem that finds a cut in a new class of cuts identified for this problem

Branch and bound – home grown, hope to incorporate within COIN-OR **Alps**. The relaxations are so tight very little enumeration required.

Computational Results

Table: Solution Results For 2009 Student Demand Data

Problem	LP Value	IP Value	Integrality Gap (%)	Number Nodes	Number Columns	Total CPU Seconds
y2009p40-25-25-25	96.62	96.62	0.00	0	628	17.28
y2009p50-25-25-25	116.37	116.51	0.12	2	980	30.53
y2009p60-25-25-25	143.93	144.28	0.24	2	1042	37.20
y2009p70-25-25-25	185.75	185.75	0.00	0	620	7.38
y2009pmean-25-25-25	195.73	195.73	0.00	0	469	3.42
y2009p95-140-140-140	373.75	373.75	0.00	2	1109	1251.28

Computational Results

Table: Solution Results for 2010 Student Demand Data

Problem	LP Value	IP Value	Integrality Gap (%)	Number Nodes	Number Columns	Total CPU Seconds
y2010p75-40-40-40	206.93	206.93	0.00	0	653	59.72
y2010p80-40-40-40	238.26	238.39	0.05	2	818	54.92
y2010p85-40-40-40	270.68	270.68	0.00	0	536	17.97
y2010p90-40-40-40	328.73	333.43	1.43	18	1871	25.51
y2010p90-40-40-60	318.71	333.43	4.62	144	8340	499.39
y2010p90-40-60-40	318.28	320.22	0.61	6	1443	76.71
y2010p90-60-40-40	313.54	313.54	0.00	0	444	19.64
y2010p92-40-40-60	353.22	370.32	4.84	124	6772	248.86
y2010p92-40-60-40	348.07	364.49	4.72	150	9938	396.86
y2010p92-60-40-40	338.03	338.03	0.00	0	453	9.07
y2010p90-60-20-20-40	290.08	290.87	0.27	4	759	44.98
y2010p95-150-150-150	385.51	388.57	0.79	4	1069	1378.60

Implementation

Input parameters:

- distances – pretty easy, use Google Maps
- student demand – what is the demand at each node? Let me explain:

    1. y2009p70-25-25-25 – year 2009 data, 70th percentile nonzero demand, bus capacity of 25 on each route
    2. y2010p95-150-150-150 – year 2010 data, 95th percentile nonzero demand, bus capacity of 150 of each route

  Note: there are no 150 seat buses (80 seats is the max), the idea is solve the unconstrained problem.

Implementation

Model Recommendation *Evaluation using distance metric.*

	Baseline	Recommendation
East	2.7 miles	2.5 miles
North	4.3 miles	3.3 miles
Southwest	2.3 miles	2.3 miles
Total	9.3 miles	8.1 miles

Savings: 12.9%

Implementation

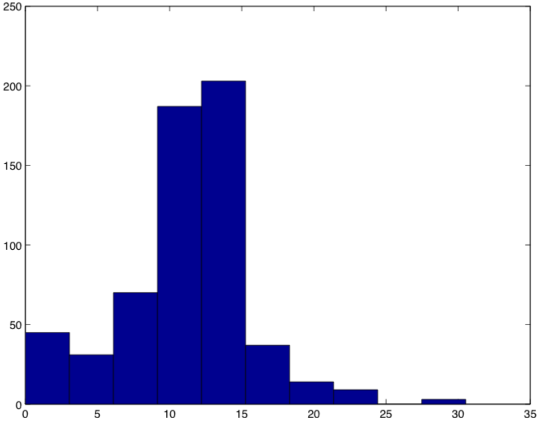
Model Recommendation *Evaluation using demand-weighted distance metric.*

Problem: What is demand? The number of students at stop varies throughout the day. Here are demand-weighted distances based on three percentiles (November, 2010 data).

Demand	Baseline	Recommendation
75th percentile	234 student miles	207 student miles
80th percentile	269 student miles	238 student miles
85th percentile	306 student miles	271 student miles

Implementation

Model Recommendation *Evaluation using demand-weighted distance metric.* Here is a comparison of demand-weighted distances based on 599 November, 2010 scenarios.



Implementation

Model Recommendation *Evaluation using demand-weighted distance metric.*

- 419/599 scenarios result in a demand-weighted decrease of 10% or more
- 275/599 scenarios result in a demand-weighted decrease of 12% or more
- 75/599 scenarios result in a demand-weighted decrease of 15% or more

Note: the modification *never does worse*!

Implementation

Model Recommendation *Evaluation based on students left waiting.* Here is a comparison of demand-weighted distances based on November, 2010 scenarios.

Case 1: We run all 599 demand scenarios with a bus capacity of 60 on North, 40 on East, and 40 on Southwest. *Both solutions have identical results!*

In two of the 599 scenarios students are left waiting. In scenario 308, two students left behind on North route, in scenario 341, five students left behind on Southwest route.

Implementation

Model Recommendation *Evaluation based on students left waiting.* Here is a comparison of demand-weighted distances based on November, 2010 scenarios.

Case 2: We run all 599 demand scenarios with a bus capacity of 60 on North, 20 on East, and 40 on Southwest. *Both solutions have identical results except for one scenario*.

Scenario	Baseline	Recommendation
8	3	3
10	1	1
14	14	14
15	14	15
44	3	3
243	1	1
346	1	1

Implementation

Some real problems:

- one-way streets (actually not a problem with Google maps)
- **narrow streets**
- left turns with no light or stop sign
- turnaround problem

Duplication of Results

All of the code and data sets (2009, 2010) are available at

`https://projects.coin-or.org/svn/OS/trunk/OS/applications/columnGen`

See also:

`https://projects.coin-or.org/OS`

Optimization Services project has solver interfaces for COIN-OR solvers **Bonmin**, **Clp**, **Cbc**, **Couenne**, **DyLP**, **Ipopt**, and **SYMPHONY**.

The paper is available at:

`http://faculty.chicagobooth.edu/kipp.martin/root/bearcat.pdf`

Formulation Comparison

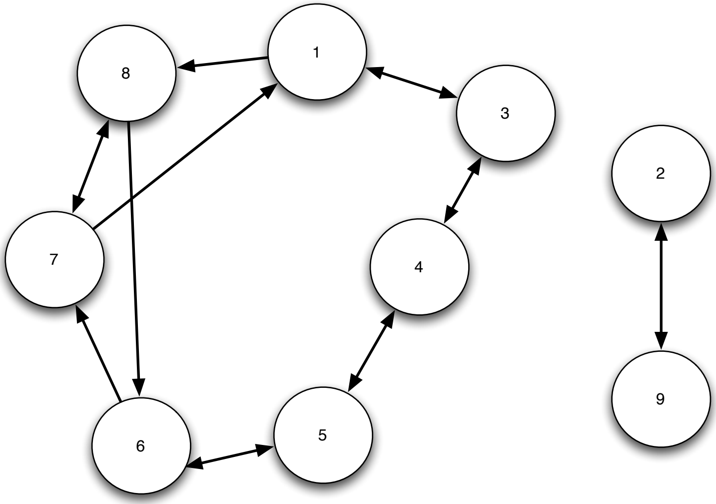


Figure: The nine city problem fractional solution illustrating $Q_x \subset P_x$.

Formulation Comparison

- The fractional solution must be the convex combination of extreme point solutions that correspond to route two cycles of length one and a route one cycle of length six so that the sum of demands is exactly seven.
- There is no other possibility because on route two there can be no cycles less than length one or greater than length one (in an extreme point solution, the hub node is never be visited more than once)
- If the route two solution has a cycle of length one then the route one cycle must be length six in every extreme point solution. However, in the figure there are no paths of length six that begin with arc (1, 3) and return to the origin.