Bearcat Transportation System Design using COIN-OR

SARAVANAN KUPPUSAMY
MICHAEL MAGAZINE
College of Business, University of Cincinnati

KIPP MARTIN
Booth School of Business, University of Chicago

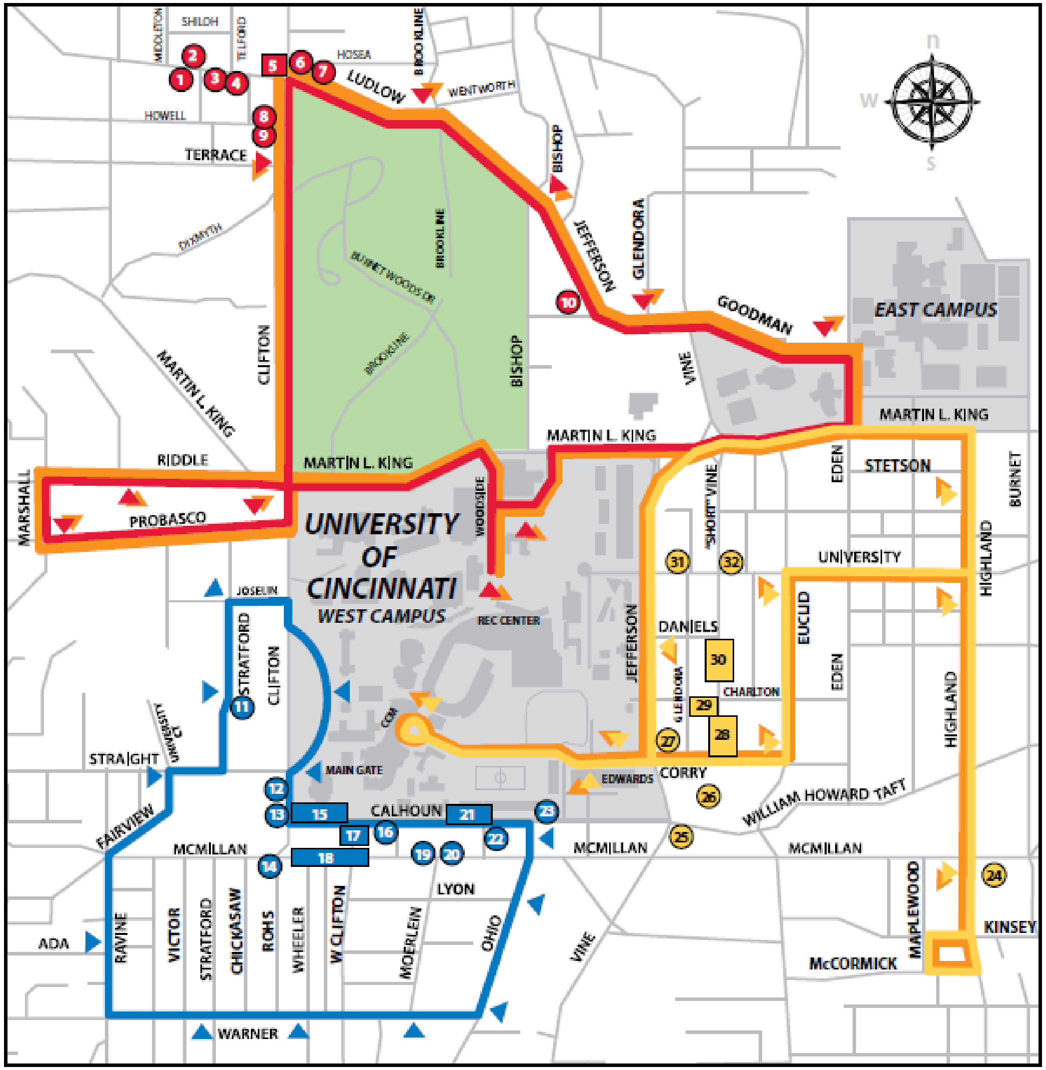
AGENDA

- PROBLEM INTRODUCTION
- BEARCAT TRANSPORTATION SYSTEM
- MODELS
- GAMS / COIN-OR CBC IMPLEMENTATION
- INITIAL RESULT
- RESTRICT THE ROUTE DISTANCE
- WHY BCP?
- BCP PROCEDURE USING COIN-OR
- COMPARISON OF GAMS & BCP PROCEDURES
- A VARIATION IN BCP PROCEDURE
- FUTURE WORK

PROBLEM INTRODUCTION

- Bearcat Transportation System (BTS) of University of Cincinnati (UC) is a student-led bus system for transporting students, faculty and staff around campus and off-campus locations.
- Student population has been changing by number and geography
- Both the student executives and university officials are interested in re-designing transportation system.

Bearcat Transportation System – Current Map



NORTH ROUTE (DAY) Route Stop Location	3.7 Miles
SOUTHWEST ROUTE (DAY & EVENING) Route Stop Location	2.1 Miles
EAST ROUTE (DAY) Route Stop Location	2.8 Miles

Total Miles = 8.6 Miles

Bearcat Transportation Models

- Bearcat Transportation Problem has two model components:
  - A Master Problem that has no sub tour elimination constraints.
  - A separation problem that is used to identify the sub tours.

- To have sub tour elimination constraints in the Master problem increases the complexity and thereby computational time.
- The sub problem (separation model) is used to identify sub tours and generate violated cuts on the fly. It is more efficient and improves the computational time.

Master Problem

- $N = Number\ of\ locations$
- $M = Number\ of\ routes.$
- $H = Number\ of\ hub\ locations.$

Minimize

$$\sum_{k}^{M}\sum_{i}^{N}\sum_{j<>i}^{N} D_{ij} * X_{ijk}$$

s.t

$$\sum_{k=1}^{M} Z_{ik} \geq 1 \forall i \ldots (1)$$

Each location *i* must be covered by at least by a route

$$\sum_{\substack{j=1 \\ i<>j}}^{N} X_{ijk} = Z_{ik} \forall i,k \ldots (2)$$

Only one location can be reached from the location *i* on route *k*

$$\sum_{\substack{j=1 \\ i<>j}}^{N} X_{jik} = Z_{ik} \forall i,k \ldots (3)$$

Location *i* on route *k* can be reached from only one location

$$\sum_{i=1}^{H} Z_{ik} \geq 1 \forall k \ldots (4)$$

At least one hub location must be covered by each route *k*

$$Z_{ik}, X_{ijk} \in \{0,1\}$$

Sub Problem

$$Maximize$$
$$\sum_{e \in E} \overline{X}_e \alpha_e^k - \sum_{\substack{i=1 \\ i \neq k}} \theta_i^k$$
$$s.t$$
$$\alpha_e^k \leq \theta_i^k \leq 1, e \in \delta(i), i \in V$$
$$\theta_i^k \geq 0$$

- $G = (V, E)$, *graph with vertex set* V *and undirected edges* E
- $\delta(i) =$ *set of edges adjacent to node i*

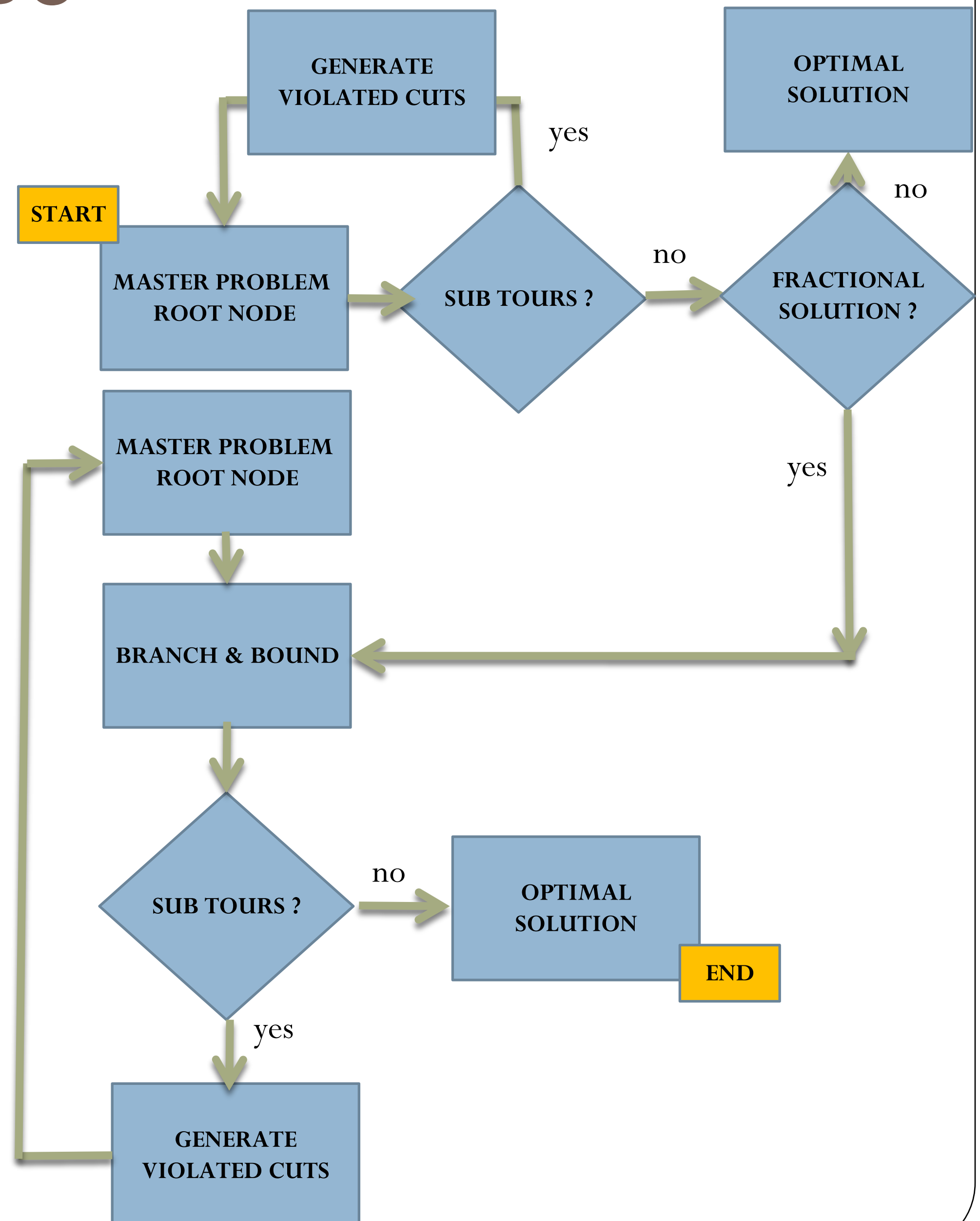
Violated Cut

$$\sum_{\substack{i \in S \\ j \in S \\ i <> j}} X_{ij} \leq Ord(S) - 1$$

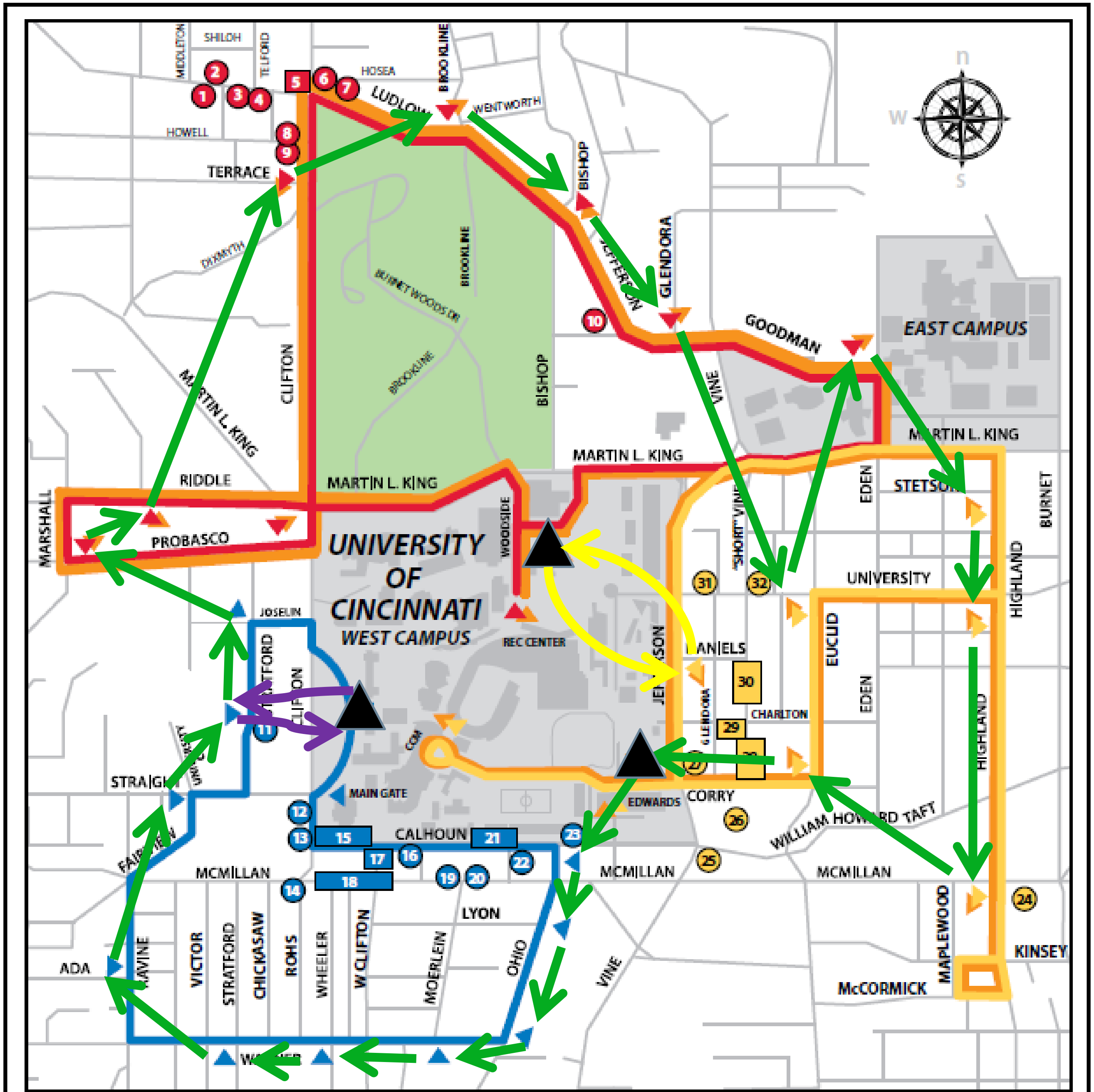
- $S =$ *set of locations for which* θ_i^k *is greater than zero*

GAMS /COIN-OR CBC IMPLEMENTATION

- Master Problem – An assignment Model
- Sub Problem – A Separation Model
- Used GAMS to code both the Master and Sub problems.



Map of an initial result



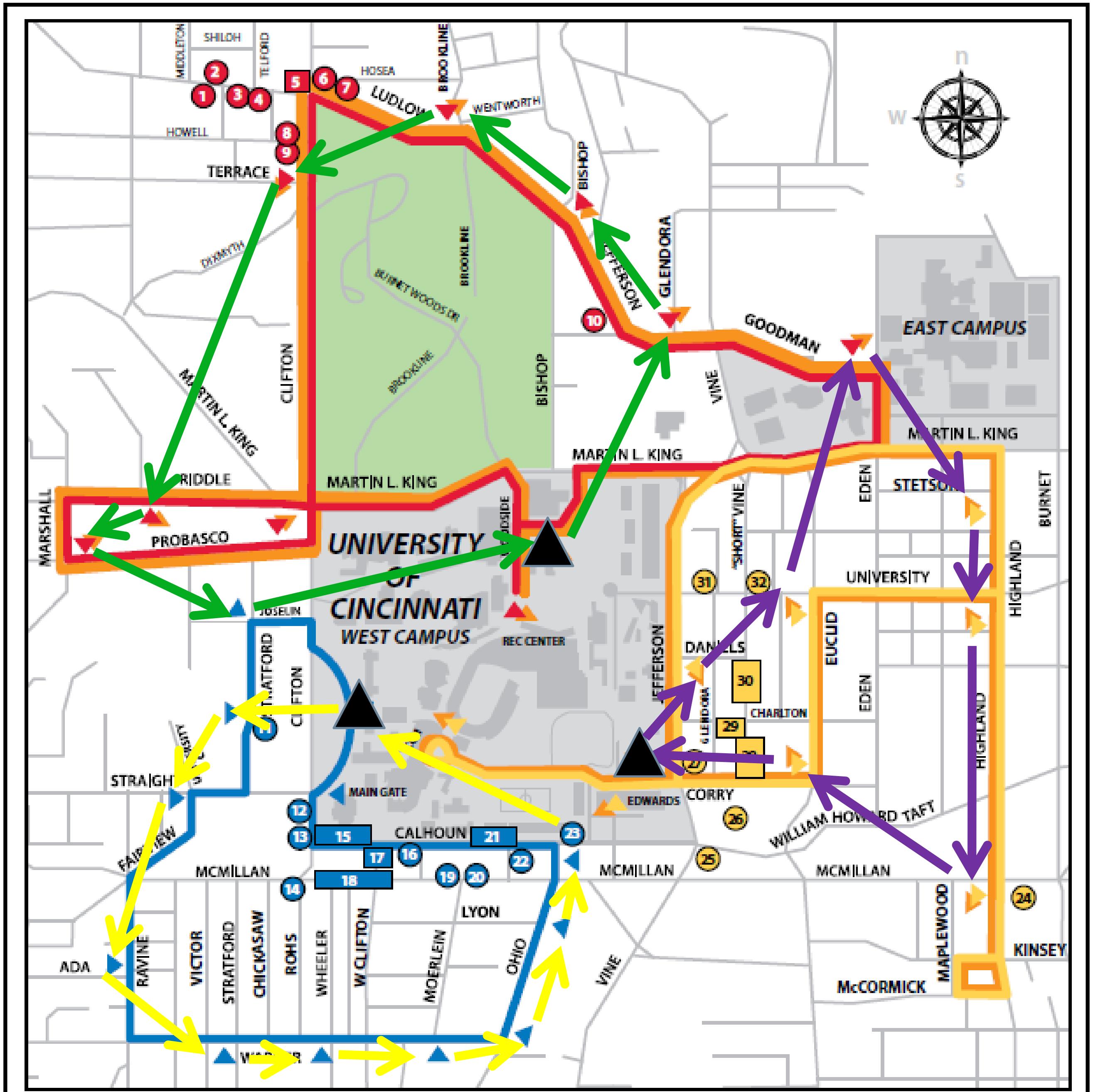
Route 1 = 0.4 Miles

Route 2 = 0.8 Miles

Route 3 = 6.0 Miles

Total Miles = **7.2 Miles**

- The total miles of the initial result is 7.2 miles which is 1.4 Miles (**16% reduction**) less than the baseline miles
- However, the routes from the initial result are **not implementable**



Restrict the route distance

- Restricted the distance covered by a route (maximum = 3.7 miles, arbitrarily chosen)

Route 1 = 2.5 Miles

Route 2 = 1.9 Miles

Route 3 = 3.2 Miles

Total Miles = **7.6 Miles**

Restrict the route distance

- The total distance from this scenario is 7.6 miles which is 1 mile (11.6%) less than the baseline total distance.

- Note, the ability to implement the routes has greatly improved.

- At least 30 trips are made during daytime. Given 1 mile reduction on a single trip, 30 miles can be saved on a daily basis.

- Assuming 200 days (excluding summer quarter & breaks) per year, 6000 miles (200x30) can be saved on an annual basis.

- Moreover, reduction in traveled distance can also increase the number of trips and decrease the waiting time at bus stations.

Why BCP (Branch Cut and Price)?

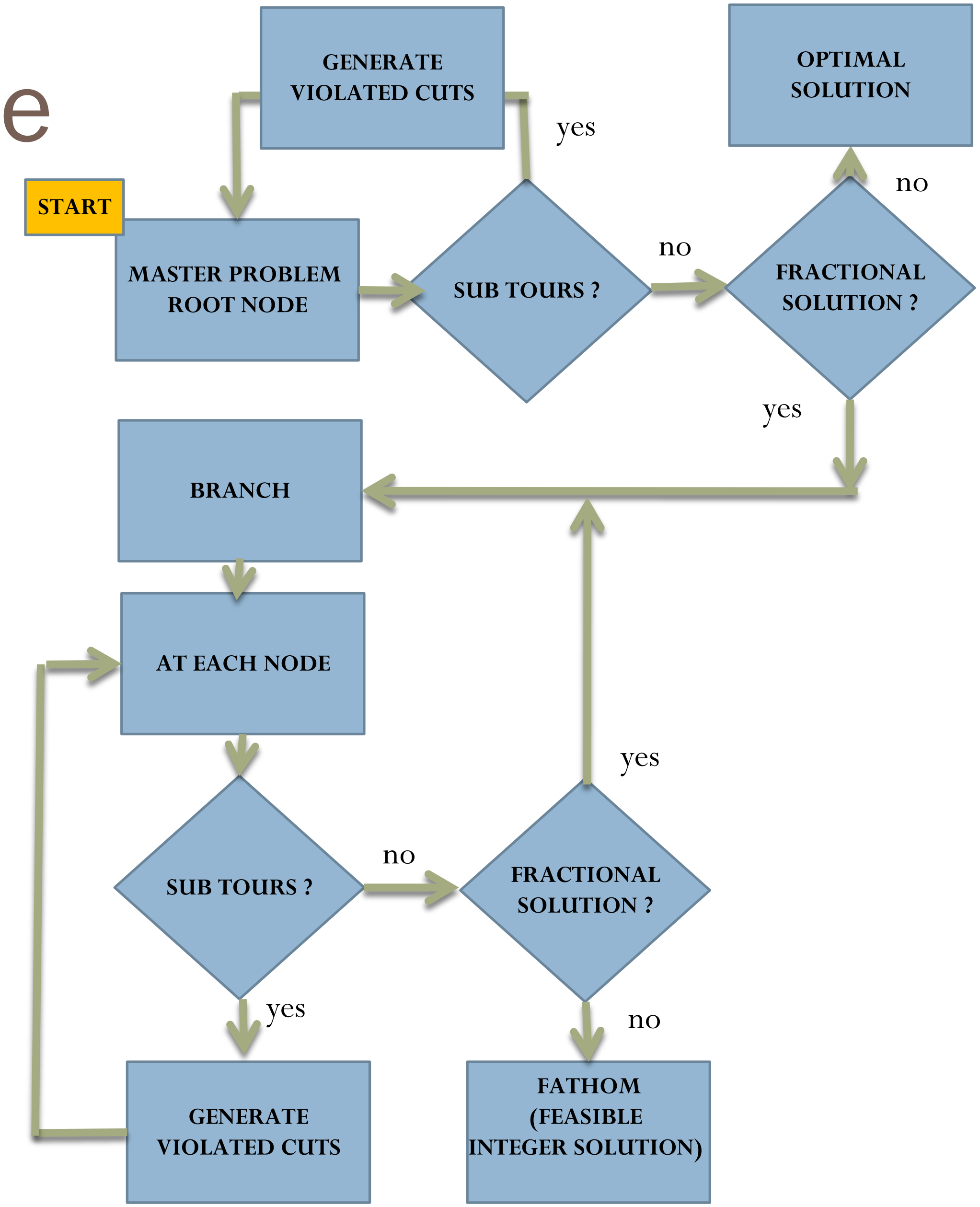
- Interested in looking at different objective functions like minimizing the variation in route distance, minimizing the maximum route distance etc.
- GAMS / COIN-OR CBC implementation takes lot of time to solve the problem with the above objective functions.
- Adding cuts at node level can be very effective in reducing the integrality gap and the amount of enumeration required in the branch & bound.
- GAMS does offer Branch-and-Cut-and-Heuristic Facility (BCH) which can be used to call separation model and add violated cuts at node level.

Why BCP?-continued

- However, at a node, BCH functionality calls the separation model only in the case of fractional solution. In the case of integer solution, BCH functionality never calls the separation model.
- We know an integer solution could have sub tours and identifying those sub tours is not possible with BCH functionality.
- The above issue can be solved by using BCP. At each node, BCP calls the separation model till no sub tours are identified.
- BCP is more effective than the GAMS/COIN-OR CBC implementation since cuts can be added at the node level in the branch and bound process. As a result, solve time in BCP should be less than that of the GAMS.

BCP- Procedure

- Violated cuts are generated at each node when sub tours are identified.
- A fractional solution with no sub tours is branched further.
- Once cuts are generated at a node , they are added to cut pool.
- The cuts in the cut pool are accessible to all nodes.



Test Problems

S.No	Test Problem
1	6LocR1
2	6LocR2
3	10LocR1
4	10LocR2
5	17LocR1*
6	17LocR2*
7	24LocR1*
8	24LocR2*
9	48LocR1*

- * are from the test library and the reference is given below
- the rest is randomly generated data
- 17LocR1- 17 City, one route problem (TSP)
- 17LocR2- 17 City, two route problem

- Reinelt, G., "TSPLIB-A Traveling Salesman Problem Library", ORSA Journal on Computing, Vol.3, No.4, Fall 1991

Comparing GAMS & BCP Implementations

		Solution		**Solve time** (seconds)		**Cuts Generated**	
		GAMS	BCP	GAMS	BCP	GAMS	BCP
1	6LocR1	68	68	4	1	3	3
2	6LocR2	80	80	4	1	2	2
3	10LocR1	118	118	4	1	0	0
4	10LocR2	131	131	6	1	2	2
5	17LocR1*	2085	2085	11	3	17	20
6	17LocR2*	2155	2155	225	1800	55	9311
7	24LocR1*	1272	1272	20	3	24	25
8	24LocR2*	1308	1312**	100	3600	29	5468
9	48LocR1*	5046	5063**	115	3600	104	712

Note:

** sub-optimal solution

A variation in BCP procedure

		Solution	**Solve time** (seconds)	**Cuts Generated**
		BCP	BCP	BCP
1	6LocR1	68	1	3
2	6LocR2	80	1	2
3	10LocR1	118	1	0
4	10LocR2	131	1	2
5	17LocR1*	2085	1	20
6	17LocR2*	2155	3600	41046
7	24LocR1*	1272	4	25
8	24LocR2*	1308	3600	28996
9	48LocR1*	5046	455	973

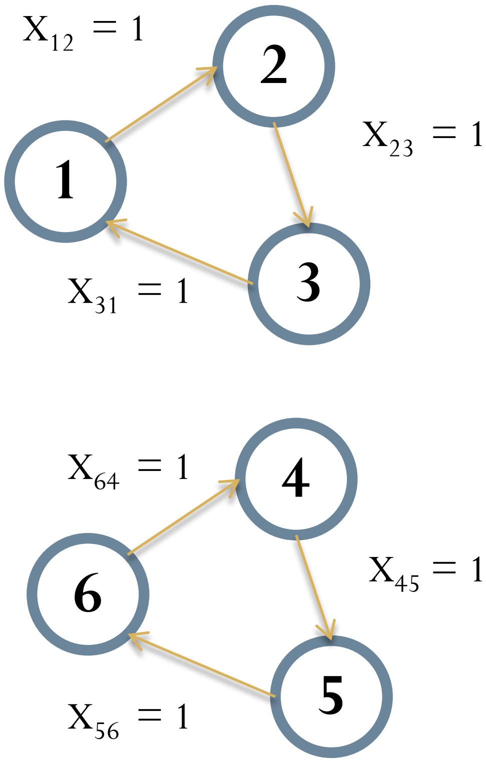
- At each node, the separation model is called when the solution is integer.
  In the case of fractional solution, the decision is to branch.
- The above variation is an attempt to simulate GAMS/COIN-OR implementation.

Future work

- It is clear that BCP’s computational time is worse than the GAMS and attempts are made to understand this counter intuitive result.
- Experiment on the variety of objective functions based on the user requirements and understand their impact on the routing solutions.
- Continue to work with University officials and student body to obtain feedback and to enhance the ability to implement the solution.

QUESTIONS??

Appendix A - Sub problem Example



- It is a 6 locations, 1 route problem (TSP)
- Hub location = 3
- k =4

Maximize

$$\alpha_{12}^{4}+\alpha_{45}^{4}+\alpha_{56}^{4}+\alpha_{64}^{4}-\{\theta_{1}^{4}+\theta_{2}^{4}+\theta_{3}^{4}+\theta_{5}^{4}+\theta_{6}^{4}\}$$

s.t

$$\alpha_{12}^{4}\le\theta_{1}^{4},\alpha_{12}^{4}\le\theta_{2}^{4},\alpha_{23}^{4}\le\theta_{2}^{4},\alpha_{23}^{4}\le\theta_{3}^{4},$$

$$\alpha_{31}^{4}\le\theta_{1}^{4},\alpha_{31}^{4}\le\theta_{3}^{4},\alpha_{64}^{4}\le\theta_{4}^{4},\alpha_{64}^{4}\le\theta_{6}^{4},$$

$$\alpha_{45}^{4}\le\theta_{4}^{4},\alpha_{45}^{4}\le\theta_{5}^{4},\alpha_{56}^{4}\le\theta_{5}^{4},\alpha_{56}^{4}\le\theta_{6}^{4};$$

$$\theta_{1}^{4}\le 1,\theta_{2}^{4}\le 1,\theta_{3}^{4}\le 1,\theta_{4}^{4}\le 1,\theta_{5}^{4}\le 1,\theta_{6}^{4}\le 1;$$

$$\theta_{1}^{4},\theta_{2}^{4},\theta_{3}^{4},\theta_{4}^{4},\theta_{5}^{4},\theta_{6}^{4}\ge 0;$$

Appendix A - Sub problem Example

- Sub problem will be solved for all the nodes ($k = 1,2,3,4,5,6$) . However, once a sub tour is identified and violated cut is generated for a node *k*, the cut is added to the cut pool and resolved again.

- Following is the optimal solution for the above problem:
  - $\alpha_{45} = \alpha_{56} = \alpha_{65} = 1, \alpha_{12} = \alpha_{23} = \alpha_{31} = 0;$
  - $\theta_4^4 = \theta_5^4 = \theta_6^4 = 1, \theta_1^4 = \theta_2^4 = \theta_3^4 = 0;$
  - Objective function value = 0+1+1+1-{0+0+0+1+1}= 1;
  - Since $\theta_4^4 = \theta_5^4 = \theta_6^4 = 1$, the indexes {4,5,6} are used to generate the following violated cut:
  - $x_{45} + x_{46} + x_{54} + x_{56} + x_{64} + x_{65} <= 2;$