The Colorful Traveling Salesman Problem

Yupei Xiong, *Goldman, Sachs & Co.*

Bruce Golden, *University of Maryland*

Edward Wasil, *American University*

Presented at 10th ICS Conference

Coral Gables, January 2007

Outline of Lecture

- Background: The MLST Problem
- Introduction to the CTSP Problem
- The CTSP is NP-complete
- A Simple Heuristic for the CTSP
- A GA for the CTSP
- Computational Results
- Conclusions and Open Questions

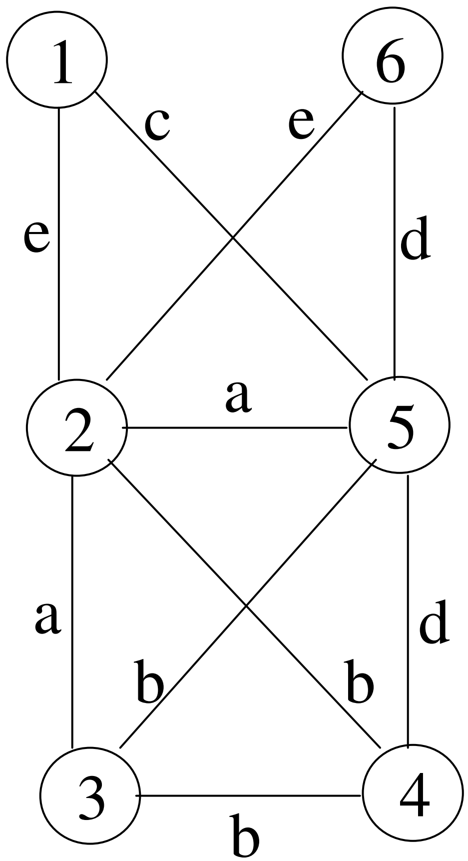
Background

- The Minimum Label Spanning Tree (MLST) Problem
  - Communications network design
  - Edges may be of different types of media (e.g., fiber optics, cable, microwave, telephone lines, etc.)
  - Each edge type is denoted by a unique letter or color
  - Construct a spanning tree that minimizes the number of colors

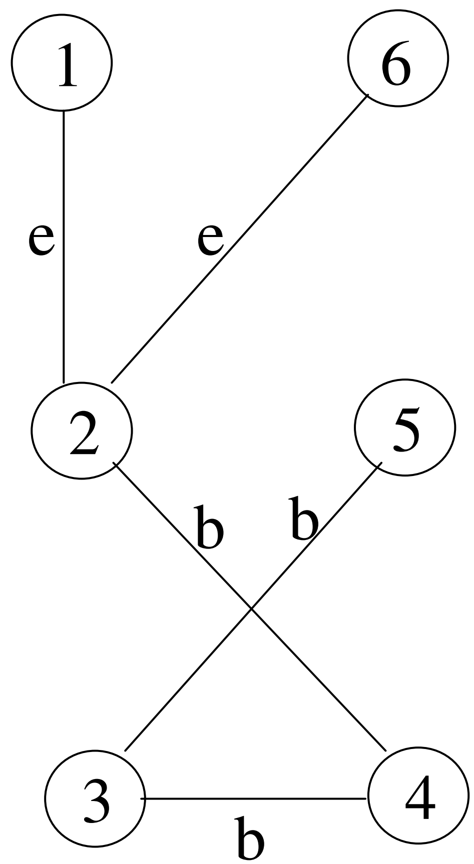
Background

- A Small Example

Input



Solution



Literature Review

- Proposed by Chang and Leu (1997)
- The MLST Problem is NP-complete
- Several simple heuristics have been proposed
- Some worst-case bounds have been obtained
- Effective metaheuristics have been proposed and tested
- See ORL (2005), IEEE TEC (2005), IEEE TEC (2006) for our work

Introduction to the CTSP

- Given an undirected complete graph with labeled edges
- Each edge has a single label
- Different edges can have the same label
- We think of each label as a unique color
- Find a Hamiltonian tour with the minimum number of colors
- A hypothetical scenario follows

Hypothetical Application

- A traveler wants to visit n cities and return home
- All pairs of cities are directly connected by railroad or bus
- There are l transport companies (colors)
- Each company controls a subset of the railroad and bus lines (edges)
- Each company charges the same flat monthly fee for using its lines

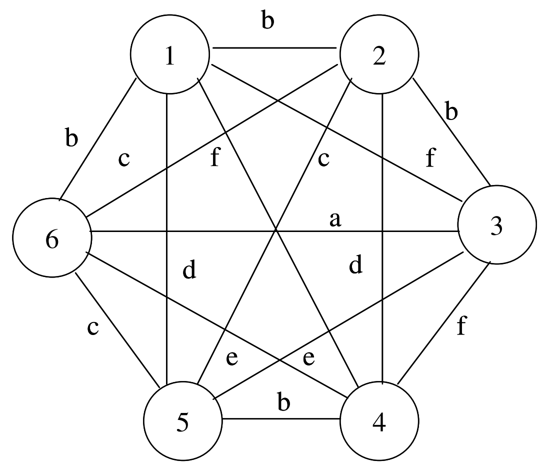
The CTSP is NP-complete

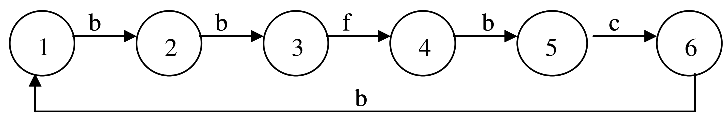
- Let HAM-CYCLE be the Hamiltonian tour problem
- HAM-CYCLE is NP-complete
- Let $G = (V, E)$ be an instance of HAM-CYCLE
- Construct an instance of CTSP
  - Form the complete graph $G' = (V, E')$ where $|E'| = \binom{n}{2}$
  - Each edge in E has label c
  - Each edge in $E' - E$ has a unique label

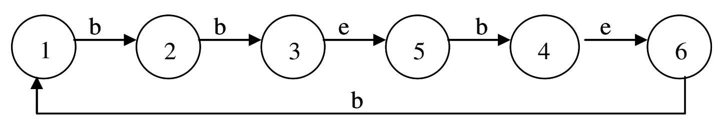
The CTSP is NP-complete

- Note that G has a Hamiltonian tour ⇔ G' has a tour with only one label
- So, if we could solve the CTSP efficiently, we could solve HAM-CYCLE efficiently
- Therefore, CTSP is NP-complete

An Example of the CTSP





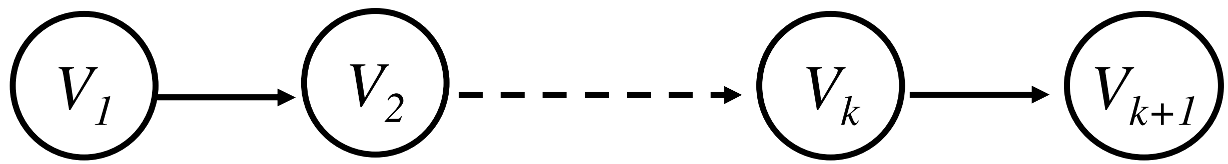


- Tour *h* has 3 labels and tour *g* has 2 labels

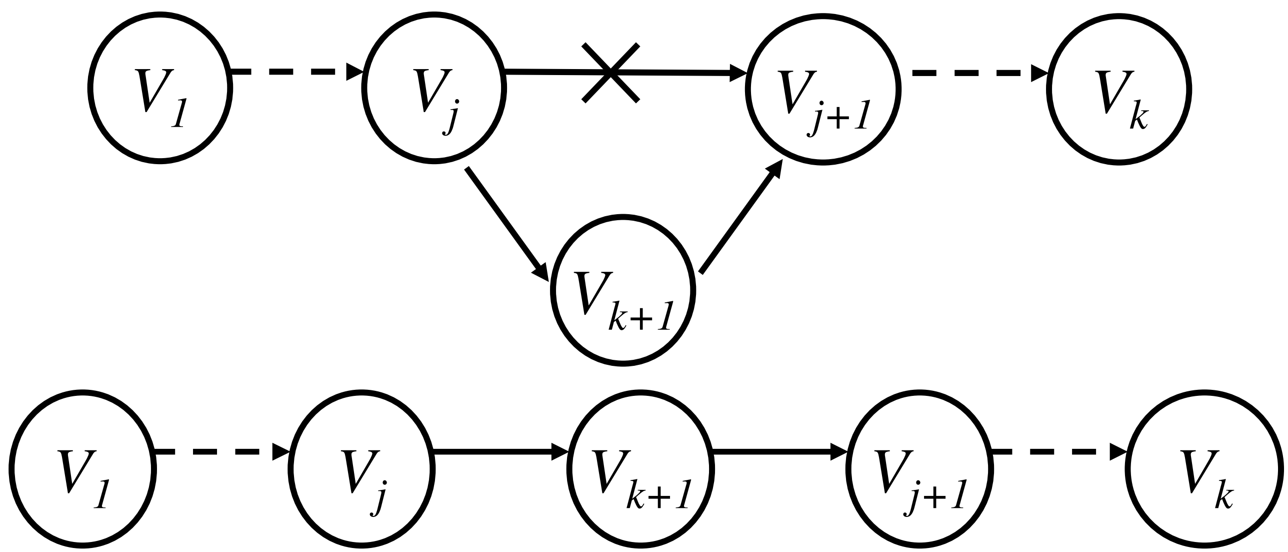
Maximum Path Extension Algorithm

- How to extend a partial tour h: $v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k$?

Case 0

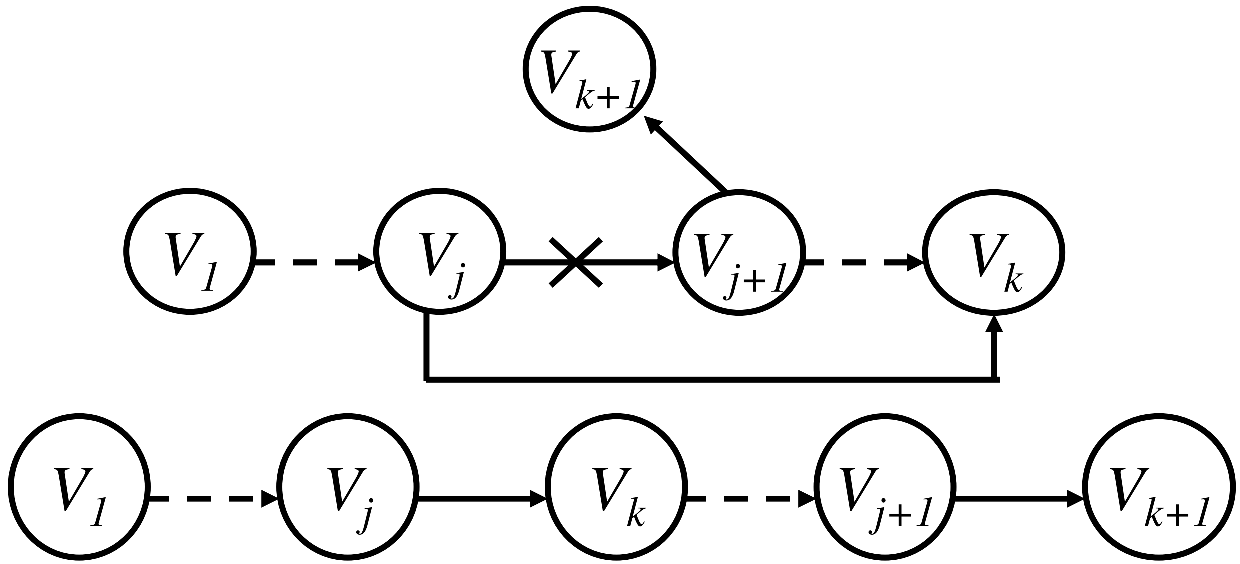


Case 1

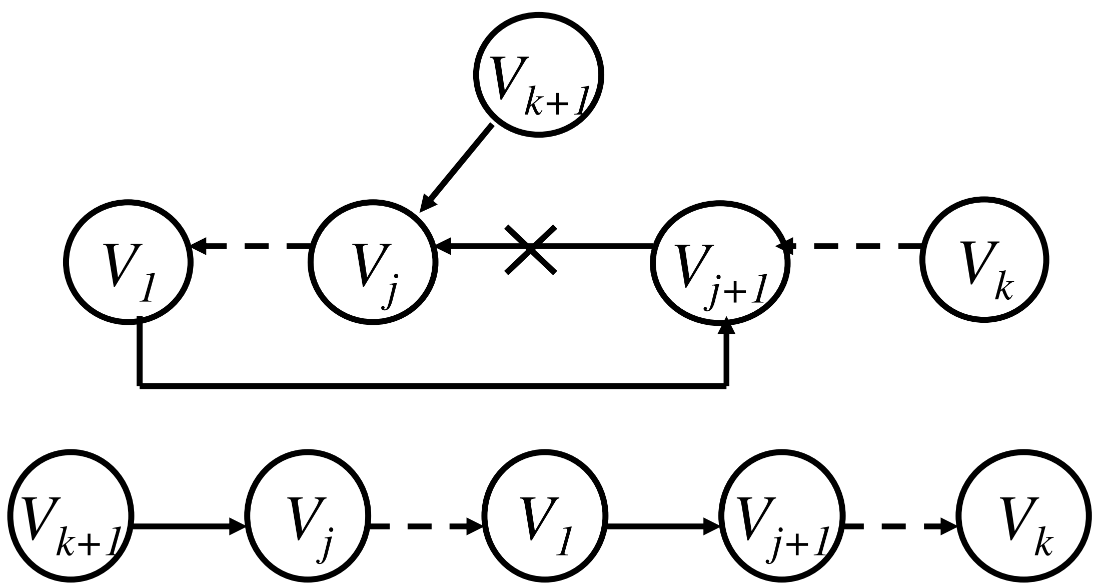


Maximum Path Extension Algorithm

Case 2

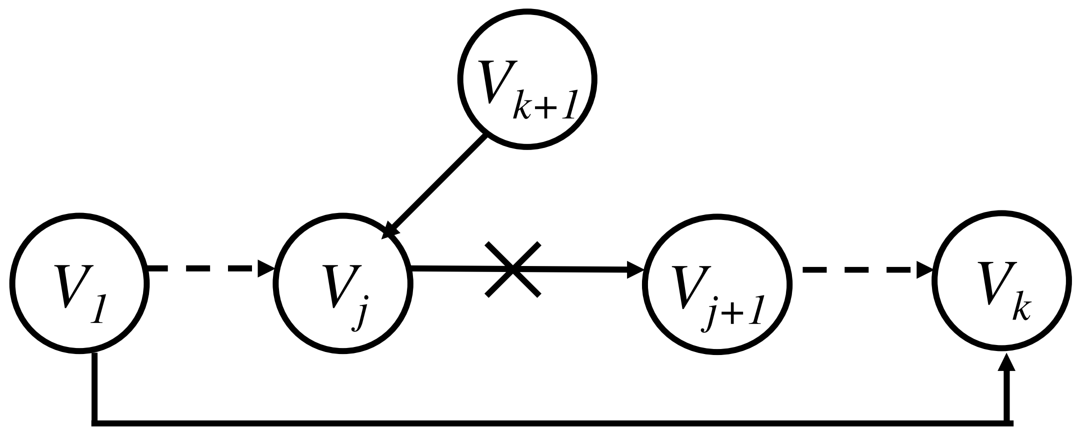


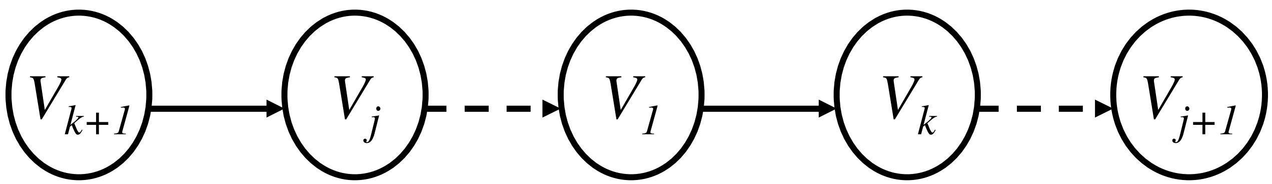
Case 3



Maximum Path Extension Algorithm

Case 4





Case 5

- If any unvisited node cannot satisfy the above cases, we extend the partial tour h by inserting an unvisited node v_{k+1} (edge(v_k, v_{k+1})) with the highest frequency label

MPEA in Detail

Step 1: Sort all the labels in G according to their frequencies, from largest to smallest.

Step 2: Randomly select $v_1 \in V$, then find $v_2 \in V$ such that the label c_{12} of the edge (v_1, v_2) has the highest frequency.

Step 3: Let $h = \{v_1, v_2\}$ and $C = \{c_{12}\}$.

Step 4: Add unvisited nodes to h according to the rules in Cases 0 to 5, until h contains all n nodes.

Step 5: Suppose $h = \{v_1, \ldots, v_n\}$ is an ordered sequence of nodes, and let c_{1n} denote the label of the edge (v_1, v_n). If c_{1n} is not in C, then add it to C.

Step 6: Output h.

MPEA and a GA

- The total running time for MPEA is O(n^3)
- Suppose we could begin with a label set *C* which contains more than one label
- Idea: use a GA to solve the MLST problem to obtain *C*
- The subgraph *H* induced by *C* is connected, spans all nodes in *G*, and has relatively few labels
- Finally, apply MPEA

Computational Experiment

- For each (n, l), we randomly generate 10 graphs
- For each graph, we run MPEA 200 times and find the best result
- We run the GA once and report the best result
- We output the average number of labels of the 10 graphs for each (n, l)
- The results are presented next

Computational Results for MPEA and GA

	MPEA	Avg. time (sec)	GA	Avg. time (sec)
$n = 50, l = 25$	2.4	0.1	2.4	0.3
$n = 50, l = 50$	4.5	0.1	4.2	0.4
$n = 50, l = 75$	5.6	0.1	5.7	0.5
$n = 50, l = 100$	6.6	0.1	6.8	0.6
$n = 100, l = 50$	3.5	0.3	3.0	0.9
$n = 100, l = 75$	4.0	0.3	4.1	1.2
$n = 100, l = 100$	5.8	0.2	5.1	1.5
$n = 100, l = 125$	6.3	0.3	6.9	1.7
$n = 100, l = 150$	7.2	0.3	6.9	1.7
$n = 150, l = 75$	3.4	0.7	3.0	5.1
$n = 150, l = 100$	4.5	0.8	4.1	6.2
$n = 150, l = 150$	5.9	0.9	5.5	7.6
$n = 150, l = 200$	7.5	0.9	7.3	8.9
$n = 200, l = 100$	3.8	1.5	3.4	10.1
$n = 200, l = 150$	5.5	1.9	4.9	13.0
$n = 200, l = 200$	6.9	1.9	6.2	14.7
$n = 200, l = 250$	8.2	2.0	7.4	17.2

- Run on a Pentium 4 PC with 1.80 GHz and 256 MB RAM

Conclusions

- The GA outperforms the MPEA in 12 cases
- The GA underperforms the MPEA in 4 cases, and one tie
- The GA yields better results and running time is very reasonable

Open Questions

- How good are the solutions?
- Recent paper by Cerulli et al. (August 2006)
  - their largest problems are the size of our smallest problems
  - tabu search procedure
  - much slower than our GA