Solution Progress

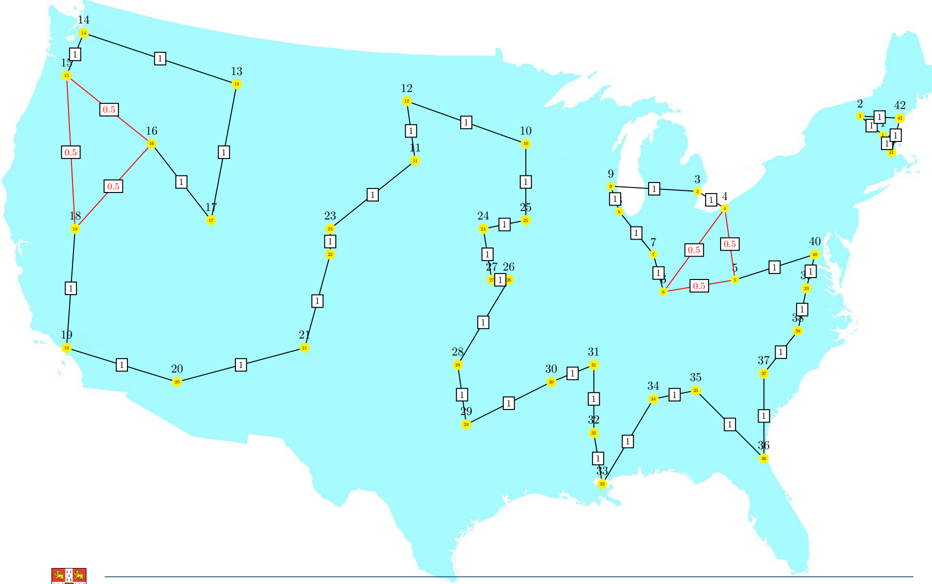
Andrei Ivašković Petar Veličković Thomas Sauerwald

Easter 2019

UNIVERSITY OF CAMBRIDGE

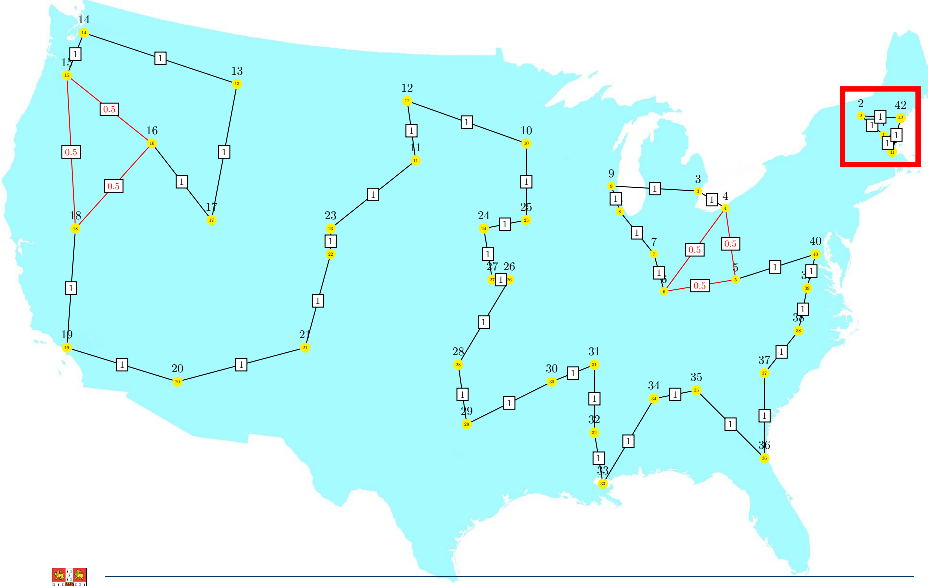
Iteration 1:

Objective value: -641.000000, 861 variables, 945 constraints, 1809 iterations



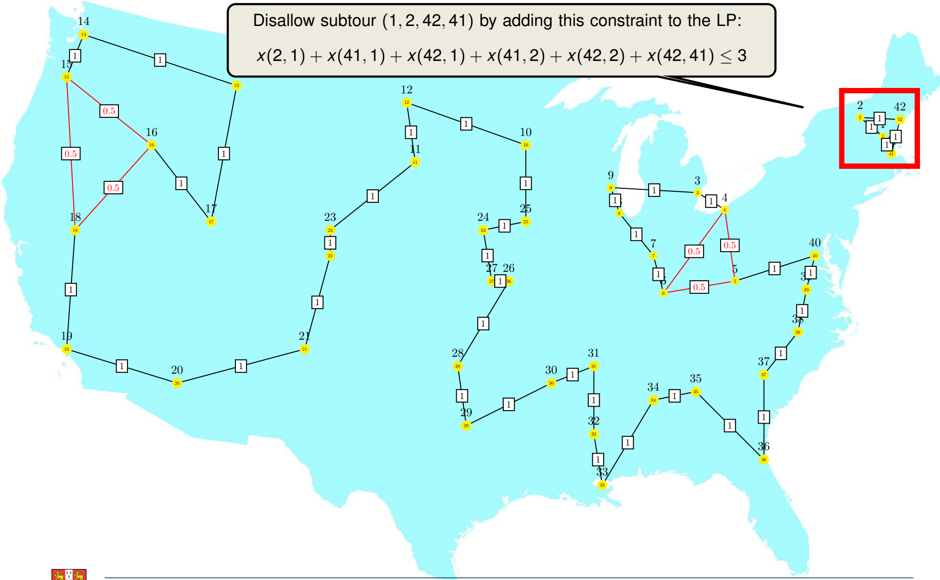
Iteration 1: Eliminate Subtour $1, 2, 41, 42$

Objective value: -641.000000, 861 variables, 945 constraints, 1809 iterations



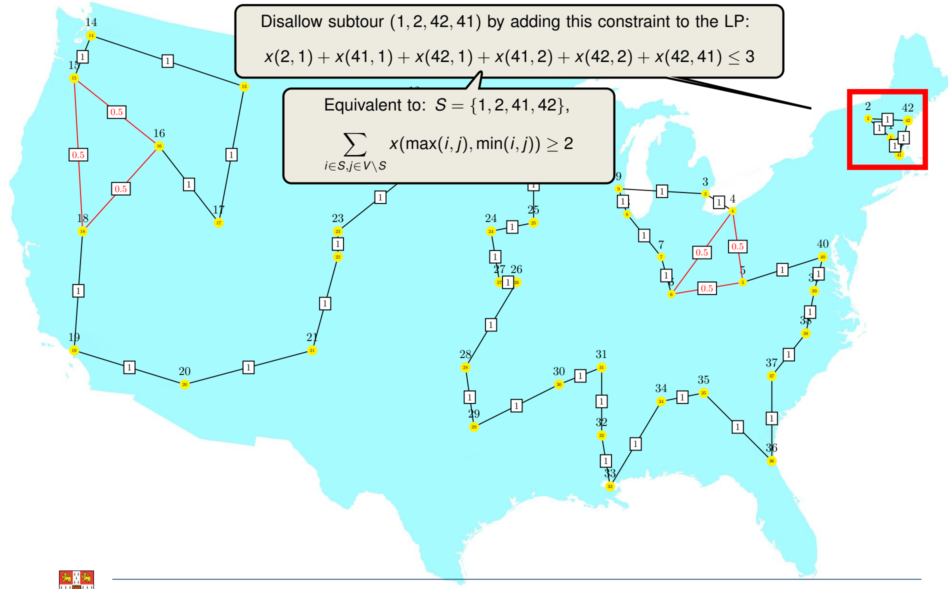
Iteration 1: Eliminate Subtour $1, 2, 41, 42$

Objective value: -641.000000, 861 variables, 945 constraints, 1809 iterations



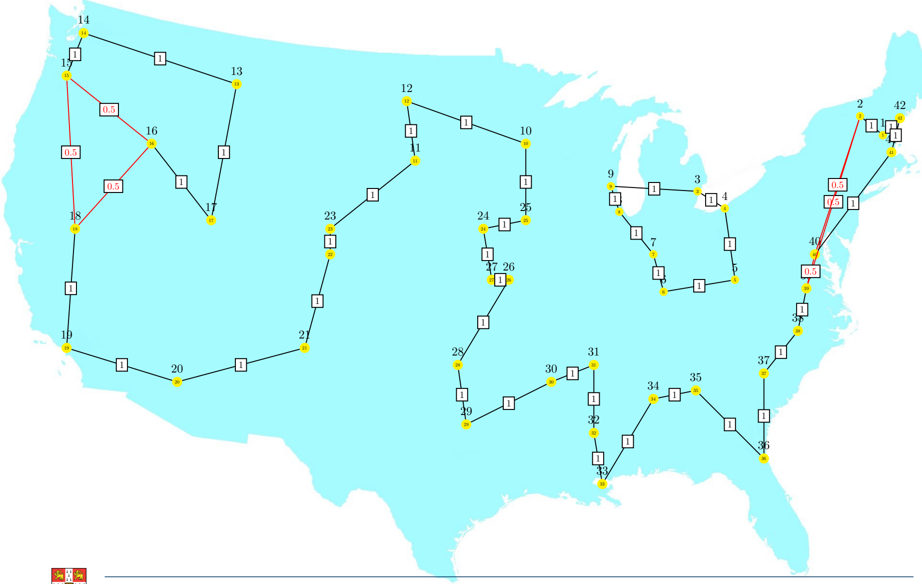
Iteration 1: Eliminate Subtour $1, 2, 41, 42$

Objective value: -641.000000, 861 variables, 945 constraints, 1809 iterations



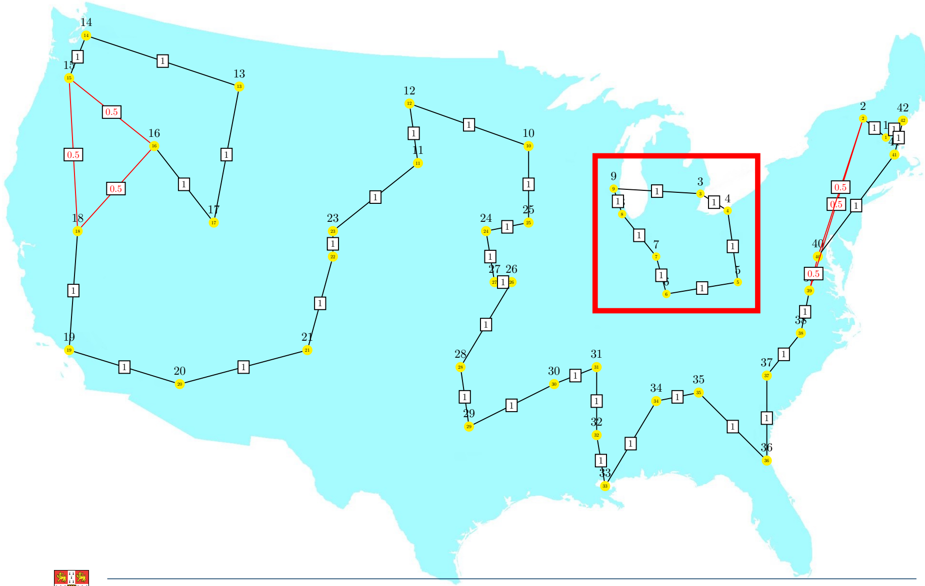
Iteration 2:

Objective value: −676.000000, 861 variables, 946 constraints, 1802 iterations



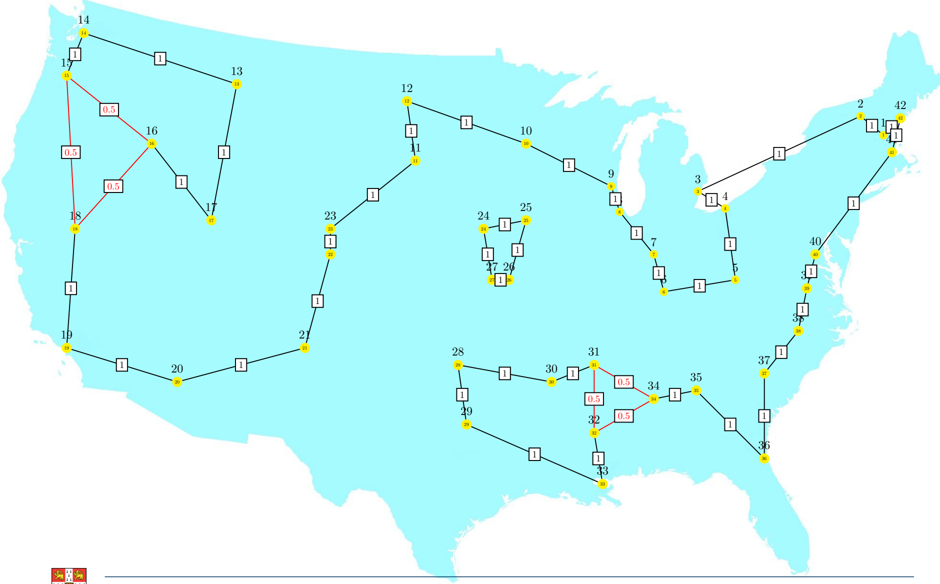
Iteration 2: Eliminate Subtour $3-9$

Objective value: -676.000000, 861 variables, 946 constraints, 1802 iterations



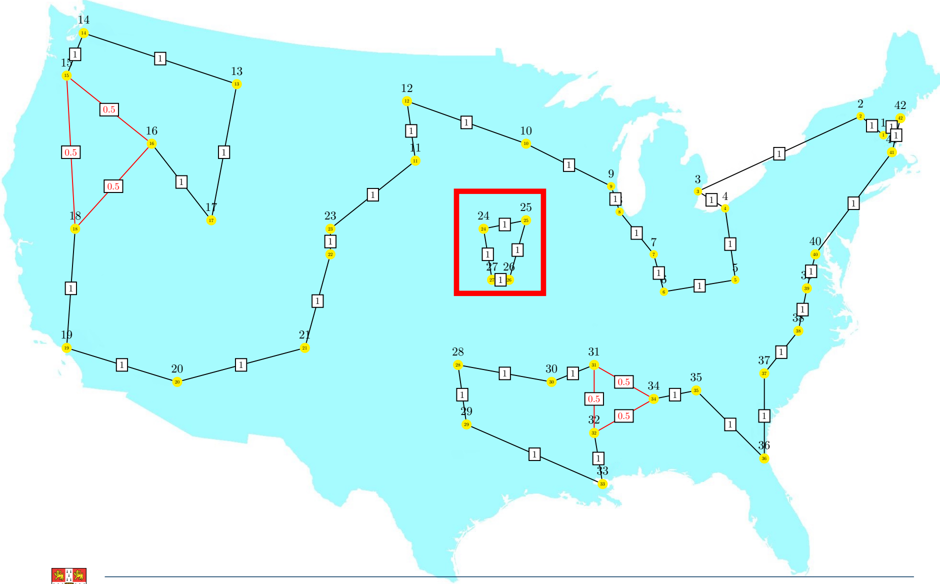
Iteration 3:

Objective value: -681.000000, 861 variables, 947 constraints, 1984 iterations



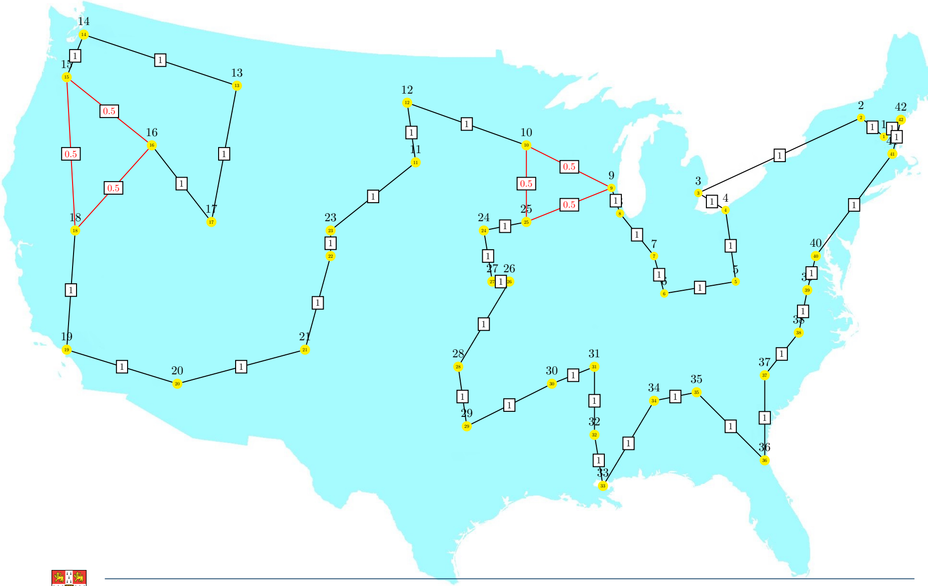
Iteration 3: Eliminate Subtour $24, 25, 26, 27$

Objective value: -681.000000, 861 variables, 947 constraints, 1984 iterations

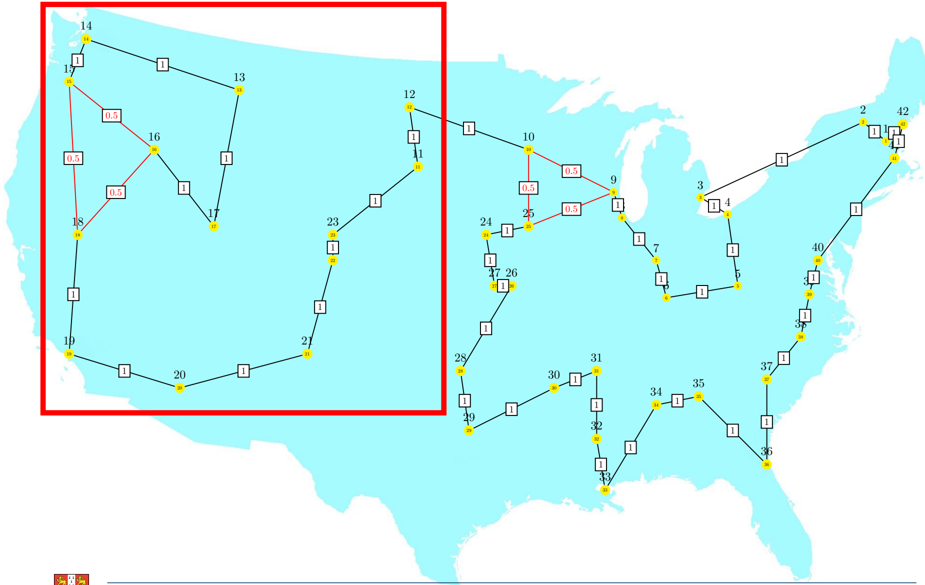


Iteration 4:

Objective value: -682.500000, 861 variables, 948 constraints, 1492 iterations

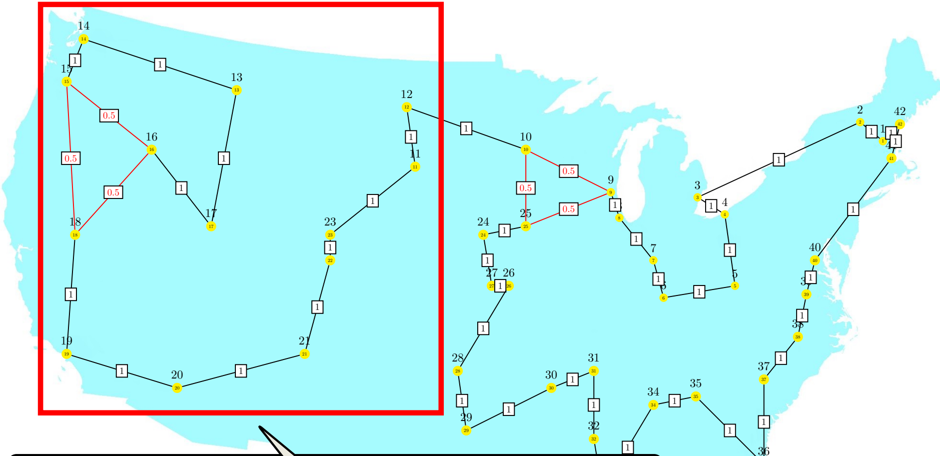


Objective value: -682.500000, 861 variables, 948 constraints, 1492 iterations



Iteration 4: Eliminate Cut $11 - 23$

Objective value: -682.500000, 861 variables, 948 constraints, 1492 iterations

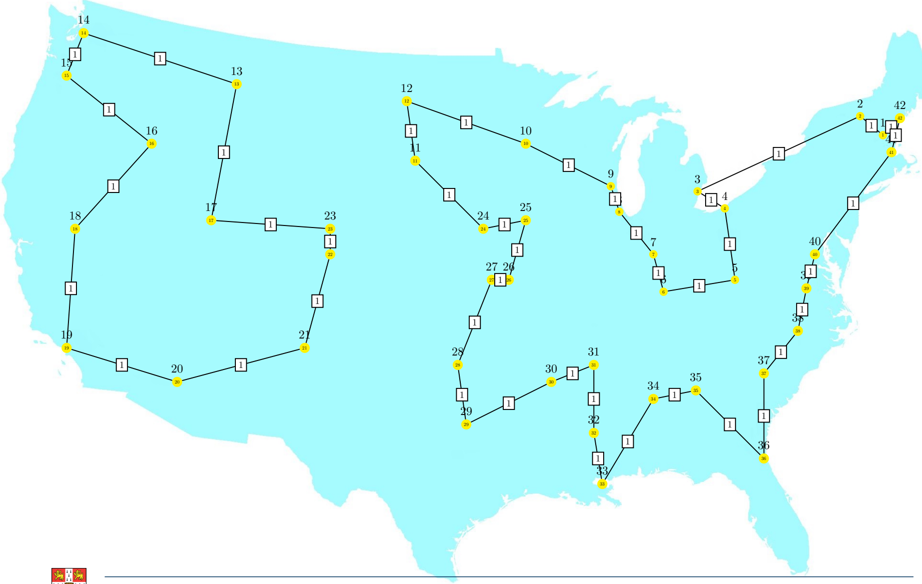


Tour has to include at least two edges between $S = \{11, 12, \ldots, 23\}$ and $V \setminus S$:

$$\sum_{i \in S, j \in V \setminus S} x(\max(i,j), \min(i,j)) \geq 2.$$

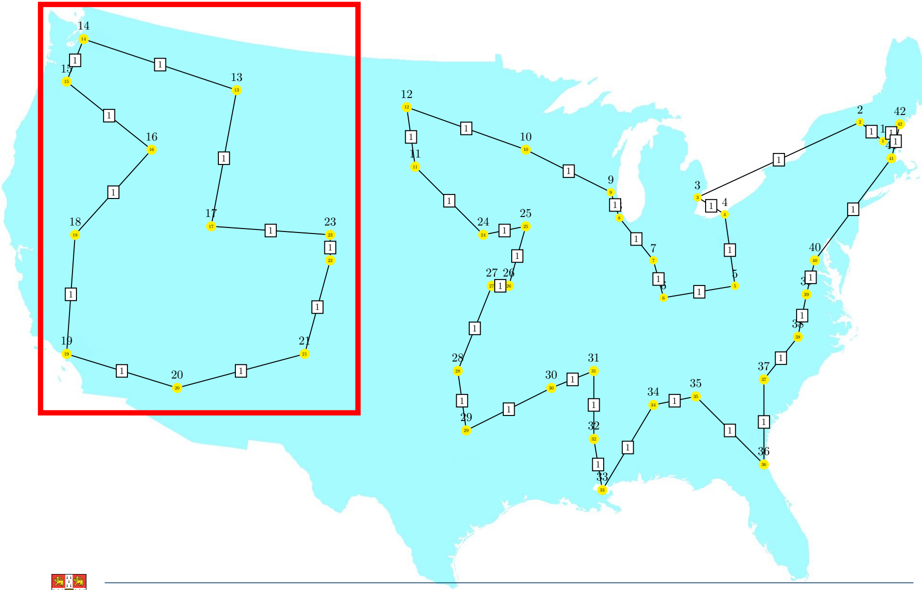
Iteration 5:

Objective value: −686.000000, 861 variables, 949 constraints, 2446 iterations



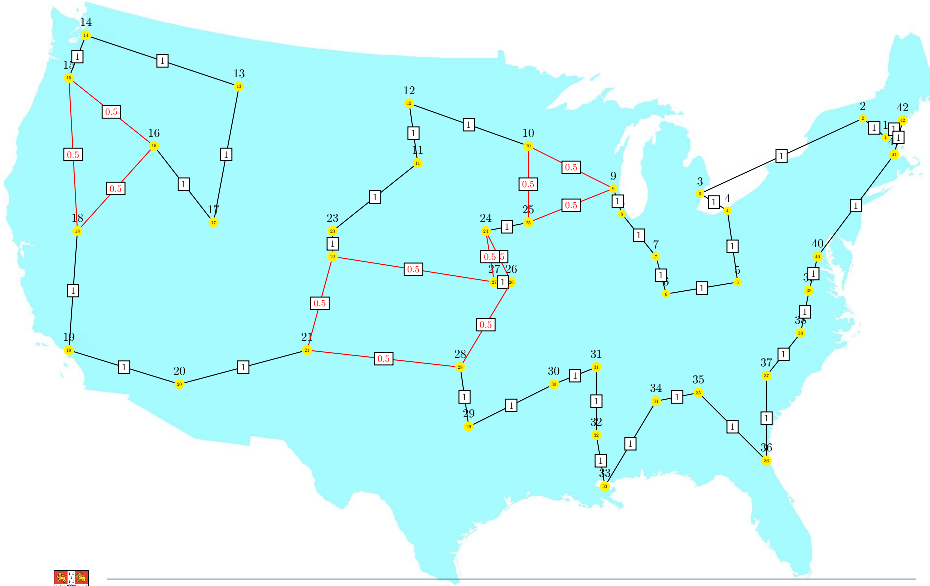
Iteration 5: Eliminate Subtour $13 - 23$

Objective value: -686.000000, 861 variables, 949 constraints, 2446 iterations



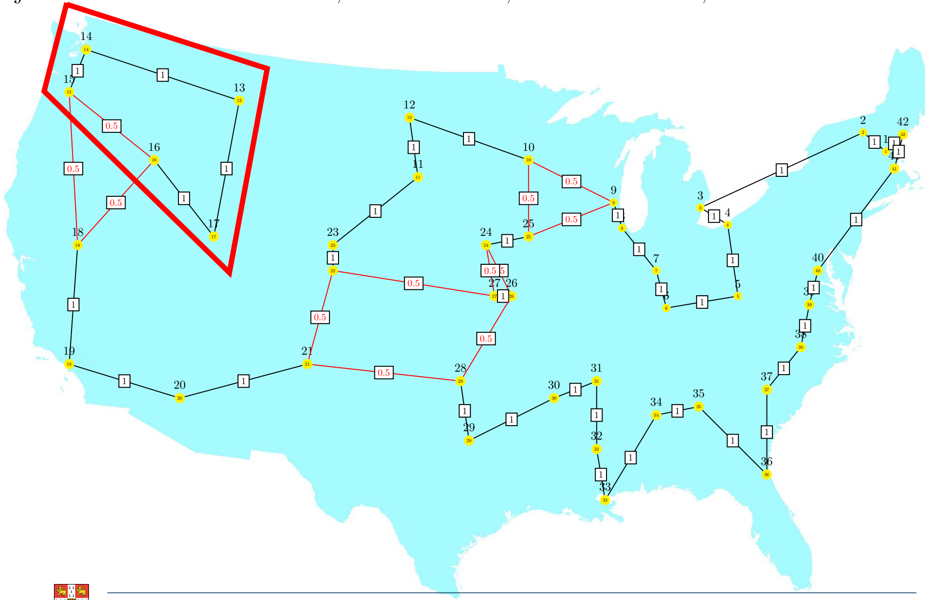
Iteration 6:

Objective value: −694.500000, 861 variables, 950 constraints, 1690 iterations



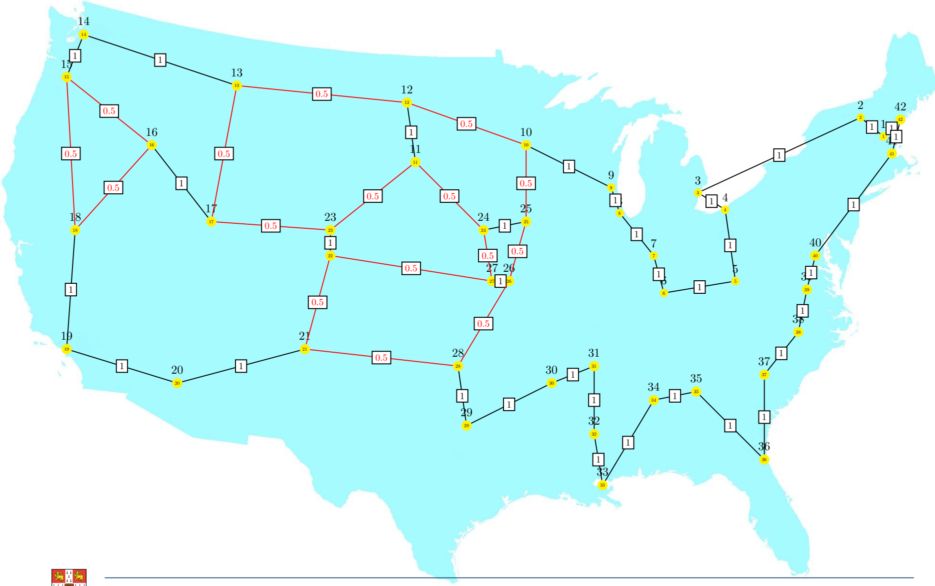
Iteration 6: Eliminate Cut 13 – 17

Objective value: −694.500000, 861 variables, 950 constraints, 1690 iterations



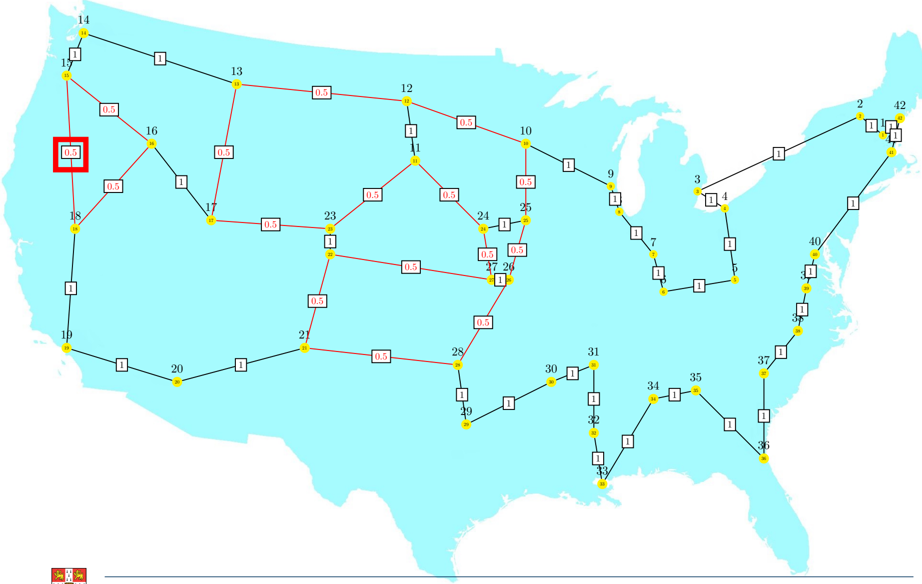
Iteration 7:

Objective value: −697.000000, 861 variables, 951 constraints, 2212 iterations



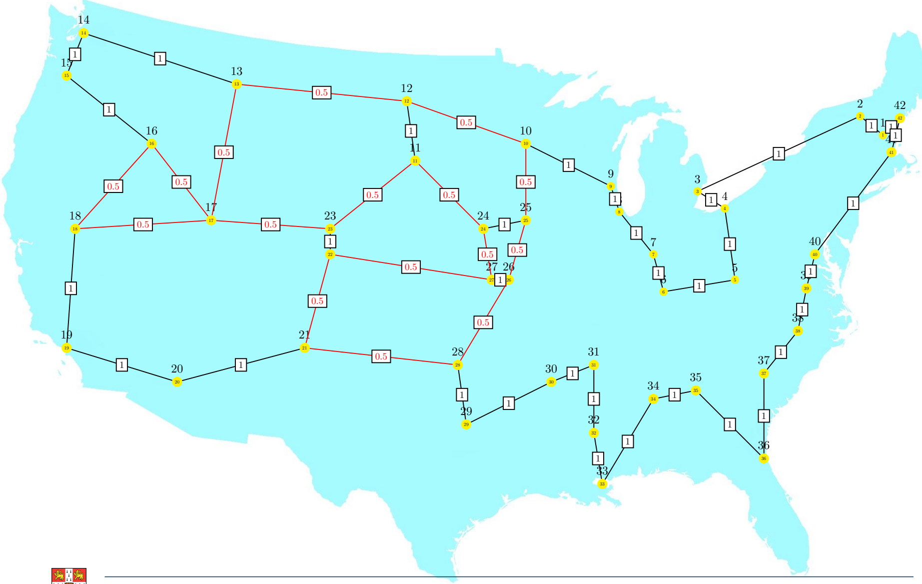
Iteration 7: Branch 1a $x_{18,15} = 0$

Objective value: -697.000000, 861 variables, 951 constraints, 2212 iterations



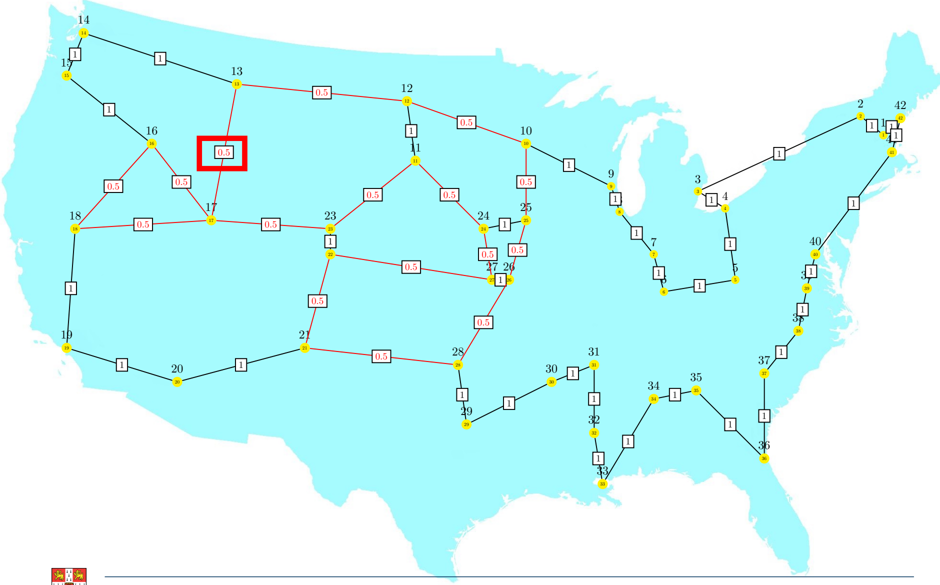
Iteration 8:

Objective value: -698.000000, 861 variables, 952 constraints, 1878 iterations



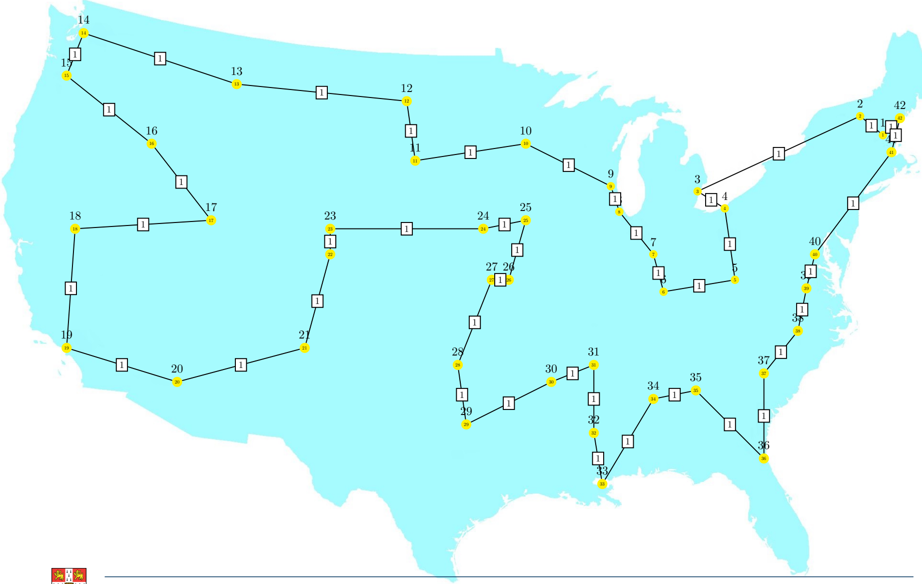
Iteration 8: Branch 2a $x_{17,13} = 0$

Objective value: -698.000000, 861 variables, 952 constraints, 1878 iterations



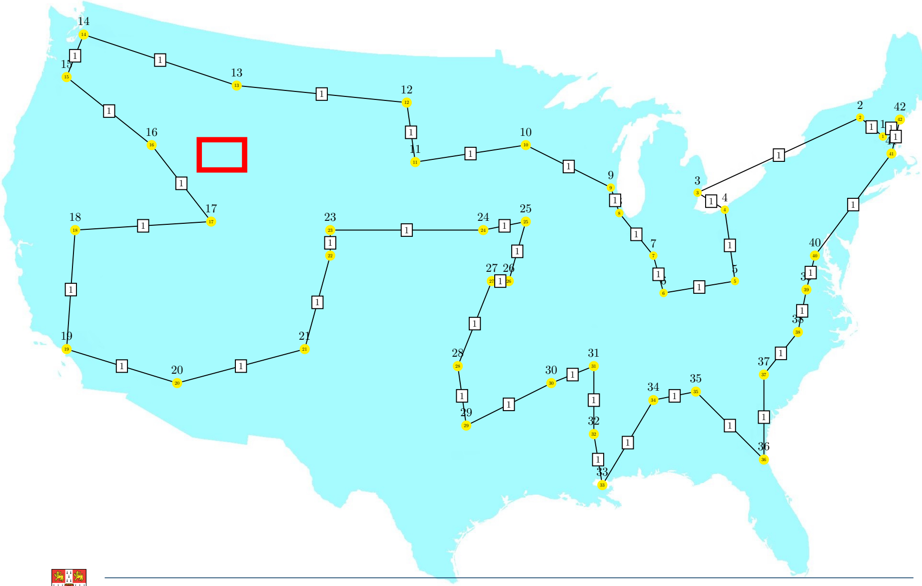
Iteration 9:

Objective value: -699.000000, 861 variables, 953 constraints, 2281 iterations



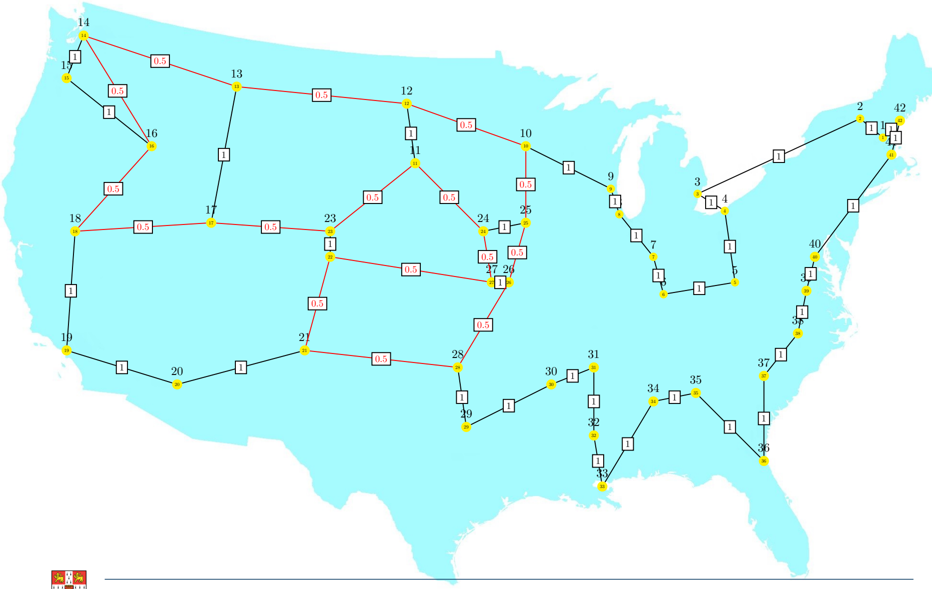
Iteration 9: Branch 2b $x_{17,13} = 1$

Objective value: -699.000000, 861 variables, 953 constraints, 2281 iterations



Iteration 10:

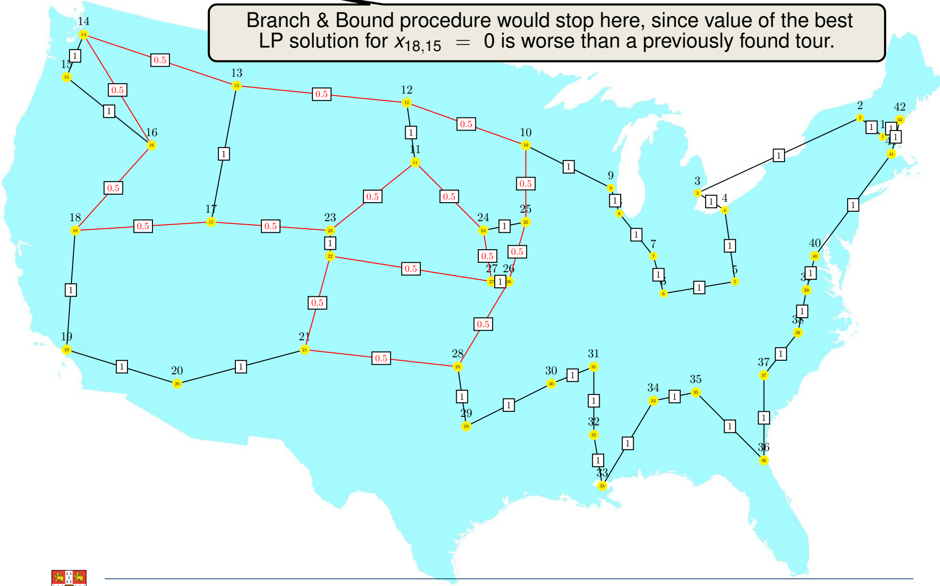
Objective value: −700.000000, 861 variables, 954 constraints, 2398 iterations



Iteration 10:

Objective value: −700.000000, 861 variables, 954 constraints, 2398 iterations

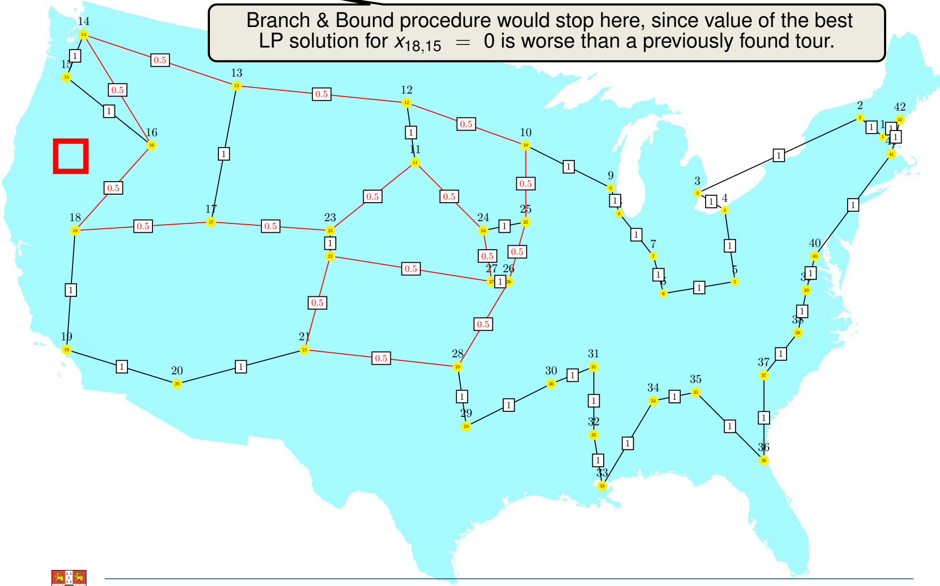
Branch & Bound procedure would stop here, since value of the best LP solution for $x_{18,15} = 0$ is worse than a previously found tour.



Iteration 10: Branch 1b $x_{18,15} = 1$

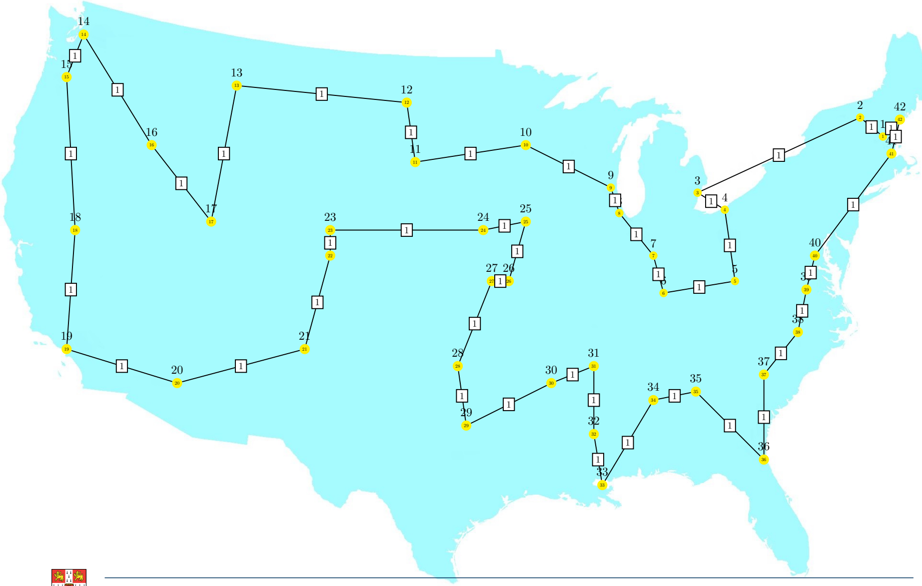
Objective value: -700.000000, 861 variables, 954 constraints, 2398 iterations

Branch & Bound procedure would stop here, since value of the best LP solution for $x_{18,15} = 0$ is worse than a previously found tour.



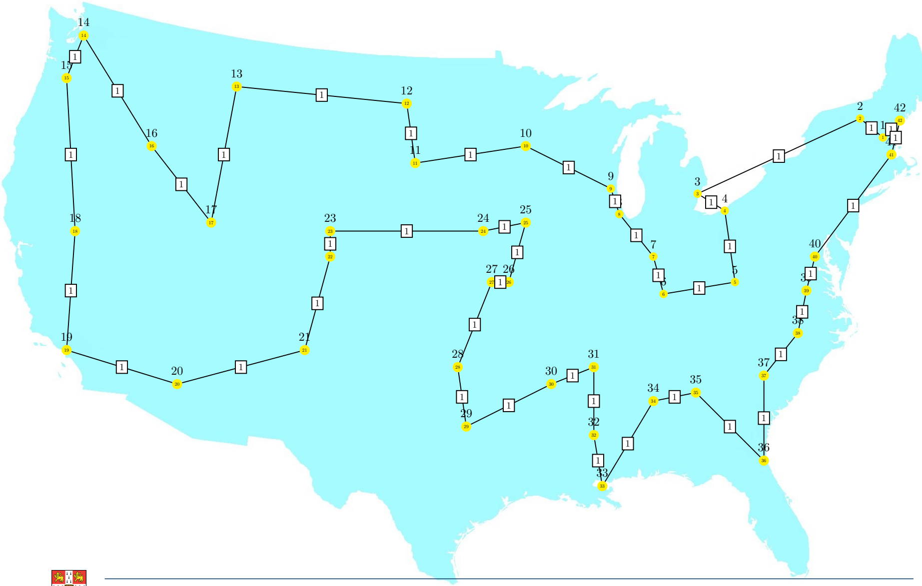
Iteration 11:

Objective value: −701.000000, 861 variables, 953 constraints, 2506 iterations



Iteration 11: Branch & Bound terminates

Objective value: −701.000000, 861 variables, 953 constraints, 2506 iterations



Branch & Bound Overview

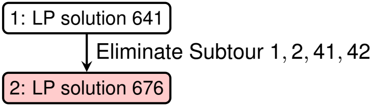
1: LP solution 641

Branch & Bound Overview

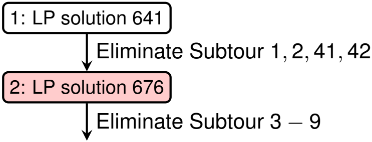
1: LP solution 641

↓ Eliminate Subtour $1, 2, 41, 42$

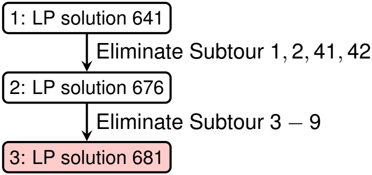
Branch & Bound Overview



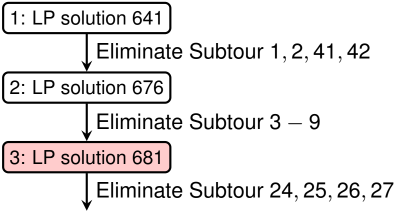
Branch & Bound Overview



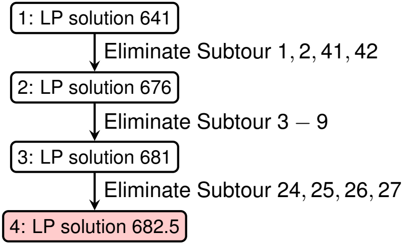
Branch & Bound Overview



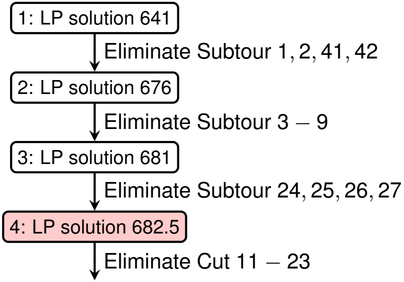
Branch & Bound Overview



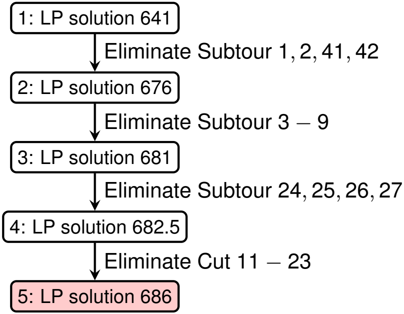
Branch & Bound Overview



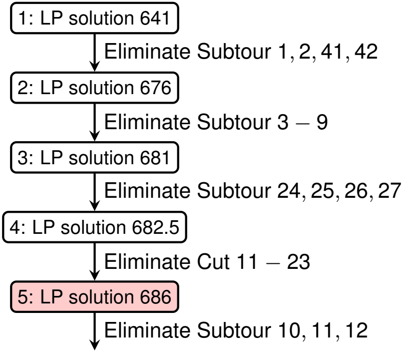
Branch & Bound Overview



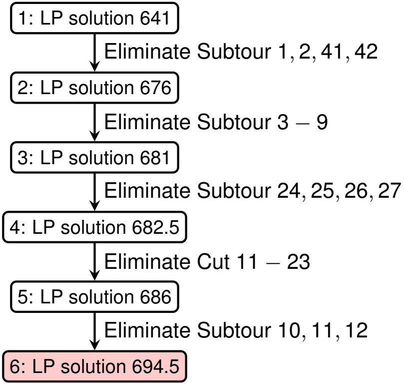
Branch & Bound Overview



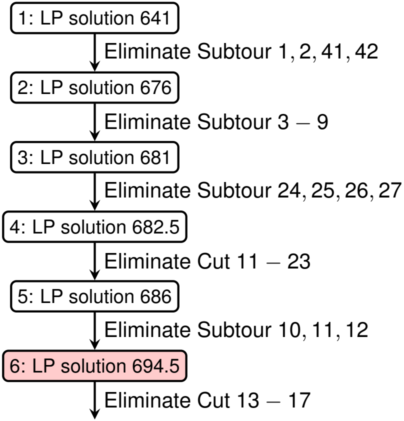
Branch & Bound Overview



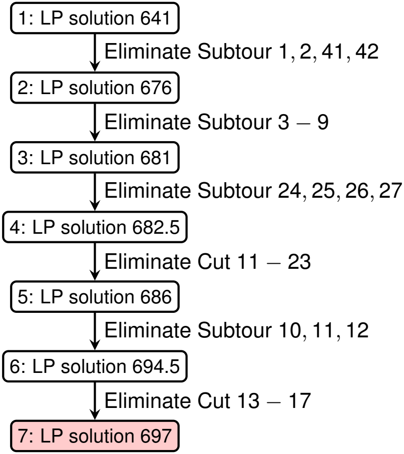
Branch & Bound Overview



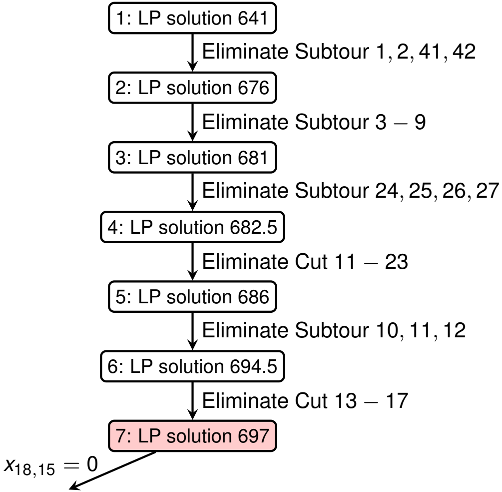
Branch & Bound Overview



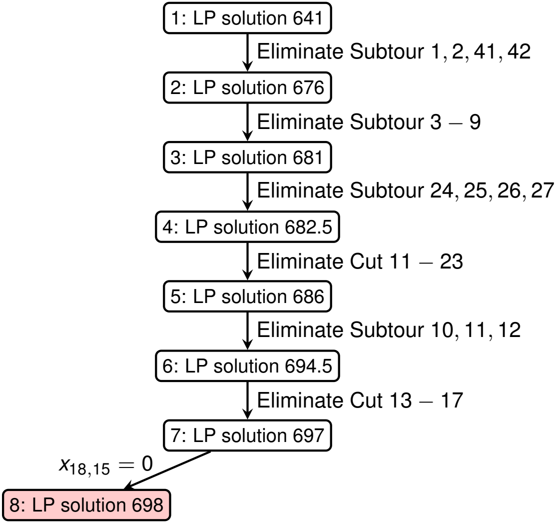
Branch & Bound Overview



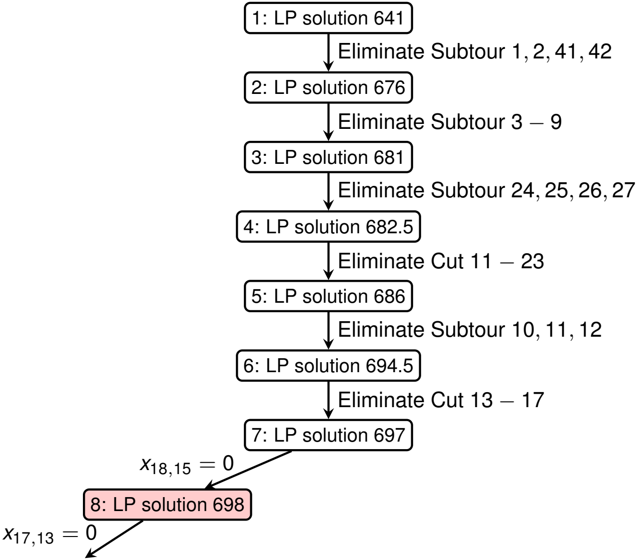
Branch & Bound Overview



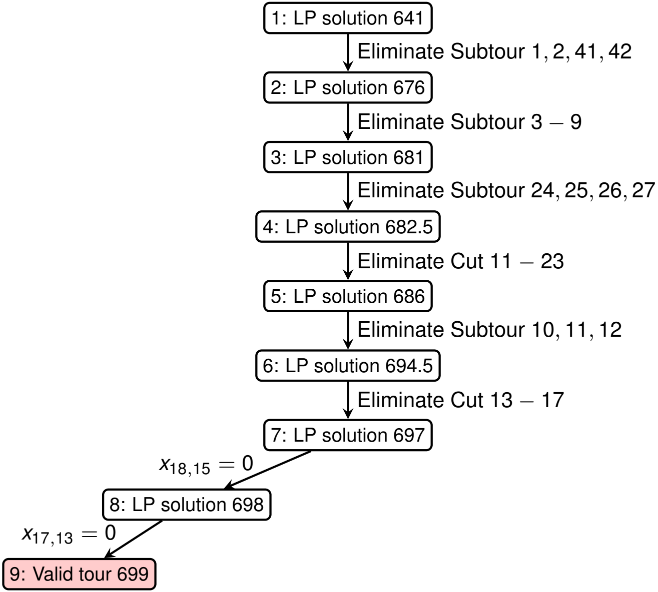
Branch & Bound Overview



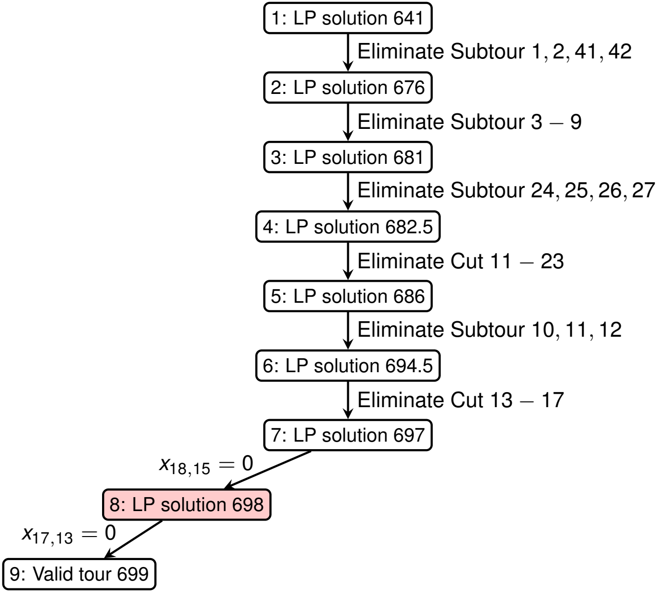
Branch & Bound Overview



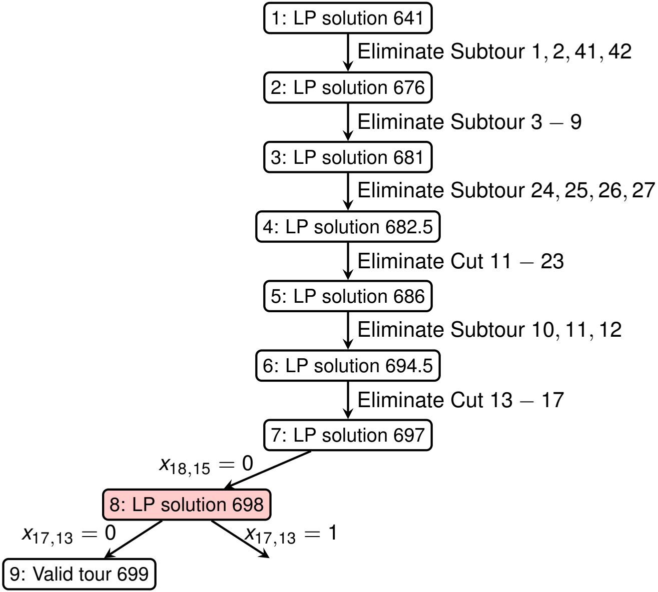
Branch & Bound Overview



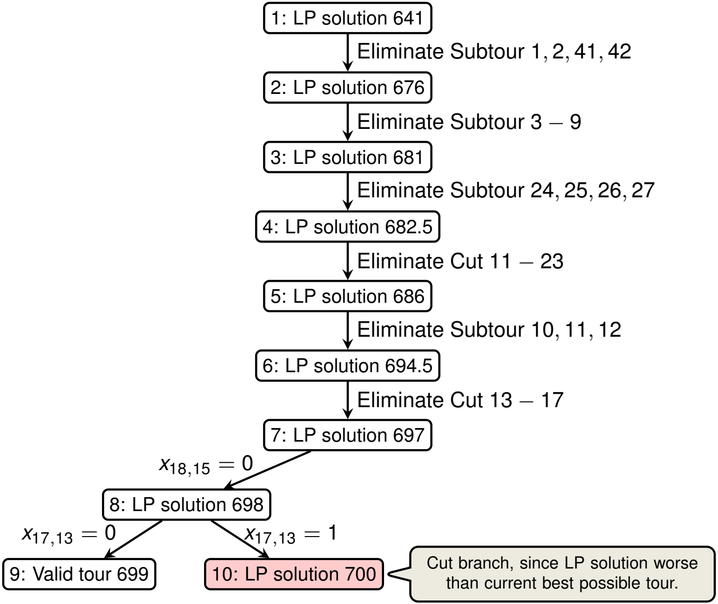
Branch & Bound Overview



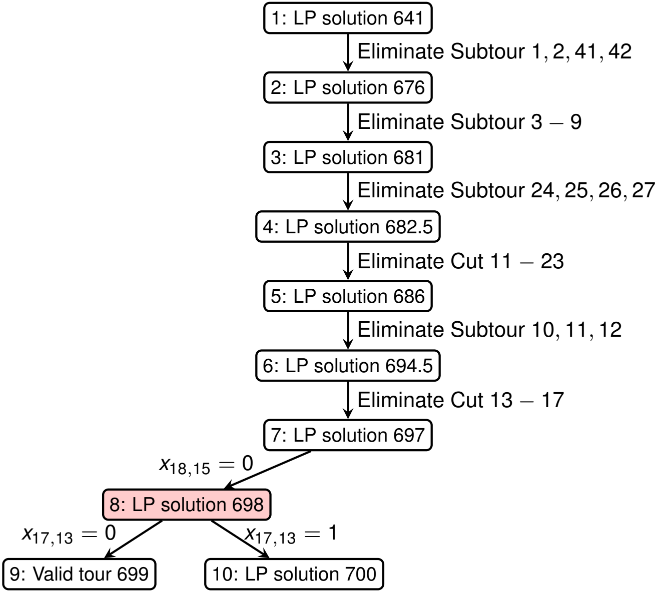
Branch & Bound Overview



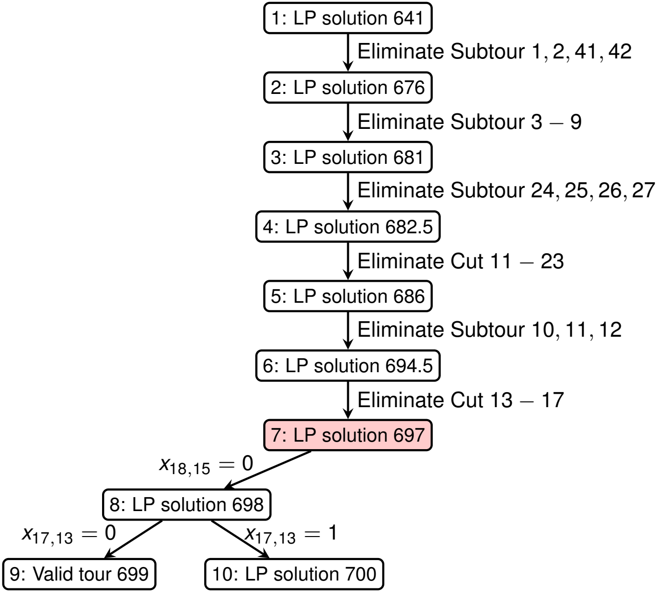
Branch & Bound Overview



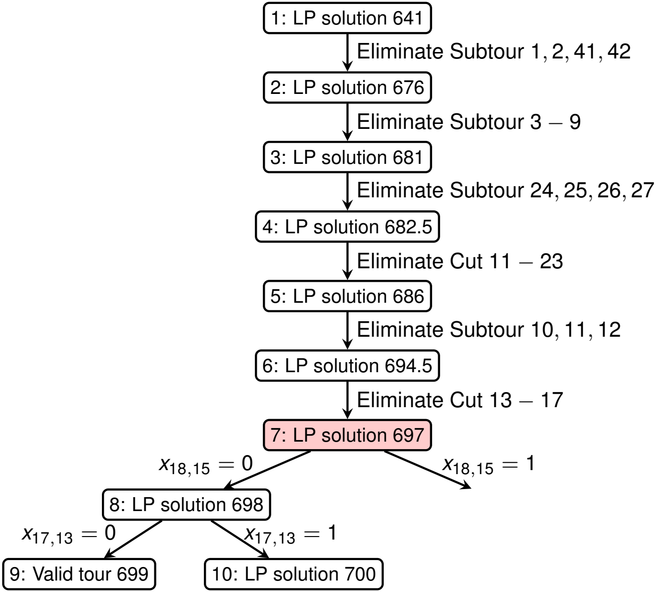
Branch & Bound Overview



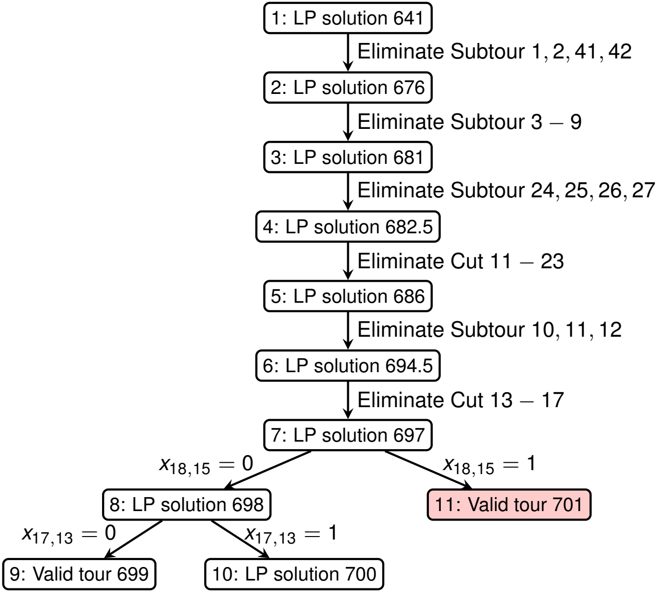
Branch & Bound Overview



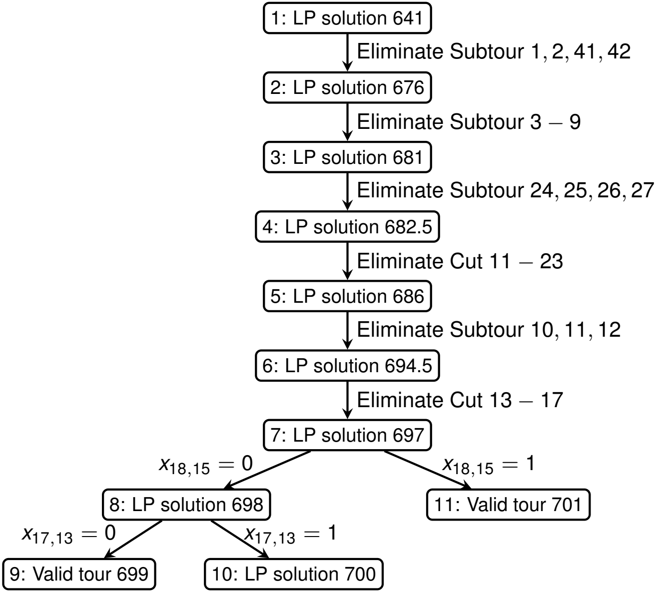
Branch & Bound Overview



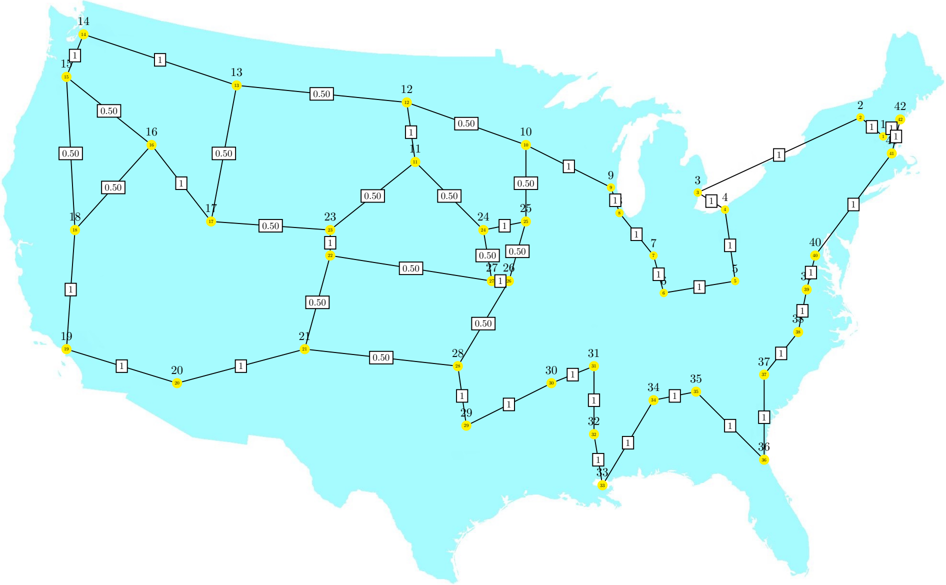
Branch & Bound Overview



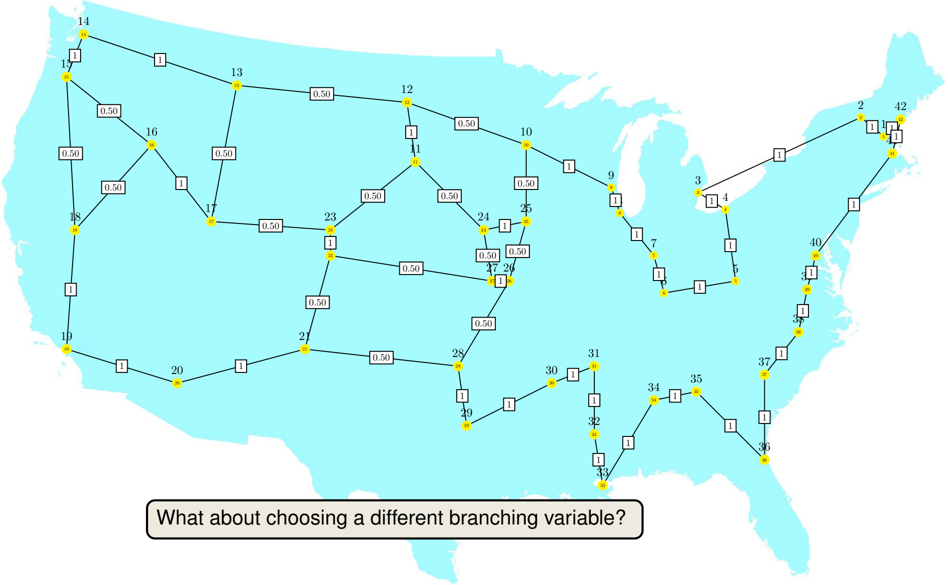
Branch & Bound Overview



Iteration 8: Objective 697

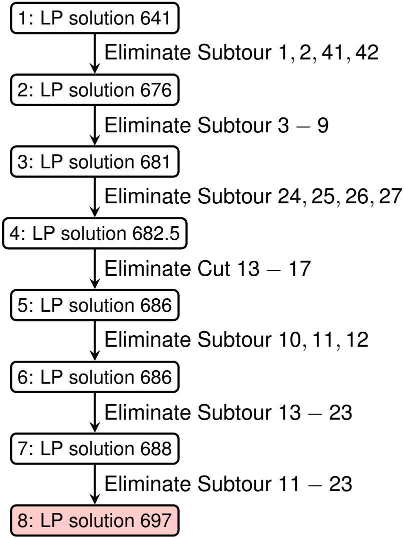


Iteration 8: Objective 697

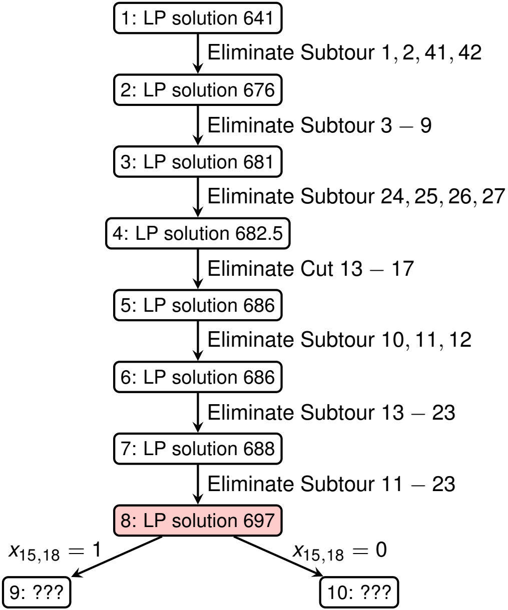


What about choosing a different branching variable?

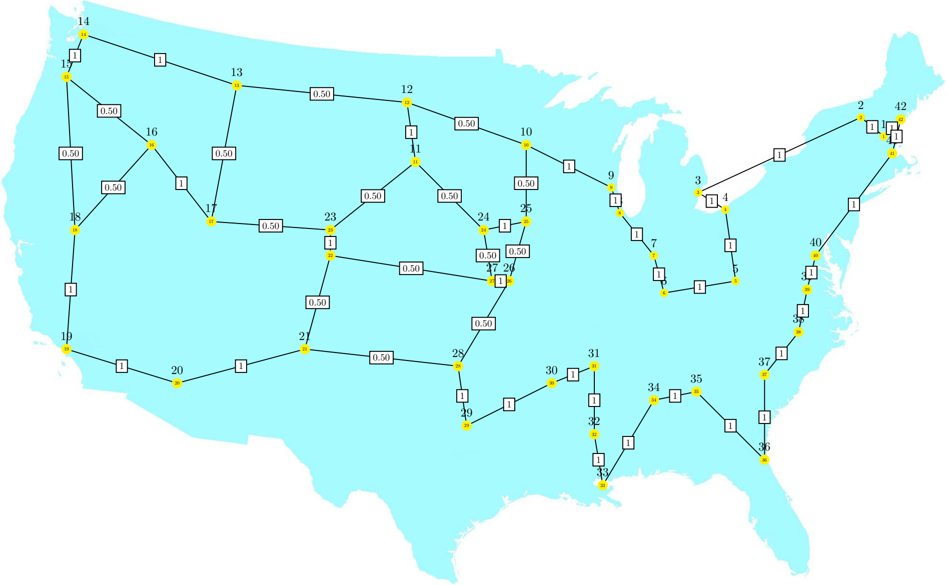
Solving Progress (Alternative Branch 1)



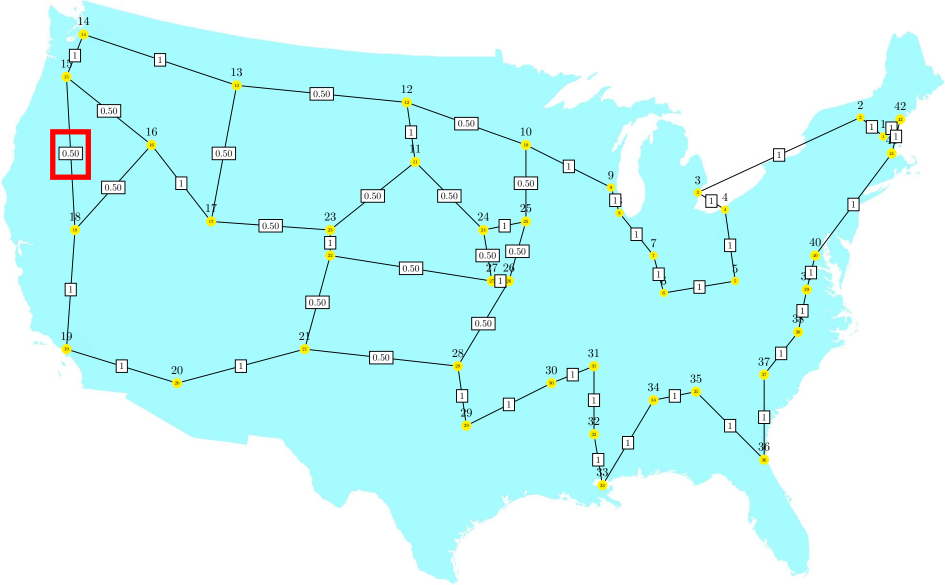
Solving Progress (Alternative Branch 1)



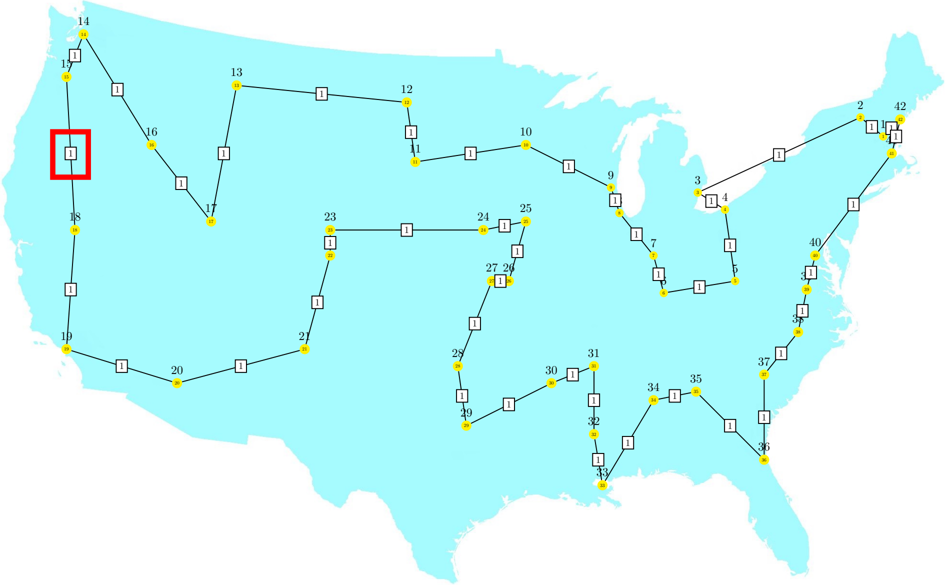
Alternative Branch 1: $x_{18,15}$, Objective 697



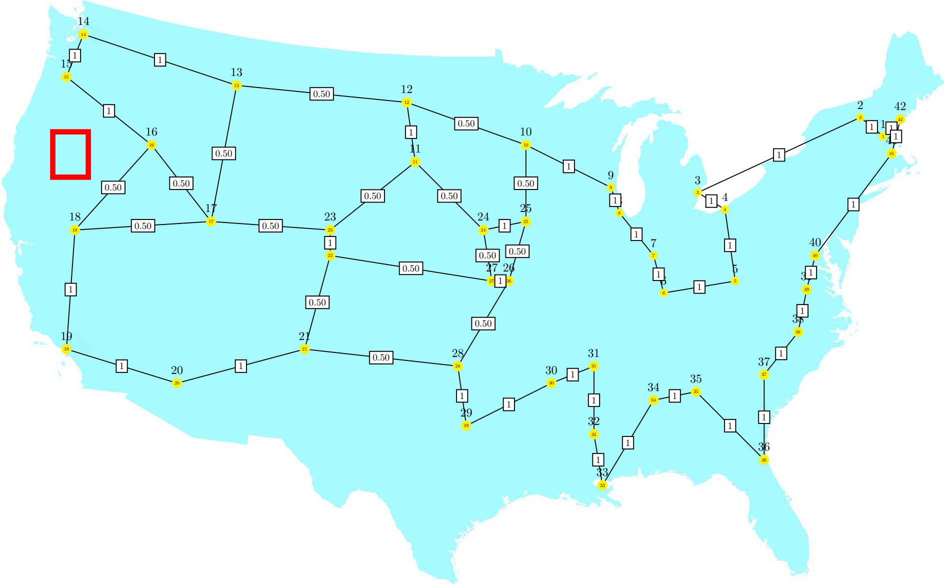
Alternative Branch 1: $x_{18,15}$, Objective 697



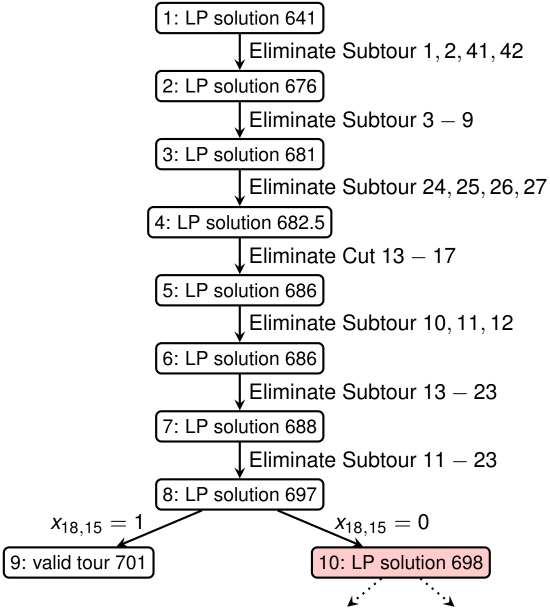
Alternative Branch 1a: $x_{18,15} = 1$, Objective 701 (Valid Tour)



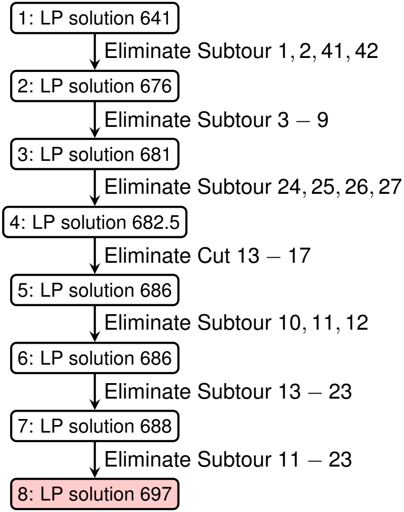
Alternative Branch 1b: $x_{18,15} = 0$, Objective 698



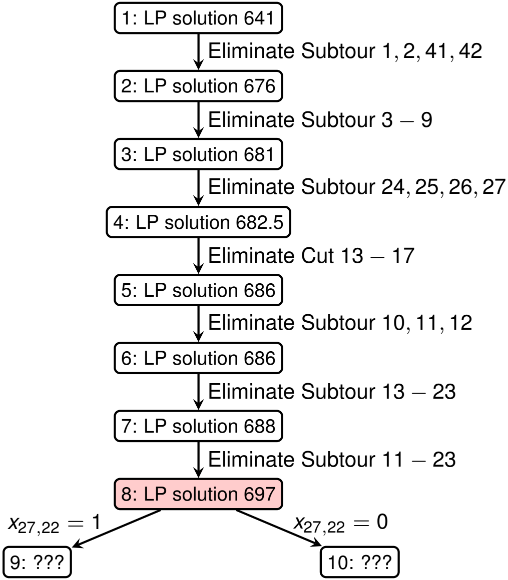
Solving Progress (Alternative Branch 1)



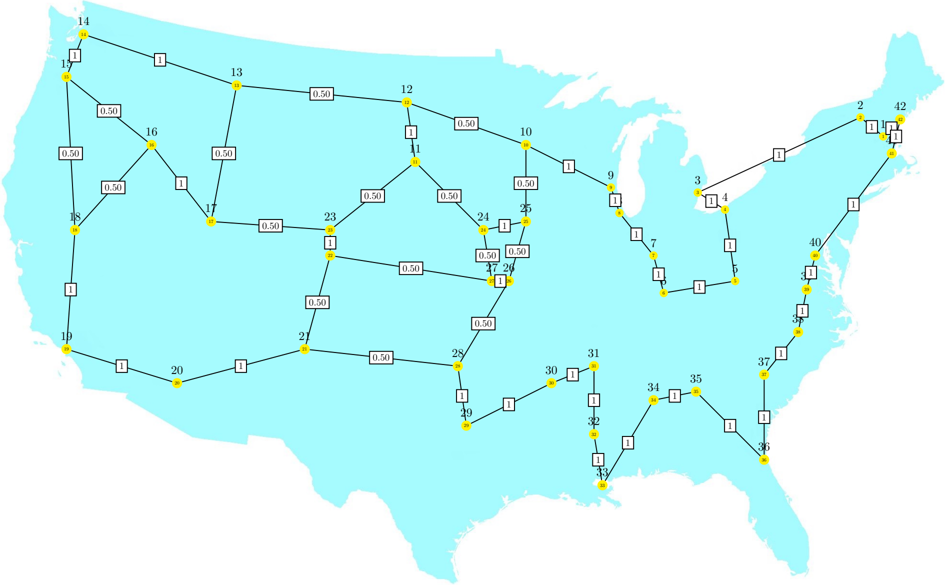
Solving Progress (Alternative Branch 2)



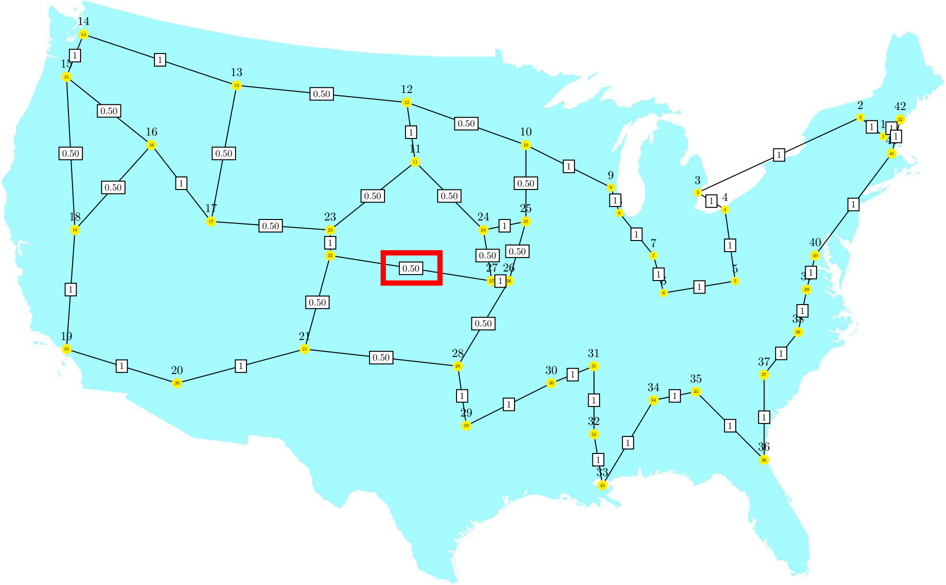
Solving Progress (Alternative Branch 2)



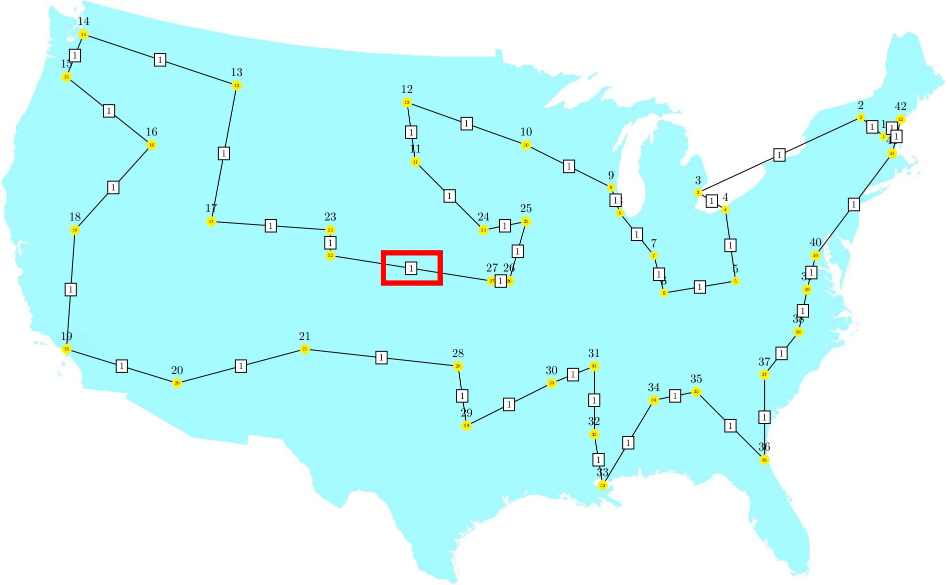
Alternative Branch 2: $x_{27,22}$, Objective 697



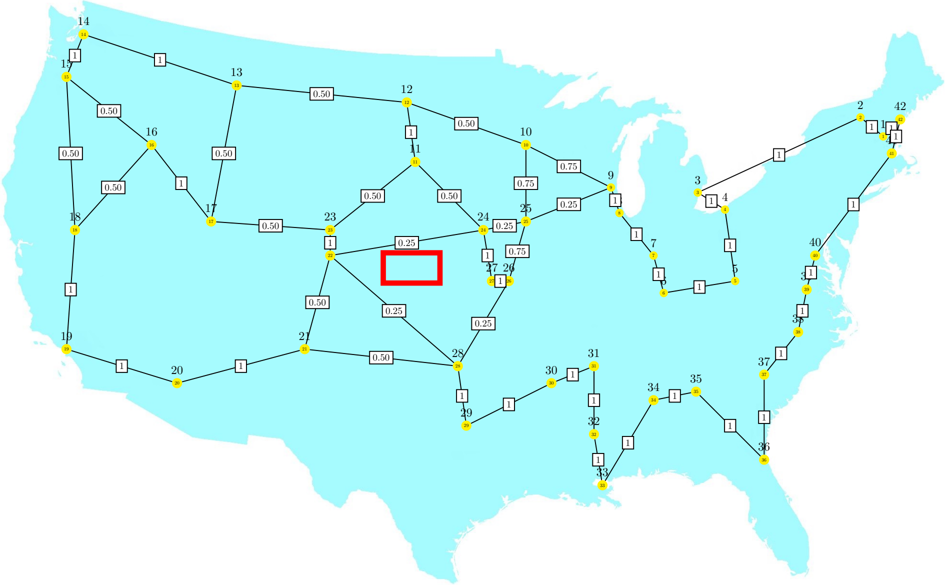
Alternative Branch 2: $x_{27,22}$, Objective 697



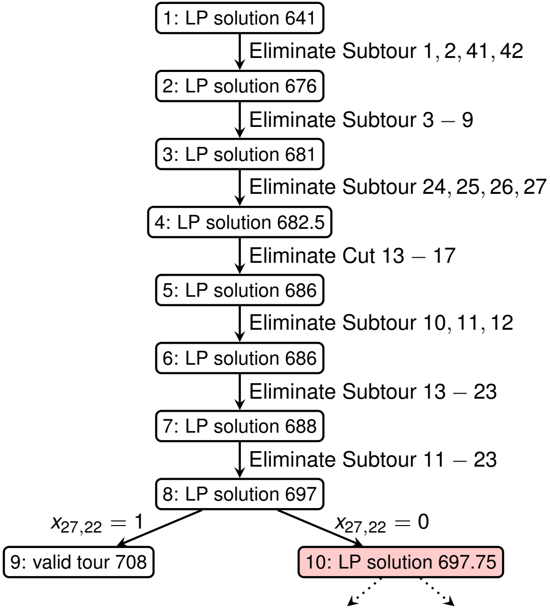
Alternative Branch 2a: $x_{27,22} = 1$, Objective 708 (Valid tour)



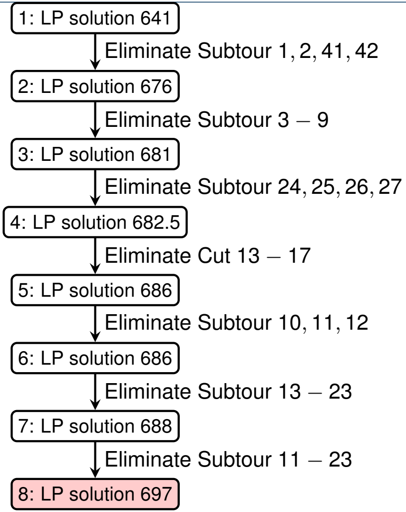
Alternative Branch 2b: $x_{27,22} = 0$, Objective 697.75



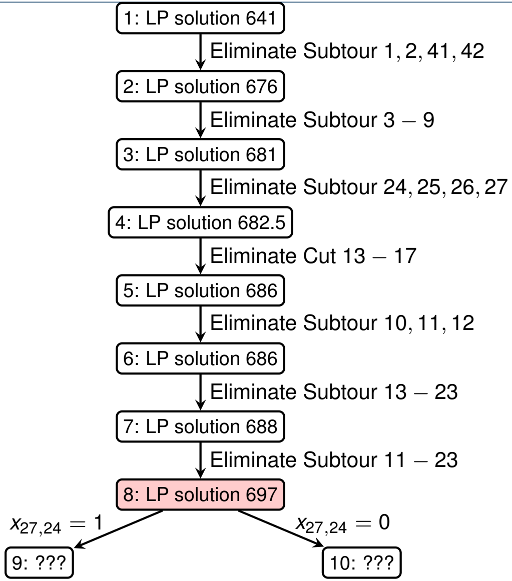
Solving Progress (Alternative Branch 2)



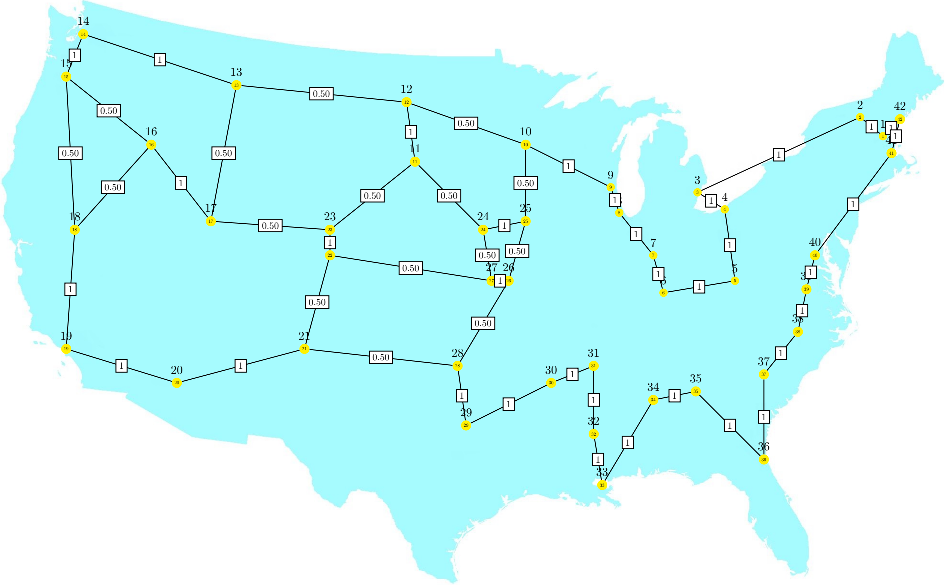
Solving Progress (Alternative Branch 3)



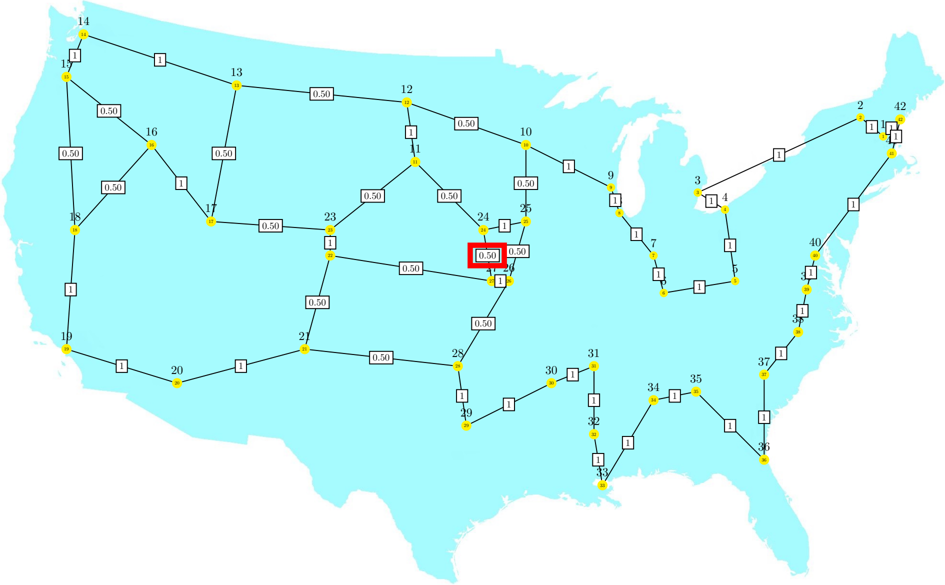
Solving Progress (Alternative Branch 3)



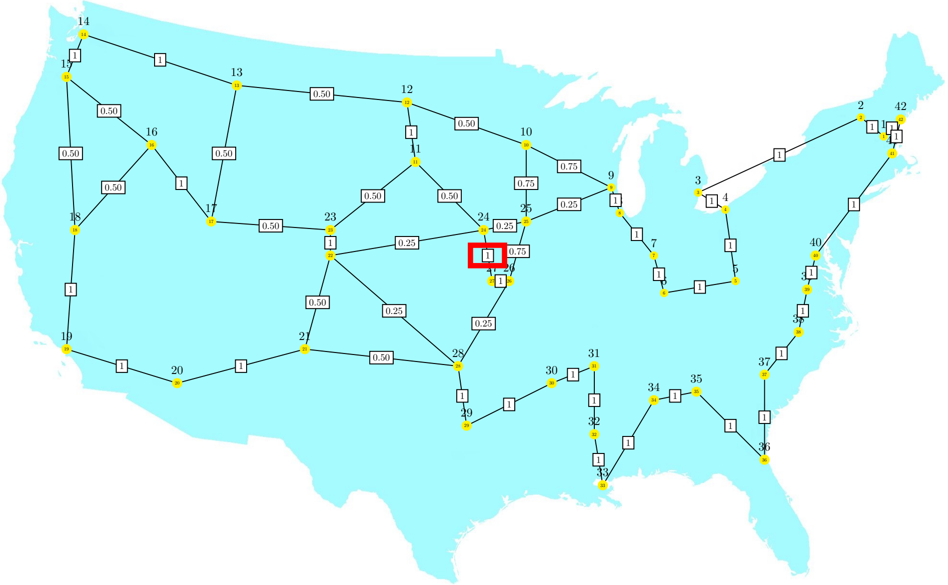
Alternative Branch 3: $x_{27,24}$, Objective 697



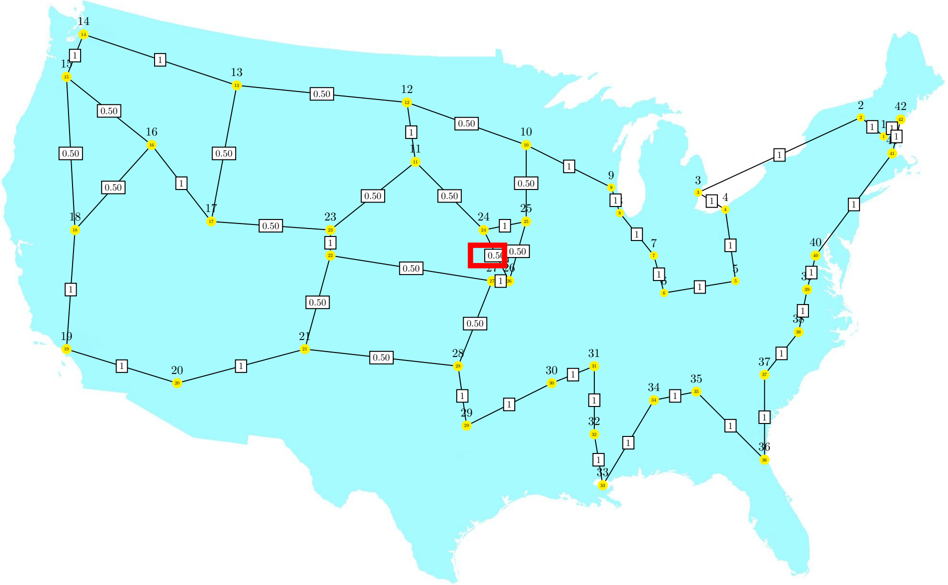
Alternative Branch 3: $x_{27,24}$, Objective 697



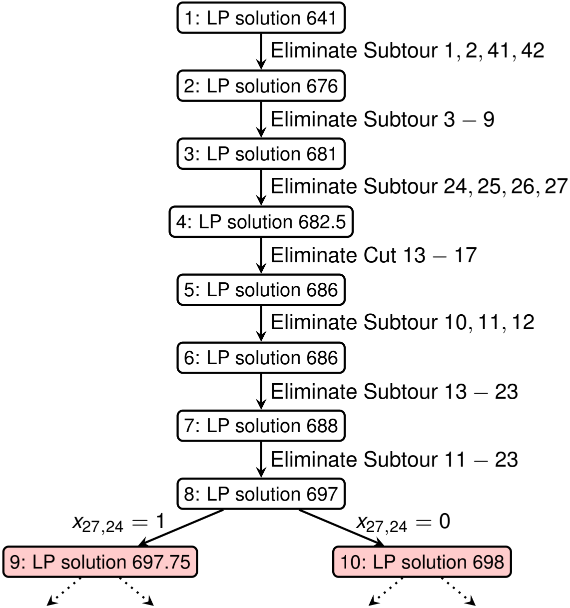
Alternative Branch 3a: $x_{27,24} = 1$, Objective 697.75



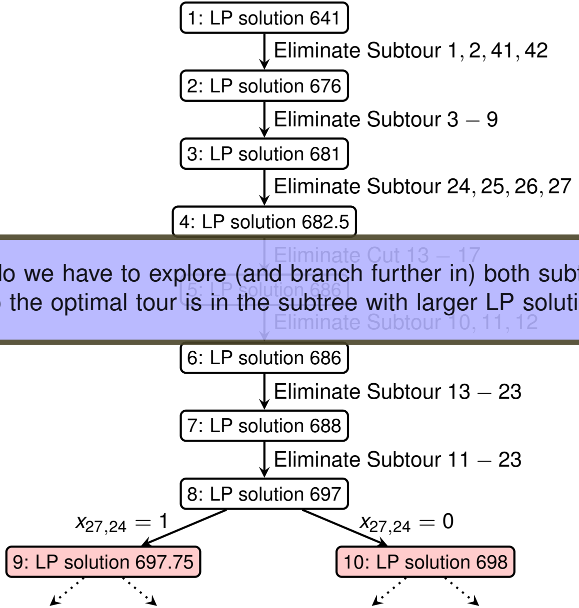
Alternative Branch 3b: $x_{27,24} = 0$, Objective 698



Solving Progress (Alternative Branch 3)



Solving Progress (Alternative Branch 3)



Not only do we have to explore (and branch further in) both subtrees, but also the optimal tour is in the subtree with larger LP solution!

Conclusion (1/2)

- How can one generate these constraints automatically?

Conclusion (1/2)

- How can one generate these constraints automatically?
  Subtour Elimination: Finding Connected Components
  Small Cuts: Finding the Minimum Cut in Weighted Graphs

Conclusion (1/2)

- How can one generate these constraints automatically?
  Subtour Elimination: Finding Connected Components
  Small Cuts: Finding the Minimum Cut in Weighted Graphs

- Why don't we add all possible Subtour Eliminiation constraints to the LP?

Conclusion (1/2)

- How can one generate these constraints automatically?
  Subtour Elimination: Finding Connected Components
  Small Cuts: Finding the Minimum Cut in Weighted Graphs

- Why don't we add all possible Subtour Eliminiation constraints to the LP?
  There are exponentially many of them!

Conclusion (1/2)

- How can one generate these constraints automatically?
  Subtour Elimination: Finding Connected Components
  Small Cuts: Finding the Minimum Cut in Weighted Graphs

- Why don't we add all possible Subtour Eliminiation constraints to the LP?
  There are exponentially many of them!

- Should the search tree be explored by BFS or DFS?

Conclusion (1/2)

- How can one generate these constraints automatically?
  Subtour Elimination: Finding Connected Components
  Small Cuts: Finding the Minimum Cut in Weighted Graphs

- Why don't we add all possible Subtour Eliminiation constraints to the LP?
  There are exponentially many of them!

- Should the search tree be explored by BFS or DFS?
  BFS may be more attractive, even though it might need more memory.

Conclusion (1/2)

- How can one generate these constraints automatically?
  Subtour Elimination: Finding Connected Components
  Small Cuts: Finding the Minimum Cut in Weighted Graphs

- Why don't we add all possible Subtour Eliminiation constraints to the LP?
  There are exponentially many of them!

- Should the search tree be explored by BFS or DFS?
  BFS may be more attractive, even though it might need more memory.

CONCLUDING REMARK

It is clear that we have left unanswered practically any question one might pose of a theoretical nature concerning the traveling-salesman problem; however, we hope that the feasibility of attacking problems involving a moderate number of points has been successfully demonstrated, and that perhaps some of the ideas can be used in problems of similar nature.

Conclusion (2/2)

- Eliminate Subtour $1, 2, 41, 42$
- Eliminate Subtour $3 - 9$
- Eliminate Subtour $10, 11, 12$
- Eliminate Subtour $11 - 23$
- Eliminate Subtour $13 - 23$
- Eliminate Cut $13 - 17$
- Eliminate Subtour $24, 25, 26, 27$

Conclusion (2/2)

- Eliminate Subtour $1, 2, 41, 42$
- Eliminate Subtour $3 - 9$
- Eliminate Subtour $10, 11, 12$
- Eliminate Subtour $11 - 23$
- Eliminate Subtour $13 - 23$
- Eliminate Cut $13 - 17$
- Eliminate Subtour $24, 25, 26, 27$

THE 49-CITY PROBLEM*

The optimal tour $\bar{x}$ is shown in Fig. 16. The proof that it is optimal is given in Fig. 17. To make the correspondence between the latter and its programming problem clear, we will write down in addition to 42 relations in non-negative variables (2), a set of 25 relations which suffice to prove that $D(x)$ is a minimum for $\bar{x}$. We distinguish the following subsets of the 42 cities:

$S_1 = \{1, 2, 41, 42\}$
$S_2 = \{3, 4, \cdots, 9\}$
$S_3 = \{1, 2, \cdots, 9, 29, 30, \cdots, 42\}$
$S_4 = \{11, 12, \cdots, 23\}$
$S_5 = \{13, 14, \cdots, 23\}$
$S_6 = \{13, 14, 15, 16, 17\}$
$S_7 = \{24, 25, 26, 27\}$.