

# Assignment 1

Anders Wind Steffensen (qrj741@alumni.ku.dk)

26-09-2017

## 1 Theoretical part - formulation and lower bounds

Given a complete directed graph  $G = (V, E), c$  with  $|V| = n$  where  $c_{ij}$  is the cost of the edge from vertex  $i$  to vertex  $j$ , the asymmetric travelling salesman problem is to find a minimum cost Hamiltonian tour of  $G$ .

The subtour integer linear programming formulation for the travelling salesman as defined in the assignment:

$$x_{ij} = \begin{cases} 1, & \text{if edge (i,j) is included} \\ 0, & \text{otherwise} \end{cases}$$

$$\text{max: } \sum_{i,j \in V} c_{ij} x_{ij} \quad (1)$$

$$\text{s.t } \sum_{i \in V, i \neq j} x_{ij} = 1 \quad \text{for } j \in V \quad (2)$$

$$\sum_{j \in V, j \neq i} x_{ij} = 1 \quad \text{for } i \in V \quad (3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \text{ for } S \subset V \text{ such that } 2 \leq |S| \leq n - 2 \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad (5)$$

### 1.1

First we prove that any feasible solution will satisfy the above constraints. In this proof we do not need to take into consideration the cost since any Hamiltonian tour is a feasible solution. In a feasible solution every vertex has exactly two edges incident to it. The constraints (2) and (3) satisfies this since for any vertex  $v$  edges  $(h, v)$  and  $(v, k)$  must exist for any  $h, k \neq v$ . A feasible solution also has every vertex incident to two edges which satisfies this constraint. Lastly a feasible solution only consists of one tour. A tour of  $h$  vertices consists of  $h$  edges, every proper subset of such a tour have strictly less than  $h$  edges. The feasible solution will therefore satisfy constraint (4). We have now shown that any feasible solution will satisfy the constraints.

could use more argumentation.

rewrite to be based on one constraint after the other

Now we need to proof that any solution satisfying the constraints is also a feasible solution. Constraints (2) and (3) are trivial in the sense that they still require each vertex to have exactly two edges incident to them as a feasible solution should have. Now that any solution requires each vertex to have two incident edges, two situations can occur, either we have a Hamiltonian tour and the solution is feasible or we have multiple smaller cycles. As described before cycle must have as many edges as vertices and constraint (4) limits the solution to not have any cycles for any proper subset of  $S$  since the number of edges has to be strictly lower than the size of the solution. Therefore a cycle can only exist for the entire graph and if it does so we have a Hamiltonian tour. This concludes our proof.

## 1.2

First we calculate the size of  $S \subset V$  such that  $|S| \leq n - 2$ . This amount to picking every subset of size  $n - 2$  out of  $n$ . Secondly we subtract any subset of size 1 which naturally is  $n$ . This gives the following size  $\frac{n!}{(n-2)!(n-(n-2)!)} - n \equiv \frac{n!}{(n-2)!2} - n$ .

$$\max: \sum_{i,j \in V} c_{ij} x_{ij} \quad (6)$$

$$\text{s.t } \sum_{i \in V, i \neq j} x_{ij} = 1 \quad \text{for } j \in V \quad (7)$$

$$\sum_{j \in V, j \neq i} x_{ij} = 1 \quad \text{for } i \in V \quad (8)$$

$$t_j \geq t_i + 1 - n(1 - x_{ij}) \text{ for } i \in V, j \in V \setminus \{i\} \quad (9)$$

$$x_{ij} \in \{0, 1\} \quad (10)$$

$$t_i \in \mathbb{R}_+ \quad (11)$$

## 1.3

The constraint runs over two variables,  $i, j$  which can have  $n$  and  $n - 1$  different values respectively. Therefore there are  $n(n - 1) = n^2 - n$  many constraints.

## 1.4

The constraints of the sub tour formulation might provide for a better branching decision in a branch and bound scenario and further more the constraints might be restricting more to integer solutions, so branch and cut might not be necessary.

need to  
think more  
on this

## **1.5**

Consider the minimum cost Hamiltonian tour, it goes through vertex 1. Vertex 1 has exactly two edges incident to it, if we remove one of them the solution is now a tree with 1 as root covering every vertex in  $V$ , which is exactly what the lower bound algorithm finds (and it will always find the minimum cost one since it is less restricted). If the removed edge from the Hamiltonian tour is the edge with the lowest cost then the lower bound algorithm would also pick it, and if not the lower bound is still adhered to.

## **2 Implementation part - branch-and-bound**

### **2.1**

### **2.2**

### **2.3**

### **2.4**

### **2.5**