93

TSO - Fall '25 - Midterm Exam

Name: Adriana Concepción Aguirre Dorantes 45 + 47.5

①.- Combinatorial optimization problem

It's an optimization problem where you have to combine instances to make a solution to a problem.

+4 Too vague

② brute force enumeration +5.5

Is to resolve the problem by doing all posible solutions or combinations one by one and then what?

③ When do we say that an optimization problem is easy to solve.

+6 When there is already an algorithm that solves that problem in a reasonable time and computational cost.

④ Define and explain what a heuristic method is for solving combinatorial optimization problems.

+6 It's a method to find a feasible solution to a problem in a reasonable time not guaranteeing ~~an~~ the optimal.

⑤ Under what circumstances or when it is prefereable to use a heuristic method instead of an exact optimization method

+5.5 It depends on the time, and computational cost, if you need a quick feasible solution and your problem is not so easy to solve the best option is a heuristic.

⑥ .- What is a constructive heuristic?

+4 It's a heuristic where you add a variable at every iteration to the solution

+7 ⑦ .- Describe in detail the Nearest ~~Inserti-~~ Neighbor heuristic.

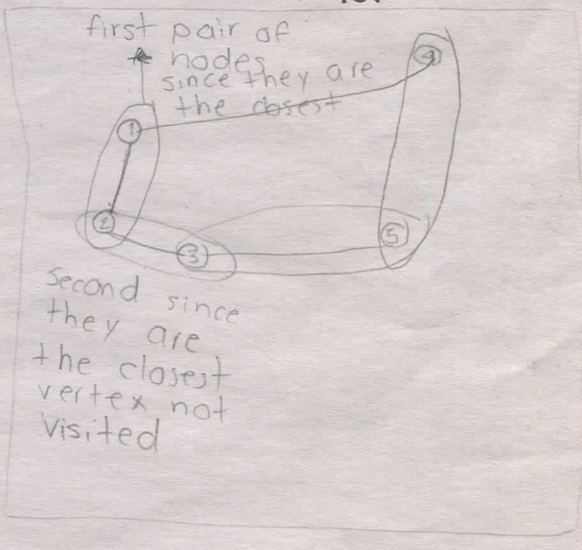
Data → Given set of cities (N) and a cost matrix C_{ij}

Decision → $\begin{cases} x_i = 0 \text{ if } v \text{ is not selected} \\ x_i = 1 \text{ if } v \text{ is selected} \end{cases}$ $\forall\, v \in V$

OptFunc → $\min \sum_{v \in V} C_{ij} X_{ij}$

Constraints → The solution must be a Tour including all nodes exactly once

→ We are going to select the closest node at every iteration. Not included to the solution yet



Inicilization: Select j

Do $t = j$ and $W = V \setminus \{j\}$

2. While $W \neq 0$

Take i from $W/\{j\} = \min \{C_{ij} / i \text{ en } W\}$

Conect t to j

Do $W = W \setminus \{j\}$ and $t = j$

⑧ Nearest Insertion +7

Given $V\{v_1, v_2, v_3 \ldots v_n\}$ and $\{C_{ij}\}$

Decision $\begin{cases} X_0 = 1 & \text{If } v \text{ is selected} \\ X_0 = 0 & \text{otherwise} \end{cases} \quad \forall v \in V$

Opt Func = $\text{Min} \sum_{v \in V} C_{ij} X_{ij}$

Constraints:

the solution must be a tour with all cities exactly once

① Select subtour with K vertex

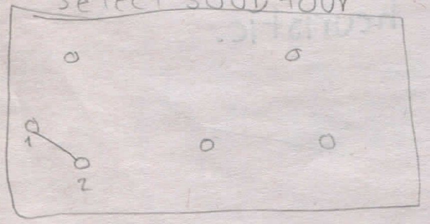
② Do $W = V \setminus \{K\}$

③ While $(W \neq 0)$ take j from W

④ Insert j with minimum increase.

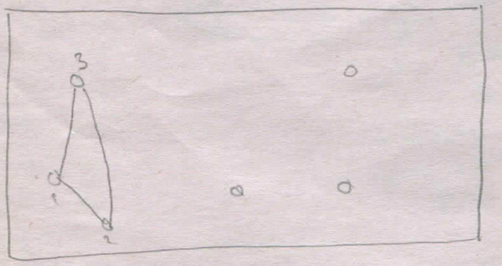
⑤ Do $W = W \setminus \{j\}$

If whe have Select soubtour



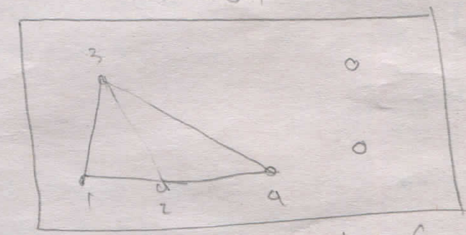
since they are the closest to each other.

1^{st} Iteration



Since the cost of adding city 3 is the minimum cost

2^{nd} Iteration



Since the cost of adding city 4 is the minimun cost among all nodes not included in the solution

9.- Set covering Problem

a) (1,0,0,1,0,0,1,0,0,1,0,0) means that subsets 1, 4, 7, 10 of J are chosen.

+6

1 covers : 1, 3, 5, 7
4 covers : 2, 3, 4, 8
7 covers : 1, 2, 3
10 covers : 2, 5, 6 8

No.
It's not a feasible solution since it doesn't cover a_9 and a_{10}

b) (0,1,0,0,0,0,1,0,1,0,0,1) = 2, 7, 9, 12

+6

2 covers : 2, 6, 9, 10
7 covers : 1, 2, 3
9 covers : 4, 7, 8
12 covers : 5, 6, 7, 10

Yes
It's a feasible solution because it covers all instances

c) $X^3 = (0,1,0,0,0,0,1,0,1,0,0,1) = 2, 7, 9, 12$

$\sum_{j \in X} c_j = 4 + 3 + 17 + 15 = 40 \Leftarrow 39$

$X^4 = (0,0,1,1,0,0,0,0,0,1,0,0) = 3, 4, 10$

$\sum_{j \in X} c_j = 22 + 6 + 12 = 40$

1 2 3
2 3 4 8
2 6 9 10

$X_5 = (0,1,0,1,0,0,1,0,0,0,0,0) = 2, 4, 7$

$\sum_{j \in X} c_j = 4 + 6 + 3 = 13$

Sorting: ① .- X^4 → Is the same cost but in X^3 we select more subsets.
② .- X^3
④ .- X_5 → It's not a feasible solution because a_5 is not being covered

+7.5

d) Design a constructive heuristic for the SCP.

+19 $S=\{1,2,m\}$ $A=\{a_{ij}\}$ $a_{ij}=\begin{cases}1 & \text{if is covered}\\ 0 & \text{, otherwise}\end{cases}$

$H=\{H_1,H_2,\dots\}$ $C_i\ \forall H_i$ $X_i\ \forall H_i$ $X_i=\begin{cases}1 & \text{if selected}\\ 0 & \text{otherwise}\end{cases}$

$\text{Min}\{C_1X_1+C_2X_2+C_nX_n\}$

St $a_{1j}X_1+a_{2j}X_2\ldots\geq 1$

$j=1,\ldots m$

$X_i=1,0$ $i=1,\ldots,n$

I got your idea but you need to learn to write pseudocode

Select decision → 1._ First we are going to pick the subset with more items

2.- Select the subset with more items not being covered

If there are two or more subsets with the same amount of items select the one with minimum cost.

3._ Continue until there's no items uncovered

e)

1._ Select subset 3, since is the subset with more items. cost (22) Items covered: 1, 2, 4, 7, 9, 10

+10

2.- Select subset 10 since is the subset with more items not being covered and with minimom cost (12).
items covered: 1, 2, 4, 5, 6, 7, 8, 9, 10 cost: 34

3._ Select subset 7 since is the subset with items not being covered with minimum cost (3)
items covered: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 cost: 37

Result:

subsets selected: 3, 10, 7

Total Cost: 37