Name: Gerardo Gael Peña Cital TSO 70

1894607

36.5+20. + 13 work during semester

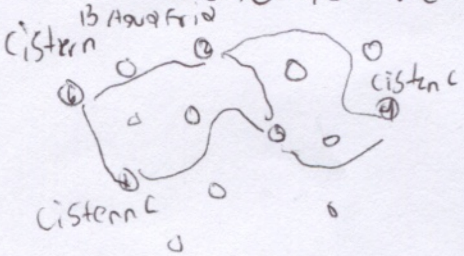
1 its a problem with needs a better solution, minimize or +1.5 maxime, can take a example for the tsp problem, combine constrais, objective, feasible solution, and better way, search a make a constructive heuristic

2 is an algorith improve an aproach to find good answers +3 heuristics problem-specif to find the best way

3 simplex method, branch and bound we can take this methods to maximize and optimation withe feaseblie solution +2.5

4 An heuristic its an algorith to construct a solution ncrementally +5 the scratch

5 For local search wee can imagene a situation like the issue with the water in 2022, if we can put the better way on the cisterns we can take more water for the people +4



the point to this drow its imagine the number and the travel for this cisterns so, we need make a heuristic to find the best way to travel, this made make a better solution

6 find the shorest travel to recorr just like the example we +3.5 start in a random position, and we need finish at the same

We need optimatize the travel to make the short time, just lik a deliver guy

7 +2 For this insertion its the same but to this time we need put or make another point in the way ITS a NP-hard meaning polynomol

8 +5 ~~No~~ because with start with nothing so we can start with simple data and the see the constrains, see how we can the heuristic takes form its hard, while they are designed build solutions step by step

10 +4 the first big difference its the method tha we can take for the best found, we need see the exactly data, no the first so to this way we can

Best Found at the end evaluate all posible moves

(11) (a) no beacuase it doesnt match withe the points

+5

+5 (b) yes we can make a match each points we can just swap

(12) (c) $X^{(4)}$ $X^{(3)}$. $X^{(5)}$ we can put in order to make a problem sort

+4 if we order to this maner, we can get matche to each othe node

(8) Yes its a NP-hard meaning a no polynomal function at

+5 seach alogorithm is know we can solve with tsp problem.

+6

```
def local search_pdp
  def calculate _ dispersion
    return (sum (D[i][j] for in a subset for j in subset
    i != j)
     Improved= True
        while improved=
              Improved= false
           best increase = 0
            best_swap= None
for i n x:
   for j: in range (len(D):
               if j not in x:

calcutate _ dispersion (new)
ncrease = current-dispersion =
If increase > best_increas
best_swap =( i,j )
If best swap?
x = remove(i)
x = add (x)
Improved = True
     return x
```

We can make a local search Iteralitive refines the solution aimig to maxime the dispersion objective function