




Master in Innovation and Research in Informatics

Algorithmic Methods For Mathematical Models

Course Project Presentation

Roger Gili i Coscojuela
roger.gili.i@estudiantat.upc.edu

Edgar Rodríguez de la calle
edgar.rodriguez.de.la.calle@estudiantat.upc.edu



A decorative network diagram at the top of the slide, featuring a series of interconnected nodes and lines. A central node is highlighted with a dashed circle and contains a large blue double quote symbol.

“

*“Forget about Mr. Bond and forget about the door. This is a **TSP** problem”.*

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric rings, suggesting a hierarchical or multi-layered structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

1.

Problem statement

Initial Information

- ⊙ A set S that contains a fixed n number of codes.
- ⊙ Each code is a sequence of m binary digits.

```
m = 10;  
n = 6;  
S =  
[  
  [ 0 0 0 0 0 0 0 0 0 0 ]  
  [ 1 0 1 0 1 0 0 0 0 1 ]  
  [ 0 0 1 0 1 1 0 1 0 1 ]  
  [ 0 1 0 1 0 1 1 0 1 1 ]  
  [ 0 1 1 0 1 1 0 0 0 0 ]  
  [ 0 1 1 1 0 0 1 0 0 1 ]  
];
```



2.

Mixed Integer Linear Programming Model

Inputs and Variables of the Model

Inputs

n : The number of codes.

m : The number of binary digits that each code has.

S : The set of n codes with m digits

Variables

f : Discrete variable -> Flips

$d_{i,j}$: Discrete variable -> Cost.
($0 \leq i, j \leq n$)

$y_{i,j}$: Binary variable -> Path
($0 \leq i, j < n$)

$x_{k,j}$: Binary variable -> Subtour
($1 \leq k, j < n$)

Objective Function & Constraints

Objective Function:

Minimize: f

Subject To:

1: Intended Meaning of f



$$\sum_{i,j=0}^n d_{i,j} y_{i,j}$$

Objective Function & Constraints

Subject To:

2: Out Constraint



$$\sum_{j=1}^n x_{k,j} = 1$$

3: In Constraint



$$\sum_{k=1}^n x_{k,j} = 1$$

4: First Code is 0..0



$$y_{0,j} = x_{1,j} : \forall j \ (1 \leq j < n)$$

5: Last Code is 0..0



$$y_{j,0} = x_{(n-1),j} \ \forall j: \ (1 \leq j < n)$$

5: No subtour



$$x_{k,i} + x_{(k+1),j} - y_{i,j} \leq 1$$
$$\forall k: \ (1 \leq k < n - 2) \ \forall i, j: \ (1 \leq i, j < n)$$



3.

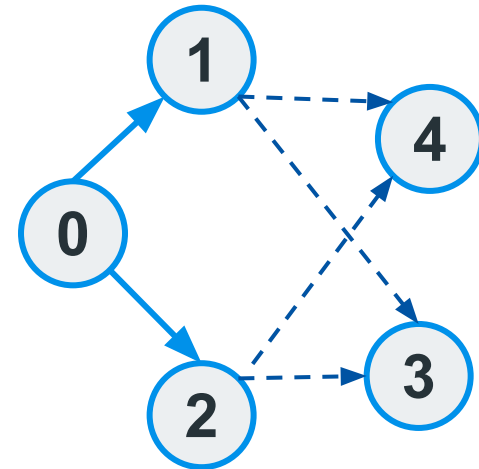
Heuristic Algorithms

Greedy

Algorithm 1 GREEDY

```
1: function GETFEASIBLELINKS( $\omega$ )
2:   Candidates = {}
3:   for i in  $\omega$  do
4:     if node[i] connections < 2 then
5:       for each j in N do
6:         if Node[j] connections == 0 then
7:           c = Cost of moving from Node[i] to Node[j]
8:           candidate = (c, Node[j])
9:           Add candidate to Candidates
10:        end if
11:      end for
12:    end if
13:  end for
14:  Sort Candidates by its cost
15:  return Candidates
16: end function
```

```
17:
18: function GREEDYSOLVER
19:   Initialize N, the initial set of codes
20:    $\omega \leftarrow \{\}$ 
21:   Add to  $\omega$  the first node in N
22:   while  $\omega$  is not a solution do
23:     Candidates = getFeasibleLinks( $\omega$ )
24:     Update  $\omega$  with first node of Candidates
25:   end while
26:   return  $\omega$ 
27: end function
```



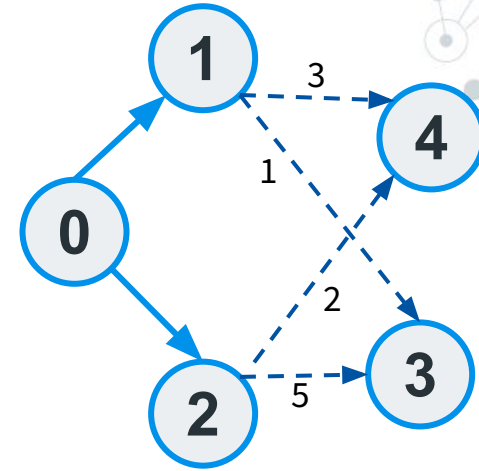
GRASP

Algorithm 2 GRASP

```

1: function GRASPSOLVER
2:   Initialize N, the initial set of codes
3:    $\omega \leftarrow \{\}$ 
4:   Add to  $\omega$  the first node in N
5:   while  $\omega$  is not a solution do
6:     Candidates = getFeasibleLinks( $\omega$ )
7:      $q_{min} \leftarrow \min\{q(c) \mid c \in \text{Candidates}\}$ 
8:      $q_{max} \leftarrow \max\{q(c) \mid c \in \text{Candidates}\}$ 
9:      $RCL_{min} \leftarrow \{c \in \text{Candidates} \mid q(c) \leq q_{min} + \alpha(q_{max} - q_{min})\}$ 
10:    Select  $c \in RCL$  at random
11:    Update  $\omega$  with random selected node form RCL
12:  end while
13:  return  $\omega$ 
14: end function

```



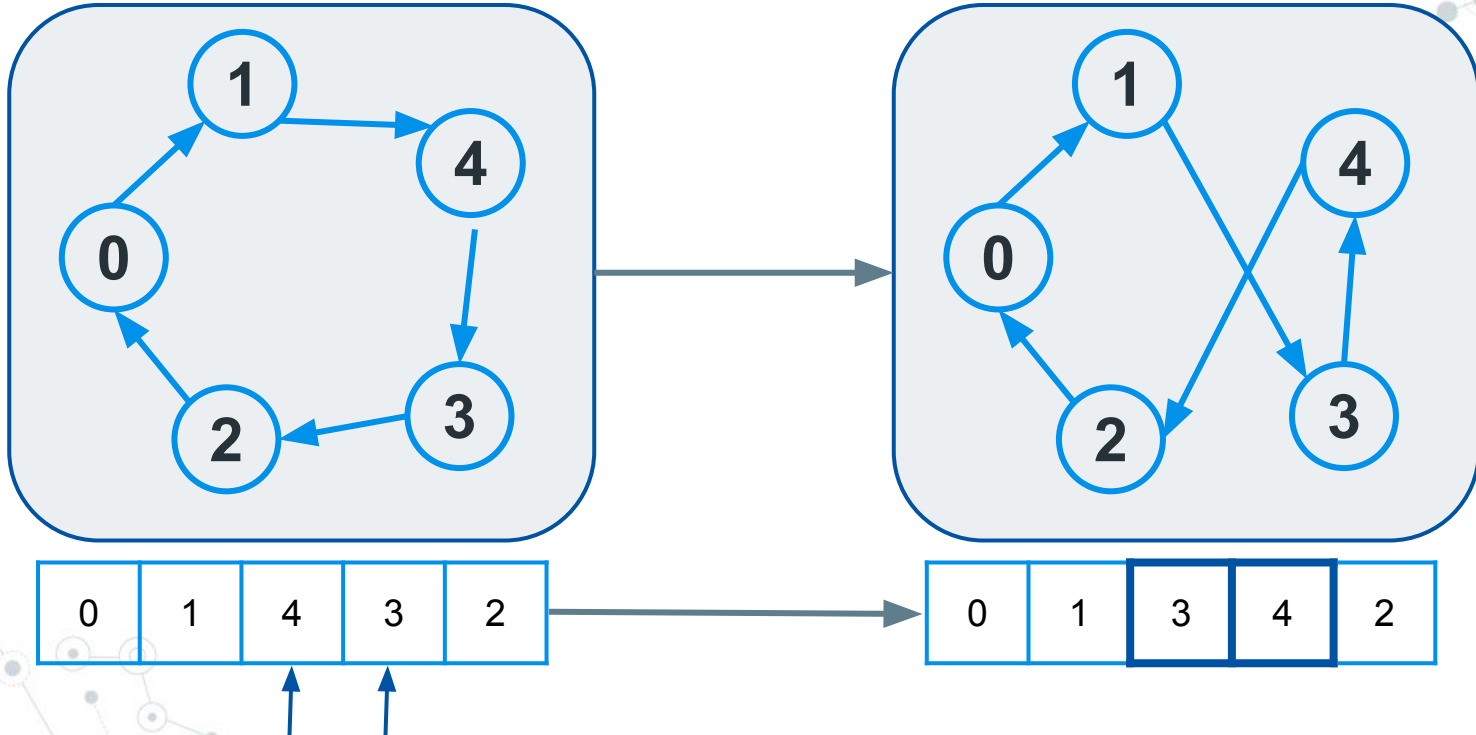
1	2	3	5
1	2	3	5

Given α of 0.7:
 $RCL_{min} = 1 + 0.7(5 - 1) = 3.8$



Local Search

2 Edge Exchange



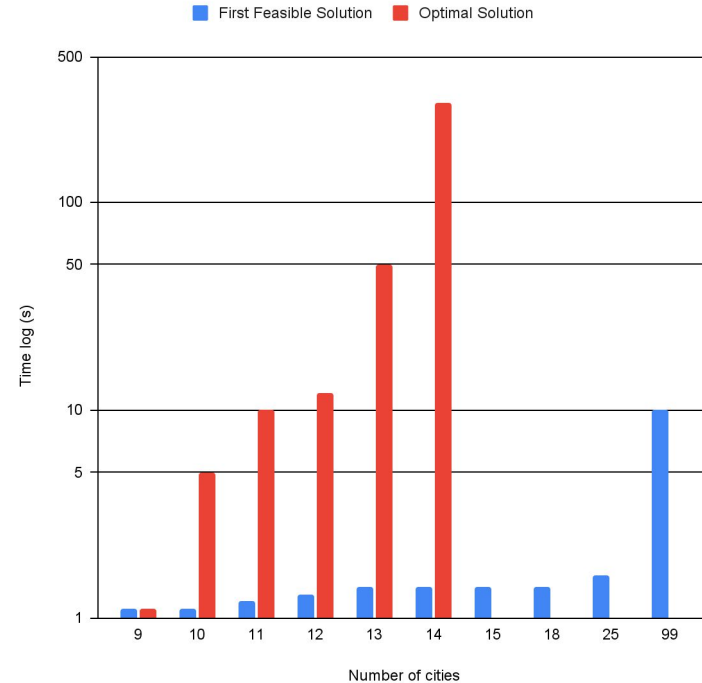
A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting different levels of connectivity or importance. The lines are thin and gray, creating a mesh-like structure.

4. Results

Cplex solving time

- Our Cplex model time grows exponentially
- With more than 15 codes our model is time infeasible

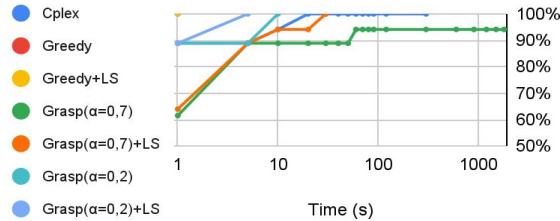
Cplex Solving time



Algorithm comparison up to 15 codes

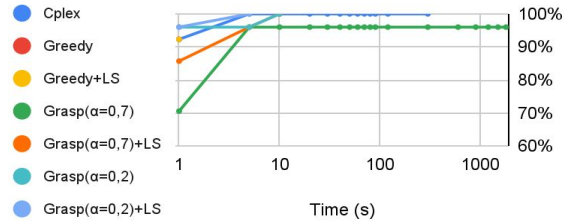
gap to optimal value

13 Codes. Higher values better



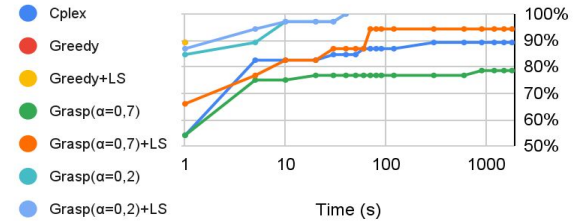
gap to optimal value

14 Codes. Higher values better



gap to best value

15 Codes. Higher values better

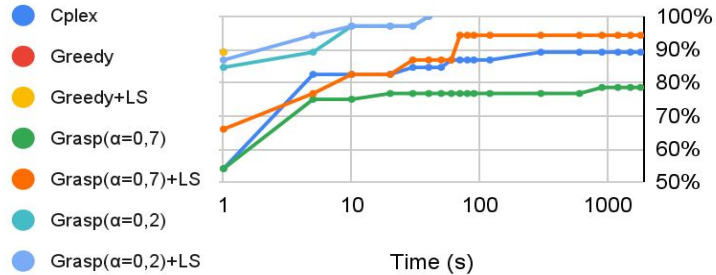


- Greedy algorithm gives good and fast solutions.
- After some tuning, lower α values give better results in GRASP.
- Combining Local Search with the Greedy and GRASP provides better results than his counterparts.

Algorithm comparison 25 and 99 codes

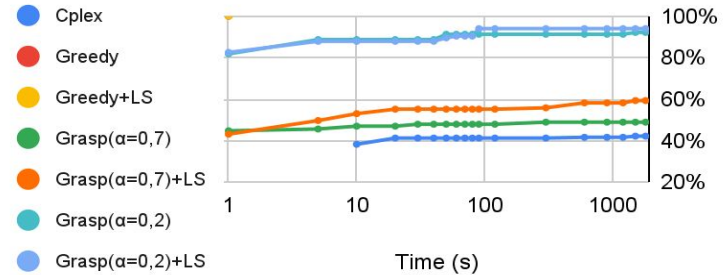
gap to best value

25 Codes. Higher values better



gap to best value

99 Codes. Higher values better



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines, with some nodes highlighted in blue.

5.

Conclusions & Future Work

Conclusions



Future work

- Grasp with increasing α value
- Try other local search algorithms (3-opt, Lin-Kernighan heuristic)
- Try other Cplex models



Thanks!

Any questions?