AMMM Final Project

Xin Tong and Qiuchi Chen

Universitat Politècnica de Catalunya
Facultat d'Informàtica de Barcelona

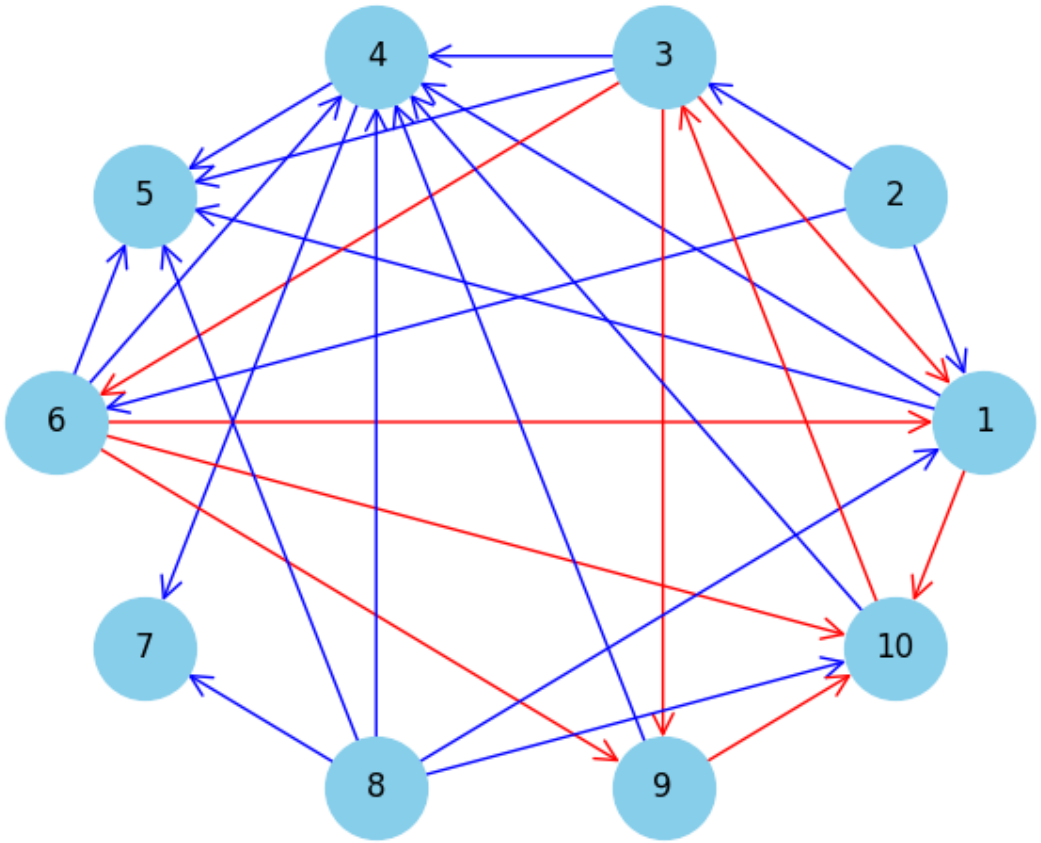
Advisors:
Prof. Enric Rodriguez and Prof. Luis Velasco

May 27, 2025

Table of Contents

Topological Sort

Cyclic graphs don't have valid topological orderings.



Edge	Indegree	Outdegree
2	0	3
8	0	5
3	2	5
6	2	5
7	2	0
9	2	2
1	4	3
10	4	2
5	5	0
4	6	2

ILP Model

Decision Variables

$$Priority_{ij}: \quad Binary \quad \forall i,j \in Members$$
$$O_n: \quad Integer \quad \forall n \in Members$$
$$Income = \sum_{i,j \epsilon Members} Priority_{ij} \cdot m_{ij}$$

Objective Function

$$\text{maximize} \quad Income$$

Constraints

$$Priority_{ij} + Priority_{ji} = 1$$
$$\forall \quad i,j \in Members \quad \text{and} \quad m_{ij} + m_{ji} \neq 0 \qquad (1)$$

$$O_i + 1 \leq O_j + (1 - Priority_{ij}) \cdot N$$
$$\forall \quad i,j \in Members \quad \text{and} \quad m_{ij} + m_{ji} \neq 0 \qquad (2)$$

Pseudo Code & Greedy Cost Function

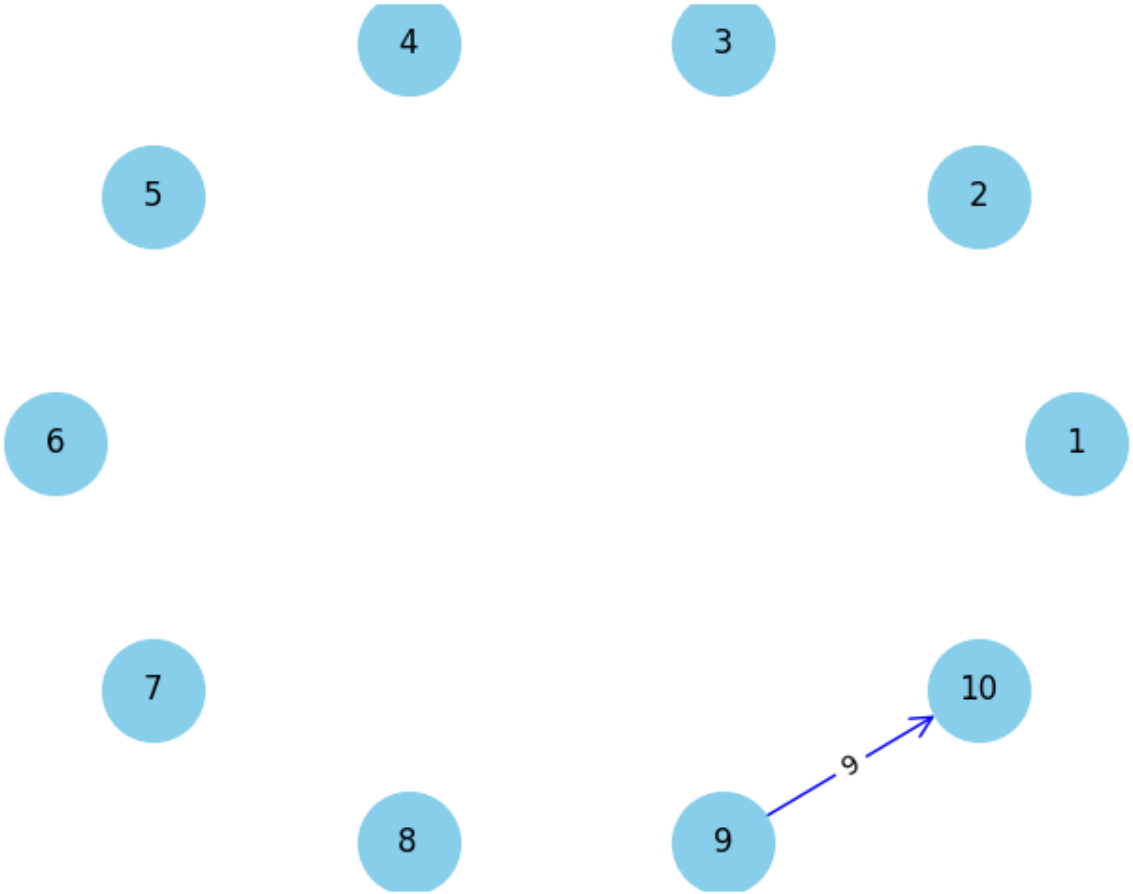
Greedy Cost Function

$$\text{q(<i,j>, S)} = \begin{cases} -1 & if\ G(S, < i, j >) \text{ is not a directed acyclic graph} \\ max(Bids_{ij}, Bids_{ji}) & if\ G(S, < i, j >) \text{ is a directed acyclic graph} \end{cases}$$

Algorithm Greedy Method

$Solution \leftarrow \phi$
$CandidatePairs \leftarrow \{< i, j > |\ i, j \in Members \quad \text{and} \quad Bids_{ij} + Bids_{ji} \neq 0\}$
while $CandidatePairs \neq \phi$ **do**
  Evaluate $q(< i', j' >, Solution) \quad \forall < i', j' > \in CandidatePairs$
  $< i, j > \leftarrow argmax\{q(< i', j' >, Solution) | < i', j' > \in CandidatePairs\}$
  $CandidatePairs \leftarrow CandidatePairs \setminus \{< i, j >, < j, i >\}$
  $Solution \leftarrow Solution \cup \{< i, j >\}$
return $Solution$

Greedy Constructive Phase (data=project.4.dat)



Solution	Excluded	Candidates	Objective
9->10: 9	10->9: 9	9->10: 9	9
		10->9: 9	
		6->9: 9	
		10->4: 9	
		8->1: 9	
		1->10: 9	
		4->10: 8	
		9->6: 8	
		4->5: 8	
		5->4: 8	
		1->5: 8	
		3->1: 8	
		1->8: 7	
		5->1: 7	
		8->4: 7	
		8->10: 7	
		10->3: 7	
		1->3: 6	
		3->5: 6	
		5->6: 6	
		2->3: 6	
		4->7: 6	
		3->4: 6	
		10->1: 5	
		4->8: 5	
		5->3: 5	
		6->5: 5	

Local Search Method

Neighborhood

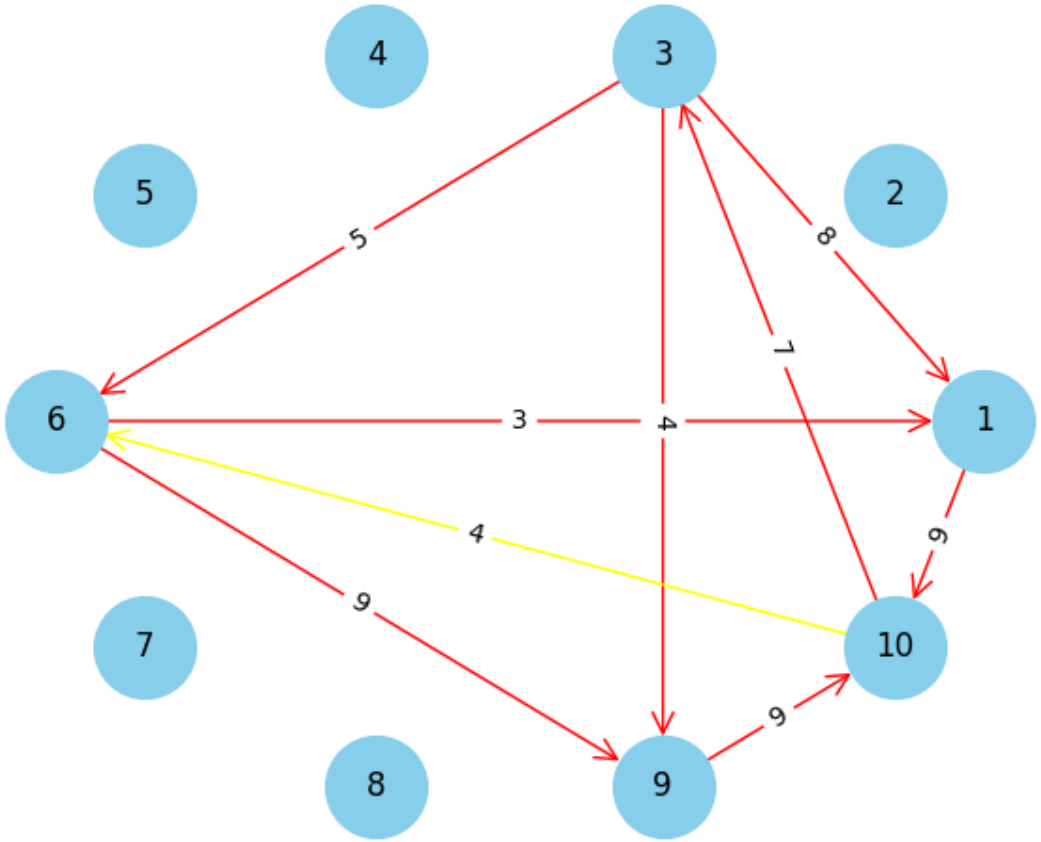
$\{$<i,j> $|< j, i >\in solution$ and $Bids_{ij} > Bids_{ji}\}$

Knockon Flipped Pairs Cost Function

$q(< i', j' >, edges^{flip}) =$

$$\begin{cases} -1 \\ \quad \text{if} \quad Bids_{i'j'} - Bids_{j'i'} + \sum_{<i'',j''>\in edges^{flip}} Bids_{i''j''} - Bidsj''i'' \geq Bids_{ij} - Bids_{ji} \\ \quad \text{or } edges^{residual} \mid edges^{flip} = edges^{residual} \mid edges^{flip} \cup \{< i', j' >\}) \\ \sum_{k'\in\{i',j'\}} indegree(k') + outdegree(k') \\ \quad \text{if} \quad Bids_{i'j'} - Bids_{j'i'} + \sum_{<i'',j''>\in edges^{flip}} Bids_{i''j''} - Bidsj''i'' < Bids_{ij} - Bids_{ji} \\ \quad \text{and } edges^{residual} \mid edges^{flip} > edges^{residual} \mid edges^{flip} \cup \{< i', j' >\} \end{cases}$$

Knockon Pairs Selection Process



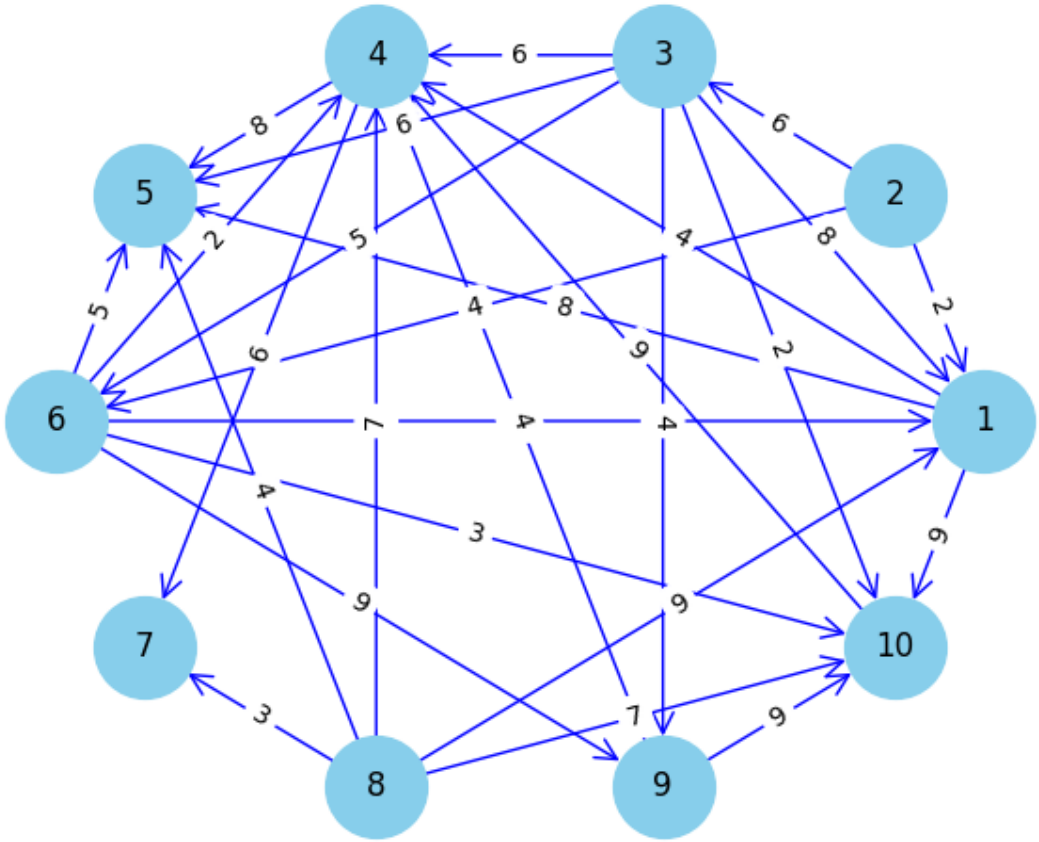
Chosen	Cand.	Degree	Impact
	6->10	8	1
	3->6	8	0
	3->9	7	1
	9->10	7	0
	6->9	7	-1
	3->1	7	-2
	6->1	7	-2
	1->10	7	-4

Pseudo Code

Algorithm Local Search Method

$solution^{current} \leftarrow doGreedyConstructionPhase()$
$improved \leftarrow$ True
while $improved =$ True **do**
  $improved \leftarrow$ False
  for each $< i, j > \in \{< i, j > | < j, i > \in solution^{current}$ and $bids_{ij} > bids_{ji}\}$ **do**
    $edges^{residual} \leftarrow doTopologicalSort()$
    $edges^{candidate}, edges^{flip} \leftarrow edges^{residual}, \phi$
    while $edges^{residual} \neq \phi$ **do**
      Evaluate $q(< i', j' >, edges^{flip}) \quad \forall < i', j' > \in edges^{candidate}$
      $< i'', j'' > \leftarrow argmax\{q(< i', j' >, edges^{flip}) \mid < i', j' > \in edges^{candidate}\}$
      $edges^{residual} \leftarrow doTopologicalSort()$
      $edges^{candidate} \leftarrow edges^{residual}$
      $edges^{flip} \leftarrow edges^{flip} \cup \{< i'', j'' >\}$
    $solution^{flip} \leftarrow solution^{current}$
    Swap $< j', i' >$ with $< i', j' >$ in $solution^{flip} \quad \forall < i', j' > \in edges^{flip}$
    if $G(solution^{flip})$ is a DAG **then**
      $improved \leftarrow$ True
      $solution^{current} \leftarrow solution^{flip}$
return $solution^{current}$

Local Search Phase (data=project.4.dat)



Solution	Excluded	Degree	Impact	Objective
9->10: 9	10->9: 9			9
6->9: 9	9->6: 8			18
10->4: 9	4->10: 8			27
8->1: 9	1->8: 7			36
1->10: 9	10->1: 5			45
4->5: 8	5->4: 8			53
1->5: 8	5->1: 7			61
3->1: 8	1->3: 6			69
8->4: 7	4->8: 5			76
8->10: 7	10->8: 2			83
3->10: 2	10->3: 7			85
3->5: 6	5->3: 5			91
6->5: 5	5->6: 6			96
2->3: 6	3->2: 4			102
4->7: 6	7->4: 3			108
3->4: 6	4->3: 1			114
3->6: 5	6->3: 5			119
1->4: 4	4->1: 5			123
3->9: 4	9->3: 5			127
6->4: 2	4->6: 5			129
2->6: 4	6->2: 4			133
9->4: 4	4->9: 3			137
6->10: 3	10->6: 4			140
8->5: 4	5->8: 1			144
2->1: 2	1->2: 3			146
8->7: 3	7->8: 2			149
6->1: 3	1->6: 1			152

Pseudo Code & RCL

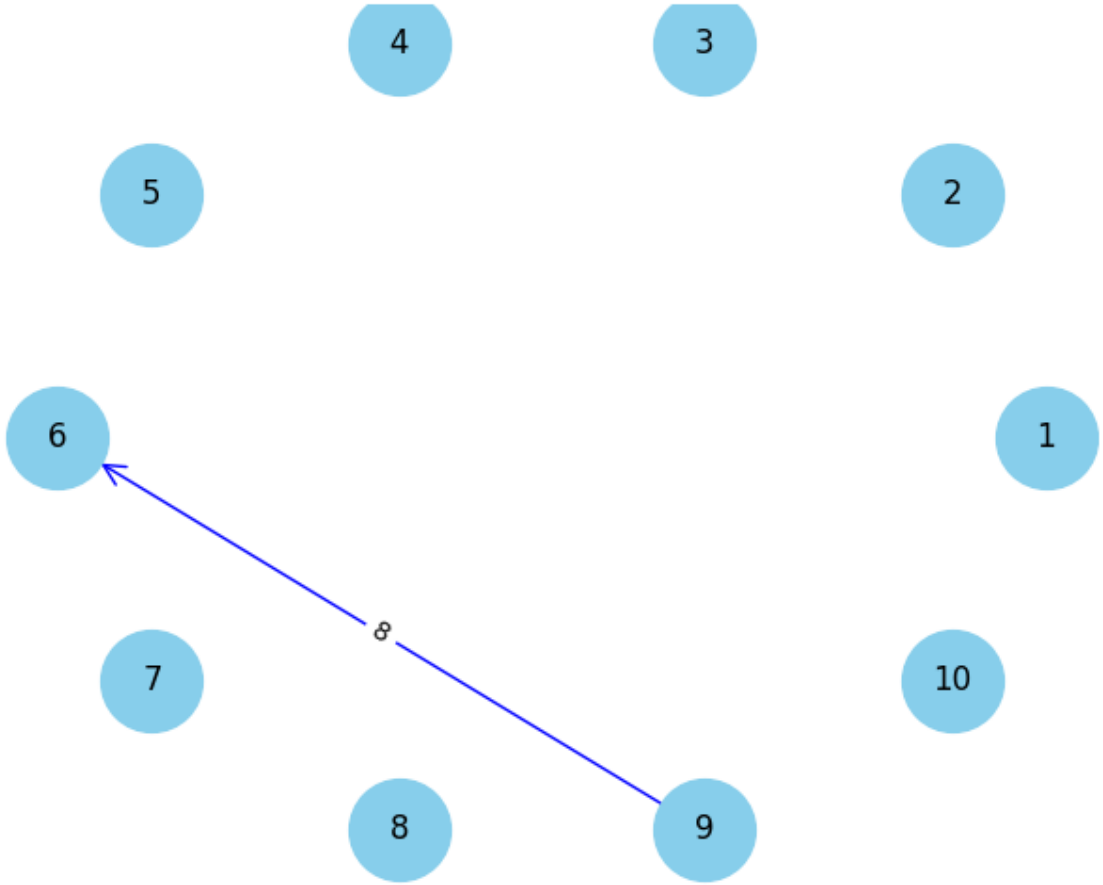
Algorithm GRASP Constructive Phase

$solution \leftarrow \phi$
$edges^{candidate} \leftarrow \{< i, j > |\ i, j \in Members \quad \text{and} \quad Bids_{ij} + Bids_{ji} \neq 0\}$
while $edges^{candidate} \neq \phi$ **do**
  Evaluate $q(< i', j' >, solution) \quad \forall < i', j' > \in edges^{candidate}$
  $edges^{candidate} \leftarrow \{< i', j' > \in edges^{candidate} \mid q(< i', j' >, solution) > 0\}$
  $q^{min} \leftarrow \min\{q(< i', j' >, solution) \mid < i', j' > \in edges^{candidate}\}$ ▷ RCL Procedure
  $q^{max} \leftarrow \max\{q(< i', j' >, solution) \mid < i', j' > \in edges^{candidate}\}$
  $\text{RCL}_{max} \leftarrow \{< i', j' > \in edges^{candidate} \mid q(< i', j' >, solution) \geq q^{max} - \alpha(q^{max} - q^{min})\}$
  $< i, j > \leftarrow$ select $< i', j' > \in$ RCL at random
  $edges^{candidate} \leftarrow edges^{candidate} \setminus \{< i, j >, < j, i >\}$
  $solution \leftarrow solution \cup \{< i, j >\}$
return $solution$

Algorithm GRASP Procedure

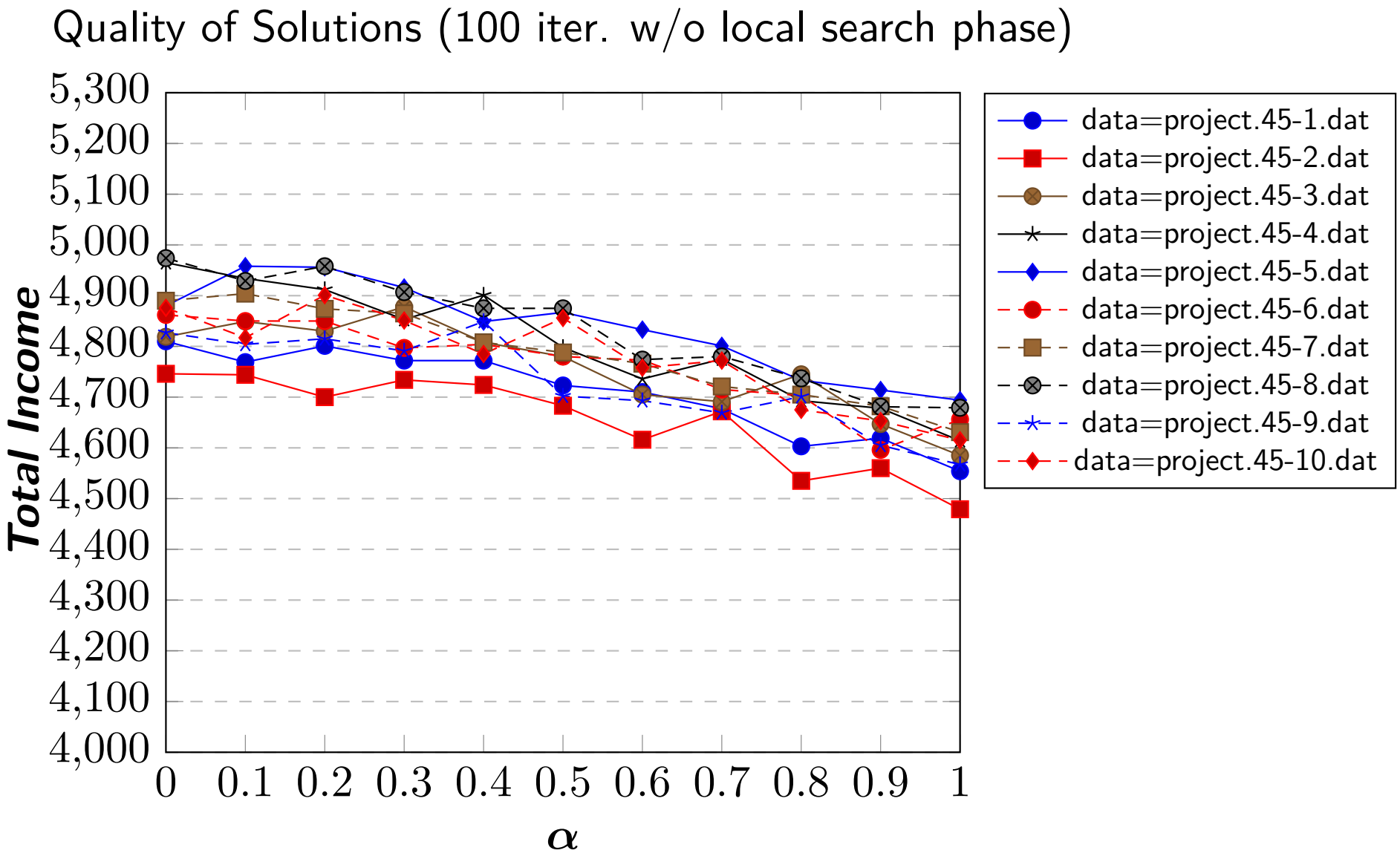
$object^{best}, solution^{best} \leftarrow 0, \phi$
for $retry = 1$ to MaxIterations **do**
  $obejctive, solution \leftarrow 0, \phi$
  $solution \leftarrow doConstructivePhase()$
  $solution \leftarrow doLocalSearchPhase(solution)$
  if $obejctive > objective^{best}$ **then**
    $objective^{best} \leftarrow obejctive$
    $solution^{best} \leftarrow solution$
return $solution^{best}$

GRASP Constructive Phase (data=project.4.dat,$\alpha = 0.2$)

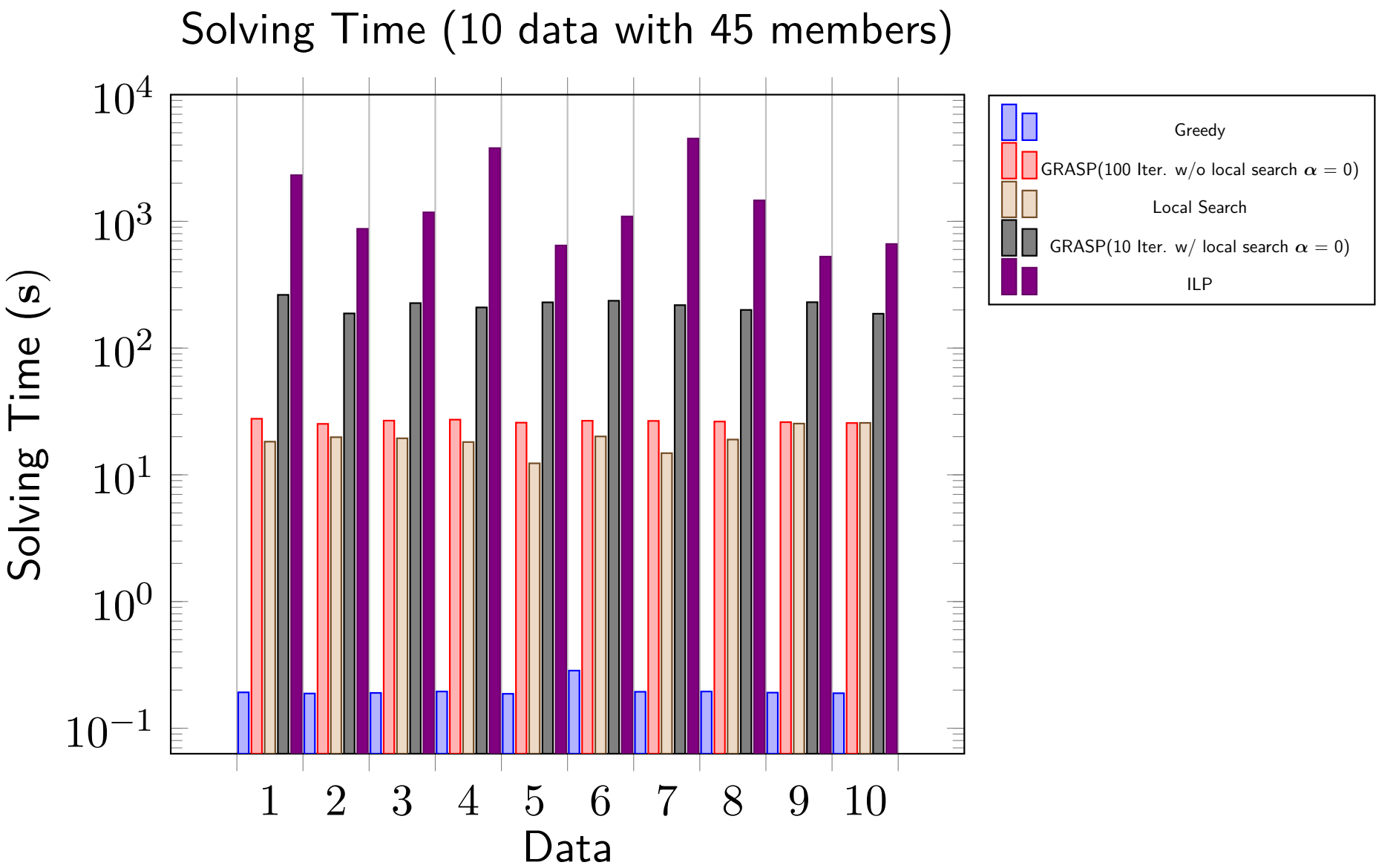


Solution	Excluded	RCL	Objective
9->6: 8	6->9: 9	9->10: 9	8
		10->9: 9	
		6->9: 9	
		10->4: 9	
		8->1: 9	
		1->10: 9	
		4->10: 8	
		9->6: 8	
		4->5: 8	
		5->4: 8	
		1->5: 8	
		3->1: 8	
		1->8: 7	
		5->1: 7	
		8->4: 7	
		8->10: 7	
		10->3: 7	
		1->3: 6	
		3->5: 6	
		5->6: 6	
		2->3: 6	
		4->7: 6	
		3->4: 6	
		10->1: 5	
		4->8: 5	
		5->3: 5	
		6->5: 5	

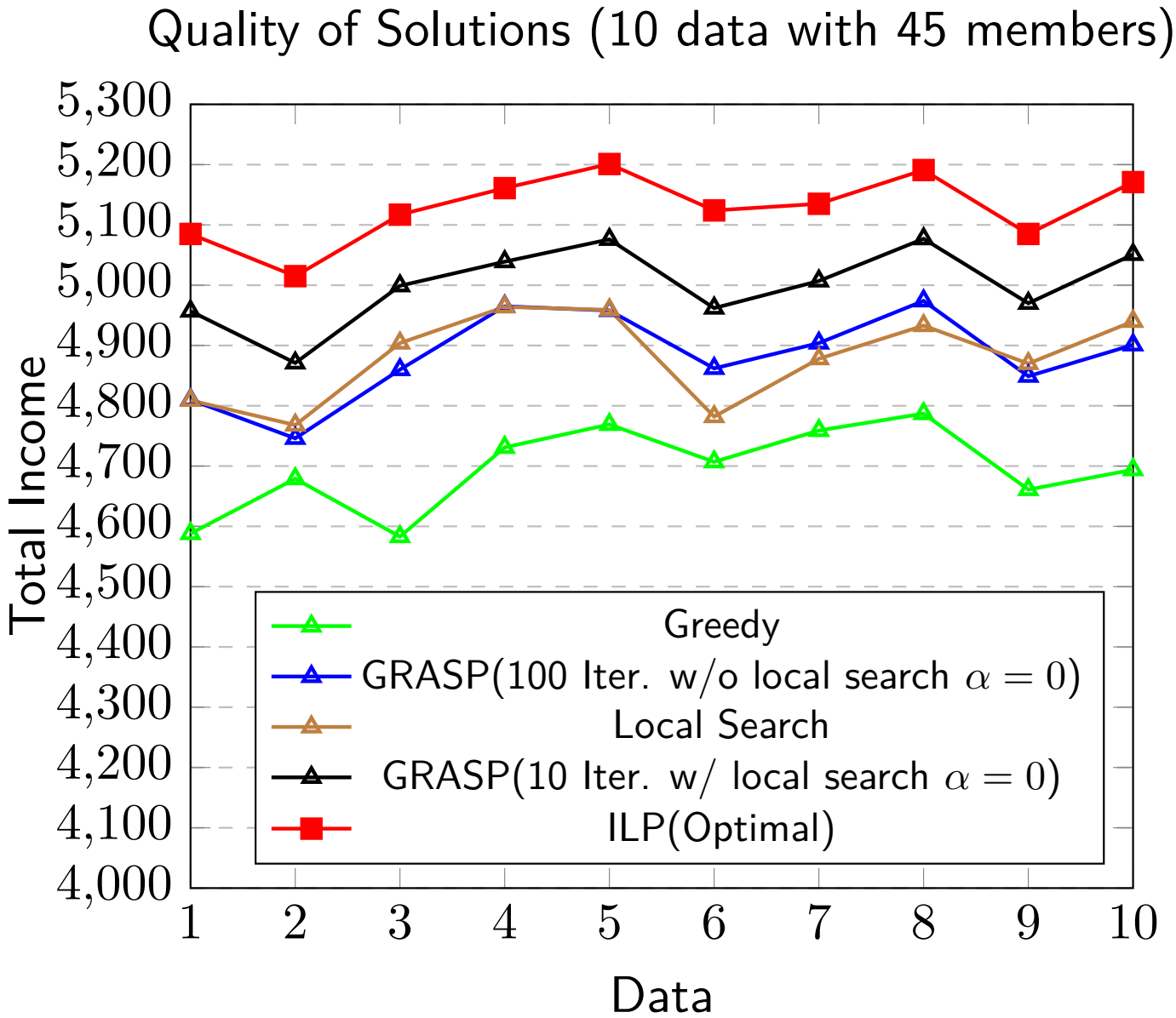
α Tuning for GRASP



Solving Time



Solution Quality



Thanks