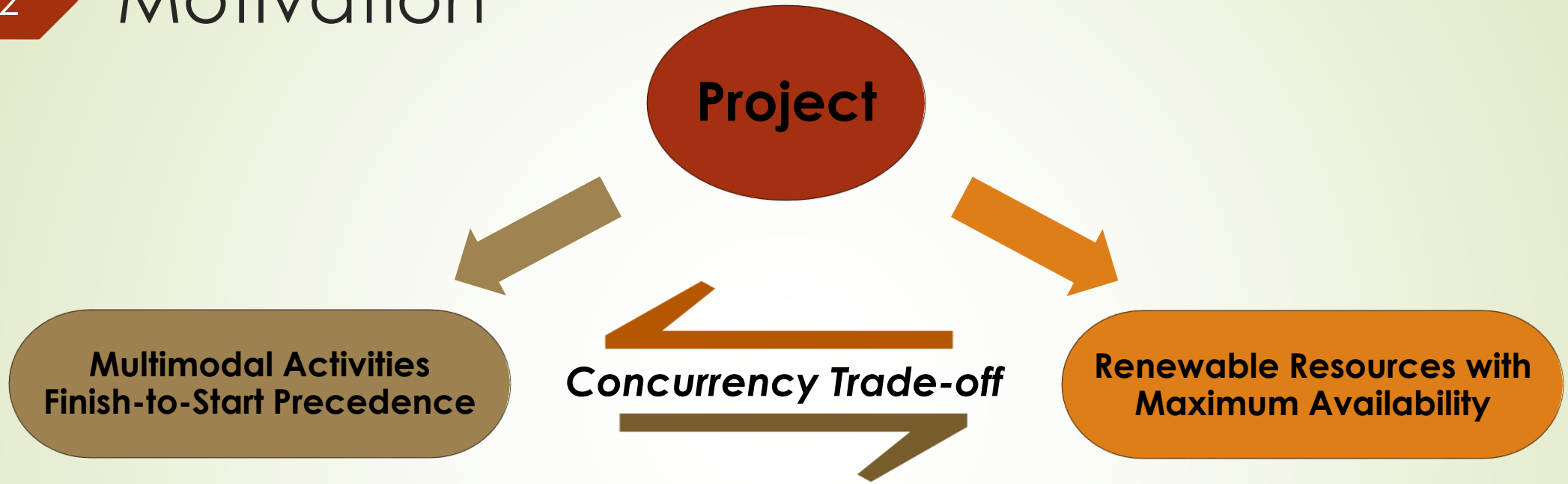
WORLD CIST 15
April 1-3, Azores, Portugal

University of Minho
School of Engineering

Concurrency Detection on Finish-to-Start Activity Precedence Networks

Anabela Tereso (Department of Production and Systems)
Rui Moutinho (Masters Degree on Informatics)

Motivation



- Allows concurrency
- Resource requirement

- Fixed availability (at all times)
- Fully released after activity completion

To assess the impact of the availability constraint

Concurrency Detection Method

Collection

- Precedence relation

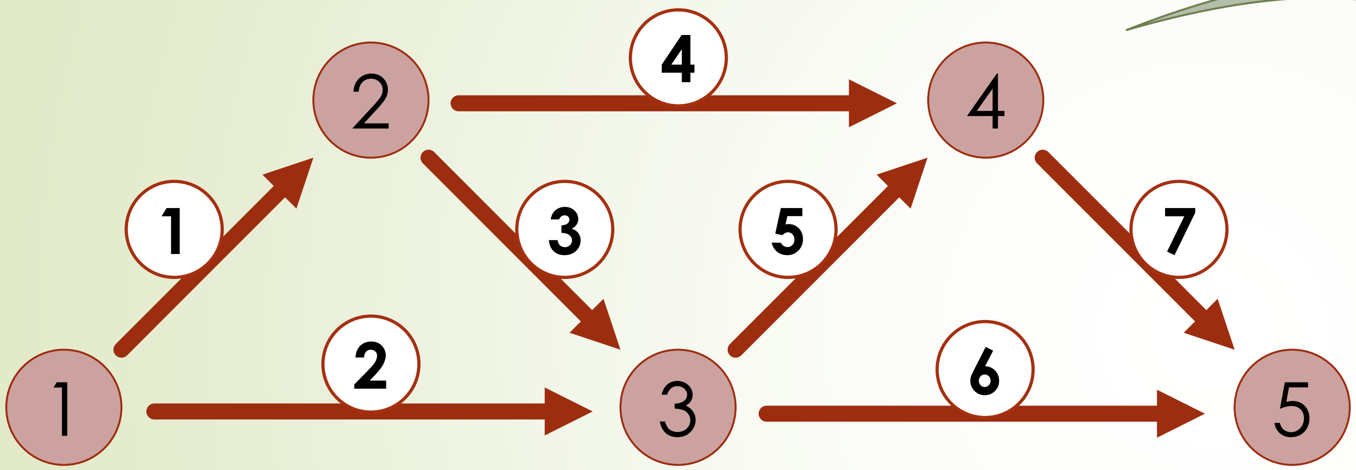
Transformation

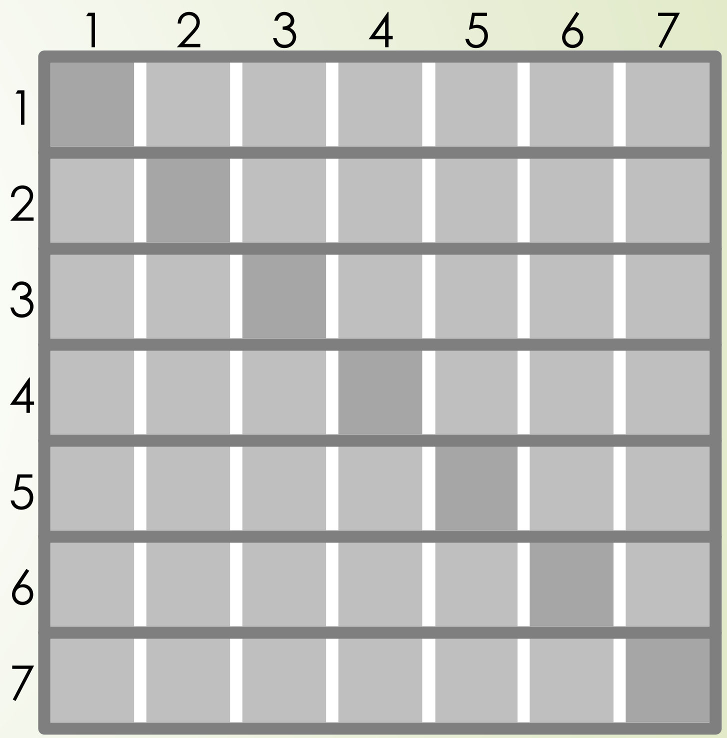
- Parallel relation

Identification

- Sets of project activities that can execute concurrently with each other

Detection Method: Collection





Finish-to-Start Precedence (matrix form)

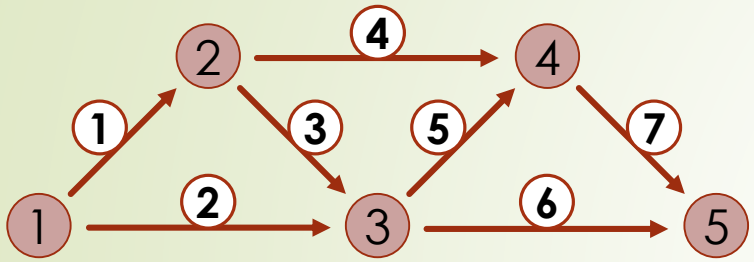
Transitive Precedence (matrix form)

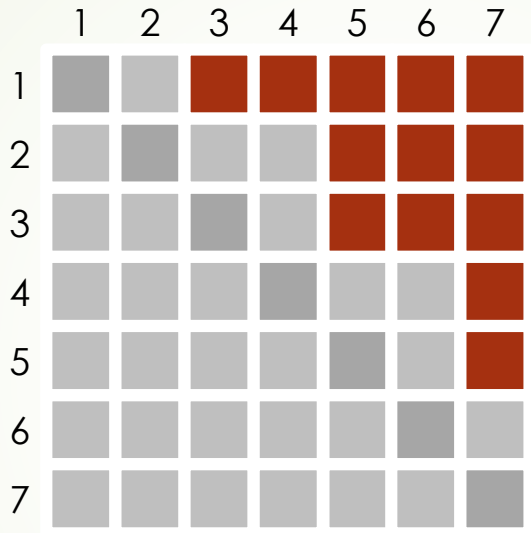
$row \times column =$ ■ reads:
"row" must precede "$column$"

Finish-to-Start Activity Precedence Relation
usually specified through either AoA or AoN networks
(here, in AoA: 7 arcs – 7 activities)

Intuitively **Transitive**

Detection Method: Transformation

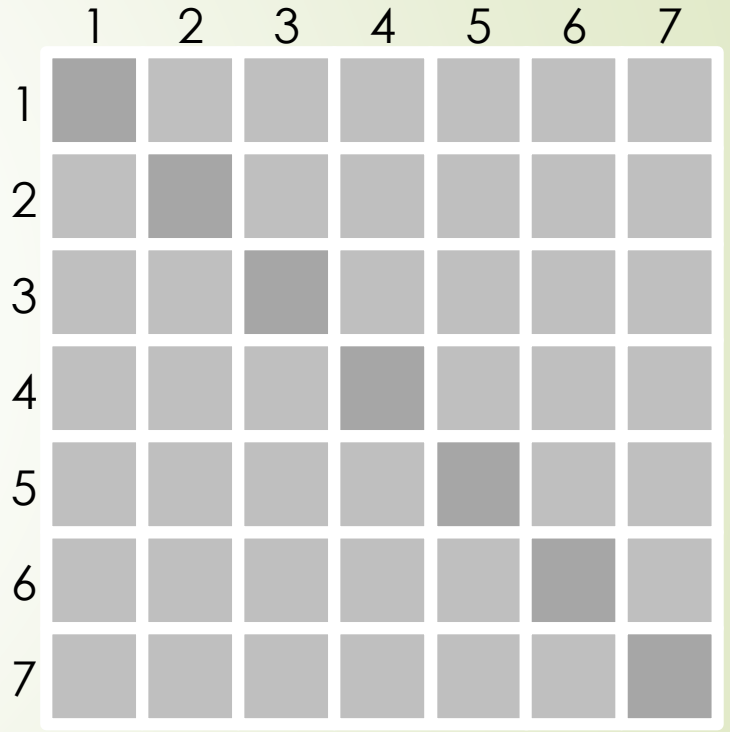




Transitive Precedence Relation

$row \times column =$ ■ means
"row" ***must occur before*** "$column$"

If *neither* activity a or b *precedes the other* they can execute *concurrently*



Parallel Relation (matrix form)

$row \times column =$ ■ means
"row" can execute in parallel with "$column$"

Detection Method: Identification

Parallel Relation Matrix → ***adjacency*** → Simple Undirected Graph

Set of activities, such:

Any two different activities may execute in parallel

+

There is no activity outside the set which can occur in parallel to any from the set

↓

The largest possible set whose activities can execute in parallel

Parallel Set

Set of nodes, such:

Any two different nodes are directly connected

+

There is no node outside the set which is directly connected to all in the set

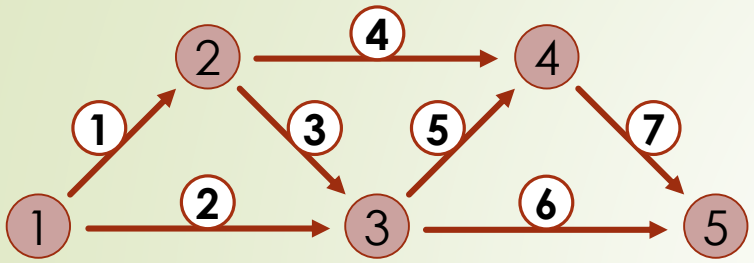
↓

The largest possible set of nodes which are connected to each other

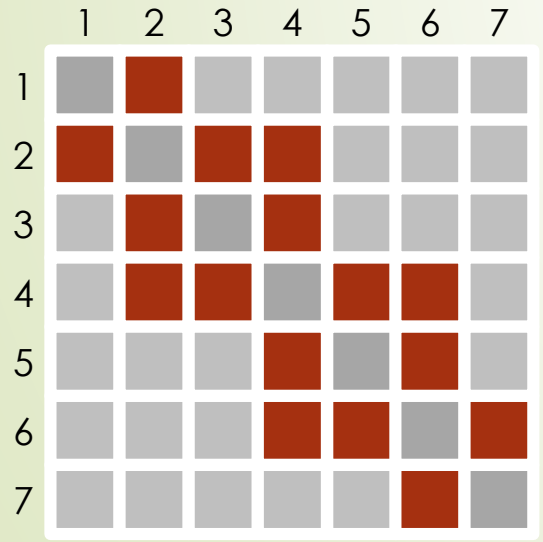
Maximal Clique

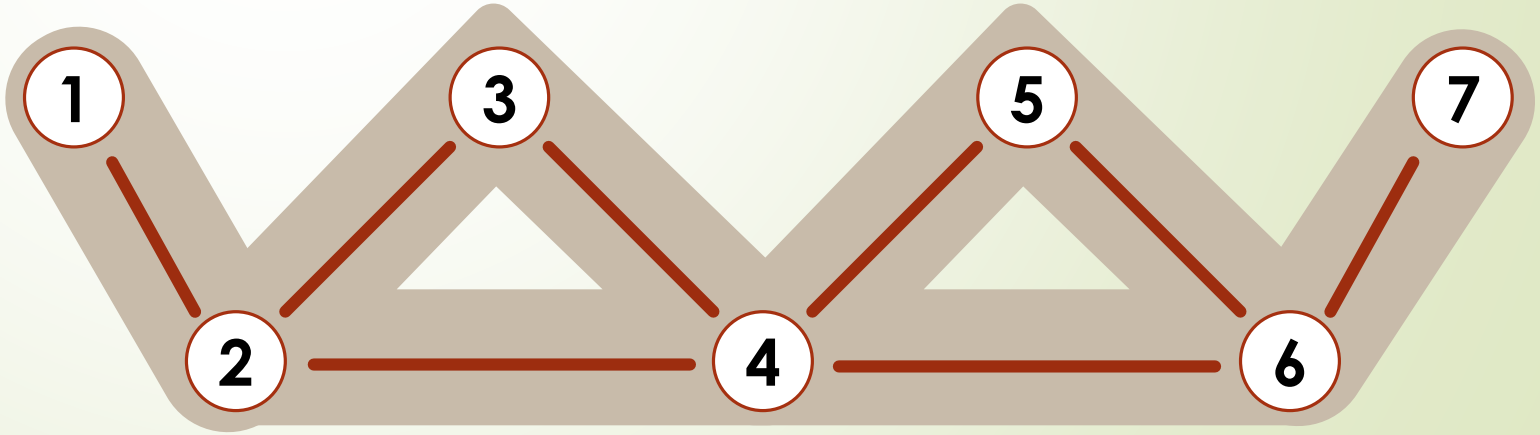
All Parallel Sets ⇋ All Maximal Cliques

Detection Method: Identification

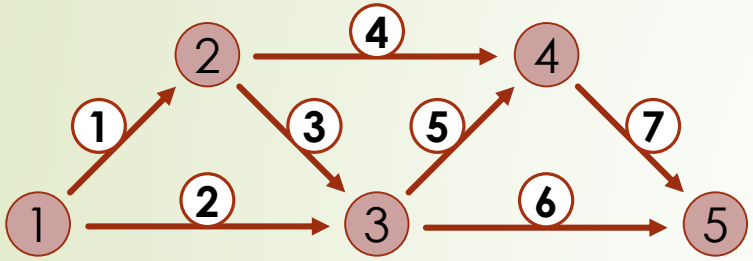


Parallel Sets:	
{1, 2}	{4, 5, 6}
{2, 3, 4}	{6, 7}



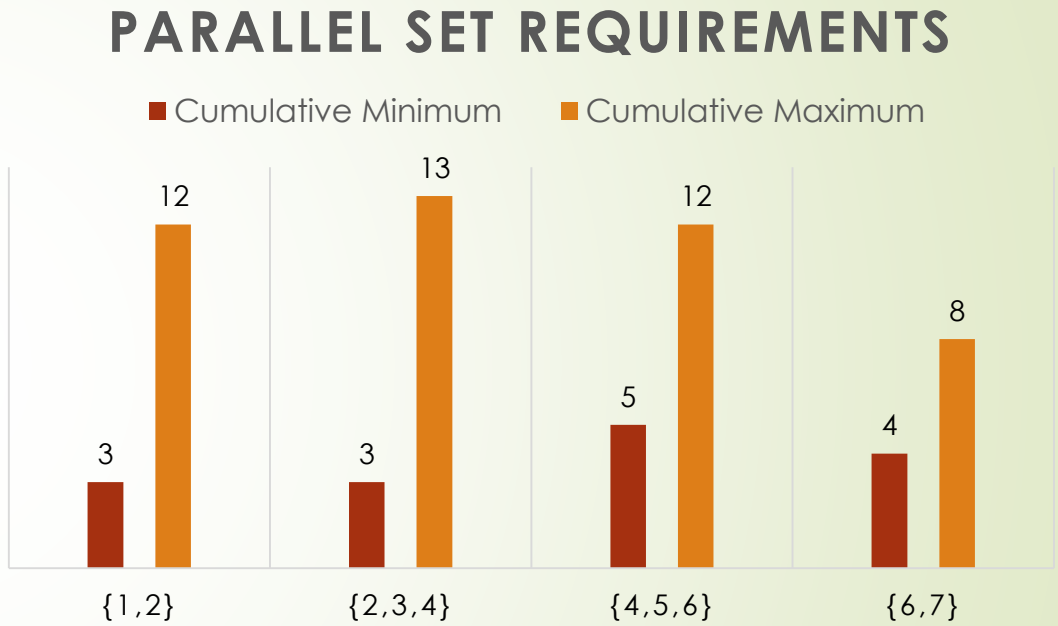


Application: Resource Availability Constraint



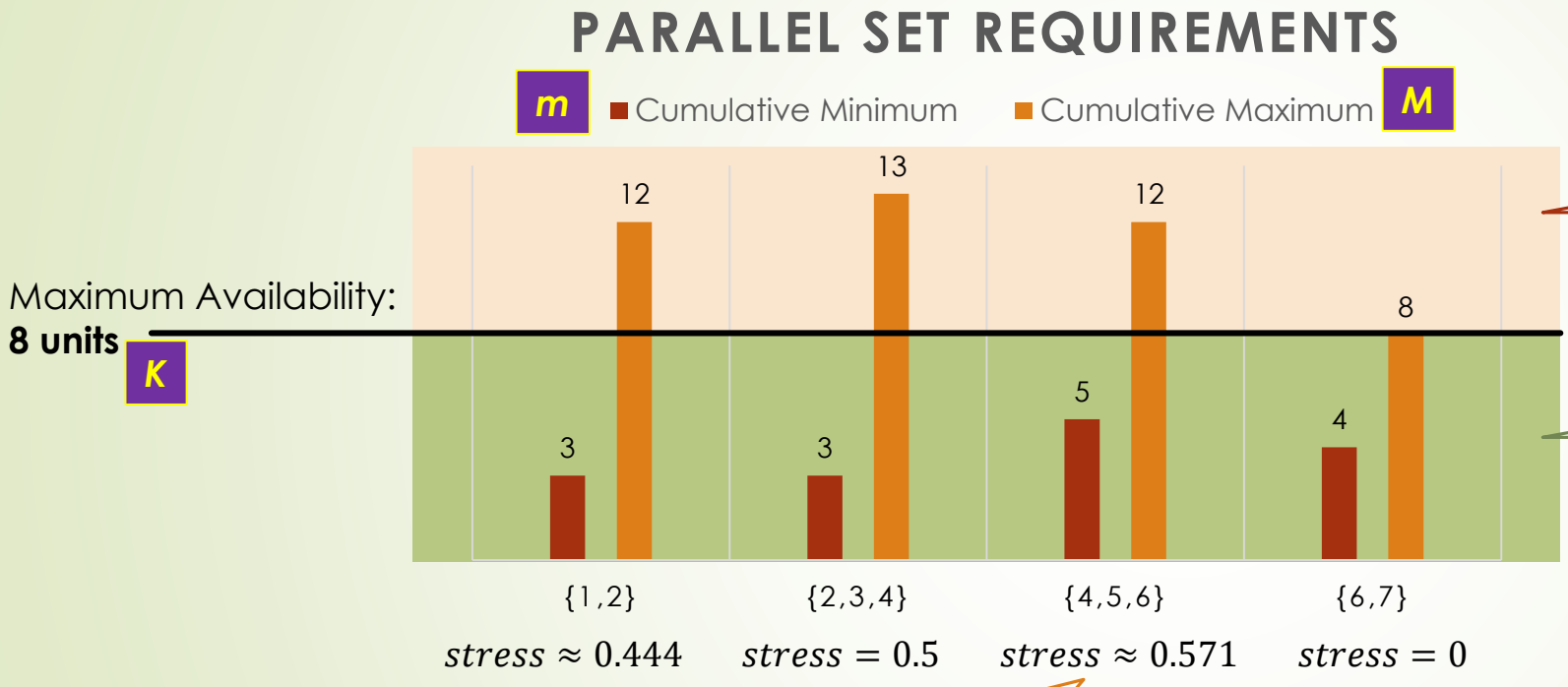
Multimodal Activities:

1. Min = 2; Max = 7
2. Min = 1; Max = 5
3. Min = 1; Max = 4
4. Min = 1; Max = 4
5. Min = 2; Max = 3
6. Min = 2; Max = 5
7. Min = 2; Max = 3



- With 13 units or more, no resource constraint is imposed
- With 5 units or more, any set is supported
- With less than 3 units, no concurrency will be possible

Application: Resource Stress



Availability **prohibits** concurrency scenarios

Availability **allows any** concurrency scenarios

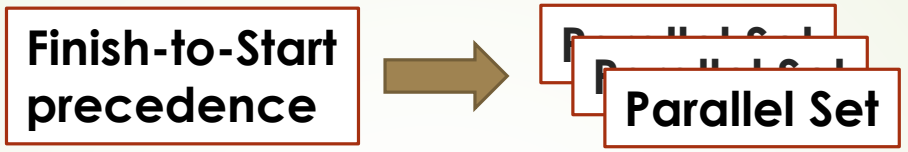
The maximu $\frac{8-8}{8-4} = 0$ .571 57%

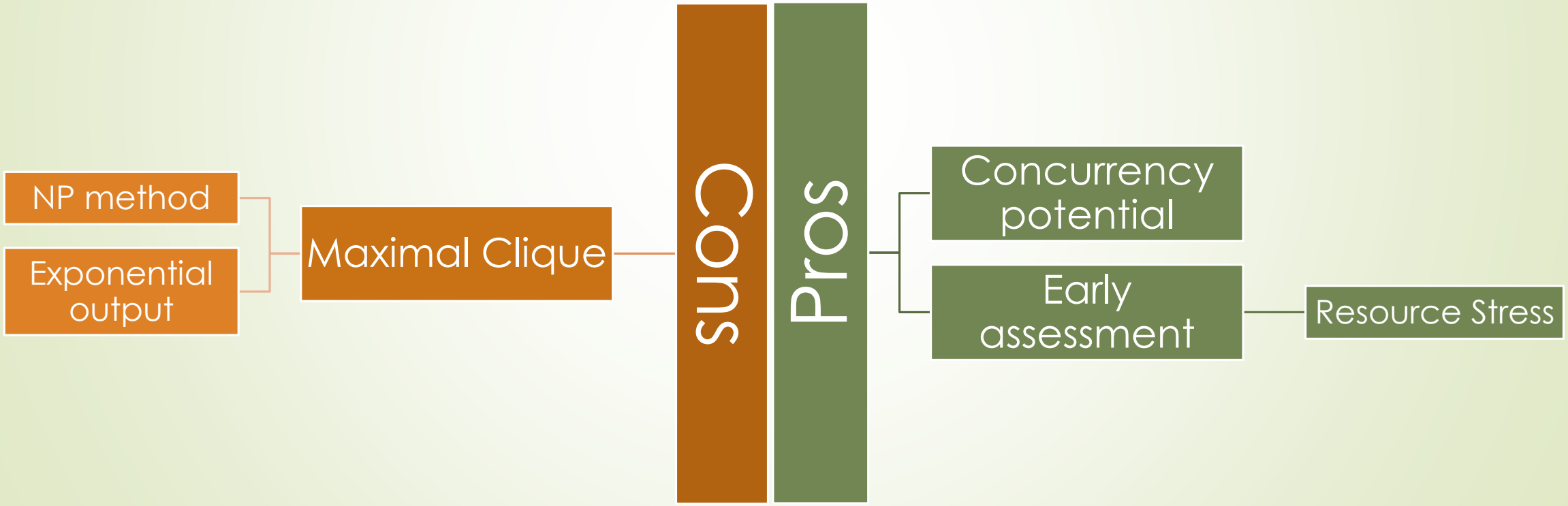
$$stress = \begin{cases} \frac{M-K}{M-m} & m \leq K \leq M \\ 0 & K < M \\ 1 & K < m \end{cases}$$

Resource Stress (of a parallel set)

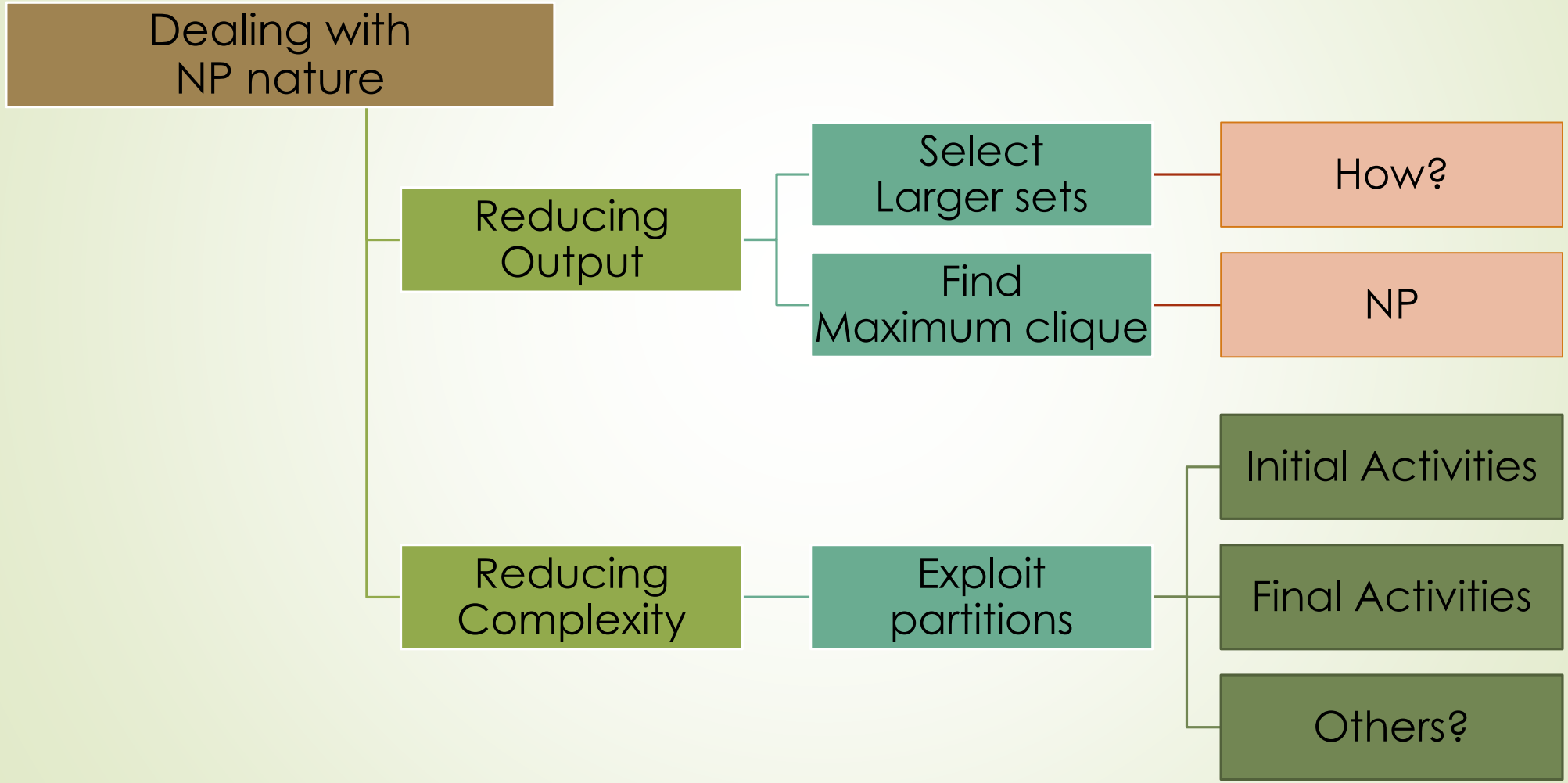
The probability of the cumulative requirement being greater than the maximum availability

Conclusions & Future Research





Conclusions & Future Research



References

- Bron, C., Kerbosch, J.: Algorithm 457: Finding all cliques of an undirected graph. Commun. ACM 16(9), 575–577 (Sep 1973)
- Cazals, F., Karande, C.: A note on the problem of reporting maximal cliques. Theoretical Computer Science 407(13), 564 – 568 (2008)
- Chang, L., Yu, J., Qin, L.: Fast maximal cliques enumeration in sparse graphs. Algorithmica 66(1), 173–186 (2013)
- Demeulemeester, E.L., Herroelen, W.S.: Project Scheduling - A Research Handbook, International Series in Operations Research & Management Science, vol. 49. Kluwer Academic Publishers (2002)
- Moutinho, R., Tereso, A.: Scheduling of multimodal activities with multiple renewable and availability constrained resources under stochastic conditions. Procedia Technology 16(0), 1106 – 1115 (2014)
- Pattabiraman, B., Patwary, M., Gebremedhin, A., Liao, W.k., Choudhary, A.: Fast algorithms for the maximum clique problem on massive sparse graphs. In: Bonato, A., Mitzenmacher, M., Praat, P. (eds.) Algorithms and Models for the Web Graph, Lecture Notes in Computer Science, vol. 8305, pp. 156–169. Springer International Publishing (2013)
- Prosser, P.: Exact algorithms for maximum clique: A computational study. Algorithms 5(4), 545–587 (2012)
- Tomita, E., Tanaka, A., Takahashi, H.: The worst-case time complexity for generating all maximal cliques and computational experiments. Theoretical Computer Science 363(1), 28 – 42 (2006), Computing and Combinatorics 10th Annual International Conference on Computing and Combinatorics (COCOON 2004)