Mixed Multi-Unit Combinatorial Auctions for Supply Chain Automation

Andrea Giovannucci
Meritxell Vinyals
Jesus Cerquides
Ulle Endriss
Juan Antonio Rodriguez-Aguilar
Pedro Meseguer

Institut d'Investigació en
Intel.ligència Artificial
(IIIA-CSIC)

UNIVERSITAT DE BARCELONA

Outline

- Motivation
- Background (MMUCA)
- Limitations of WD solvers for MMUCA
- The Improved Solver
- Empirical evaluation
- Future work

Motivations

- The organisational structure of enterprises is changing
- Increment of outsourced activity
- From monolithic to collaborative structures that tend to reduce their size

Chinese Motorbike Industry

- Small firms meet in online places and coffee shops
- Each one is assigned the task it is best at
- A self-organising system of design and production

Background

- Business partners are moving from the roles of suppliers, manufacturers, and customers to the role of collaborators
- In this environment, the choice of the best business partners is critical

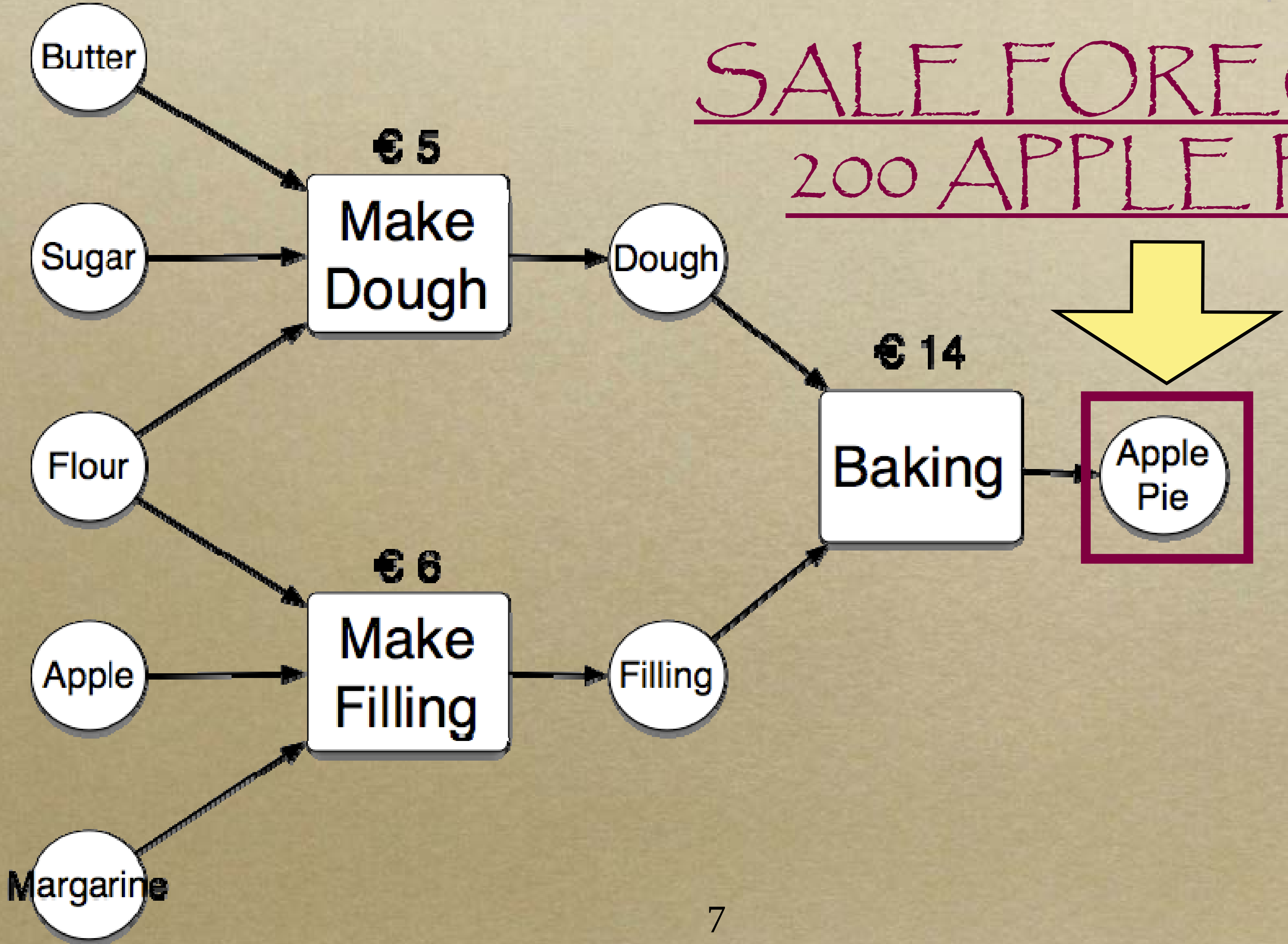
Goals

- Design a selection and coordination process among multiple partners so that:
  - it is easy to automate
  - it meets particular production requirements
  - it optimises production costs

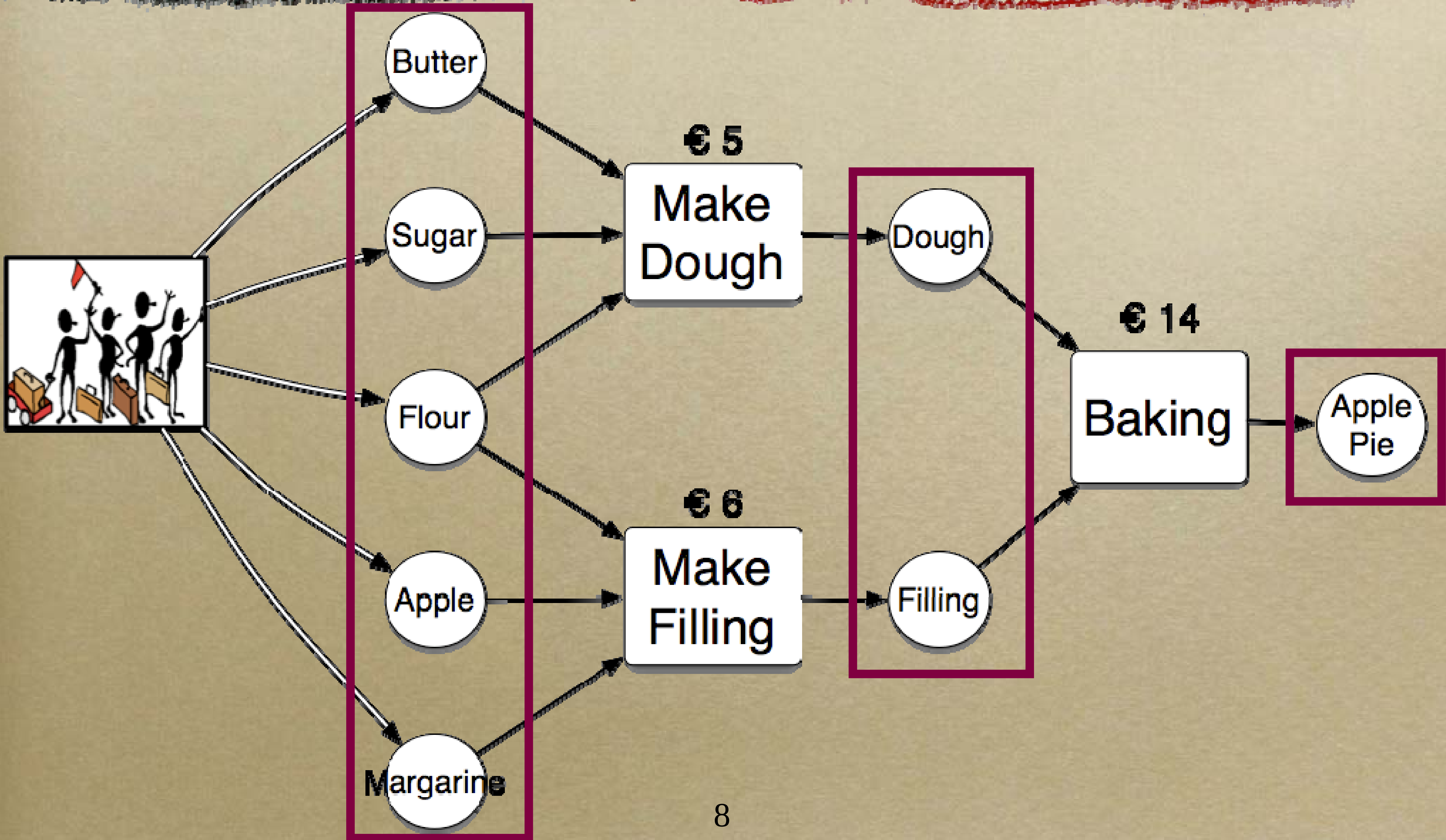
Example

SALE FORECAST

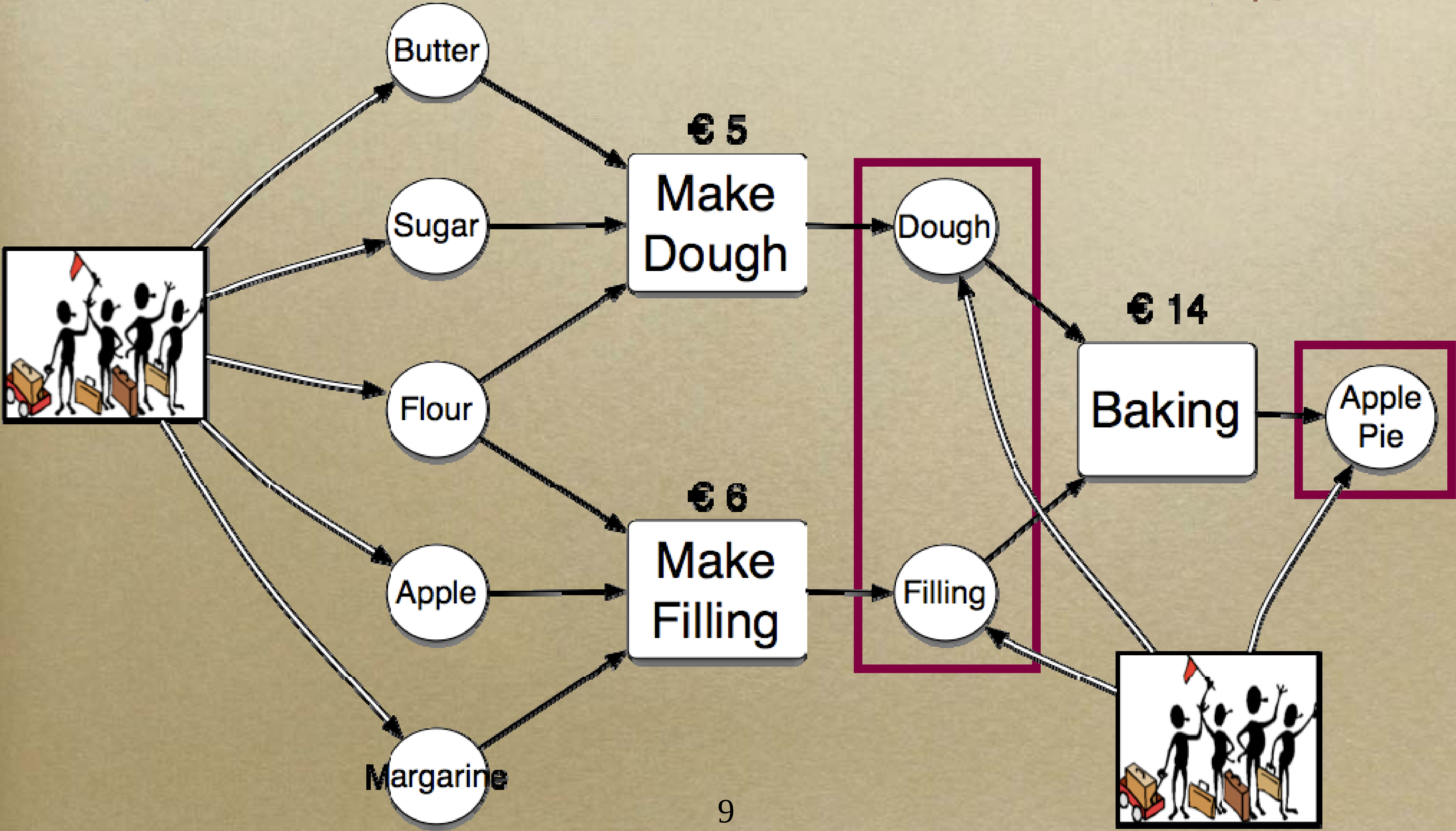
200 APPLE PIES



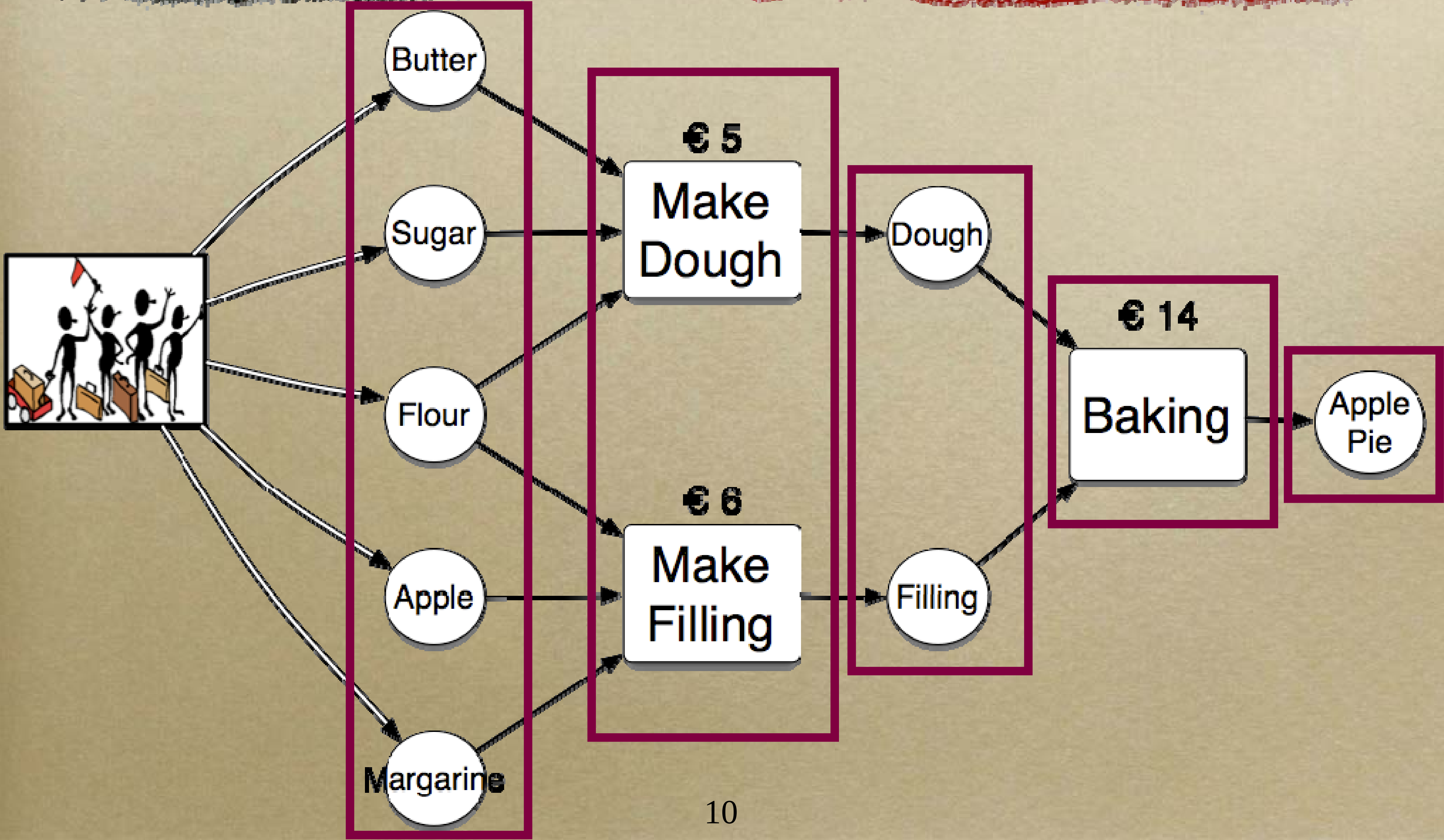
Procurement Stage



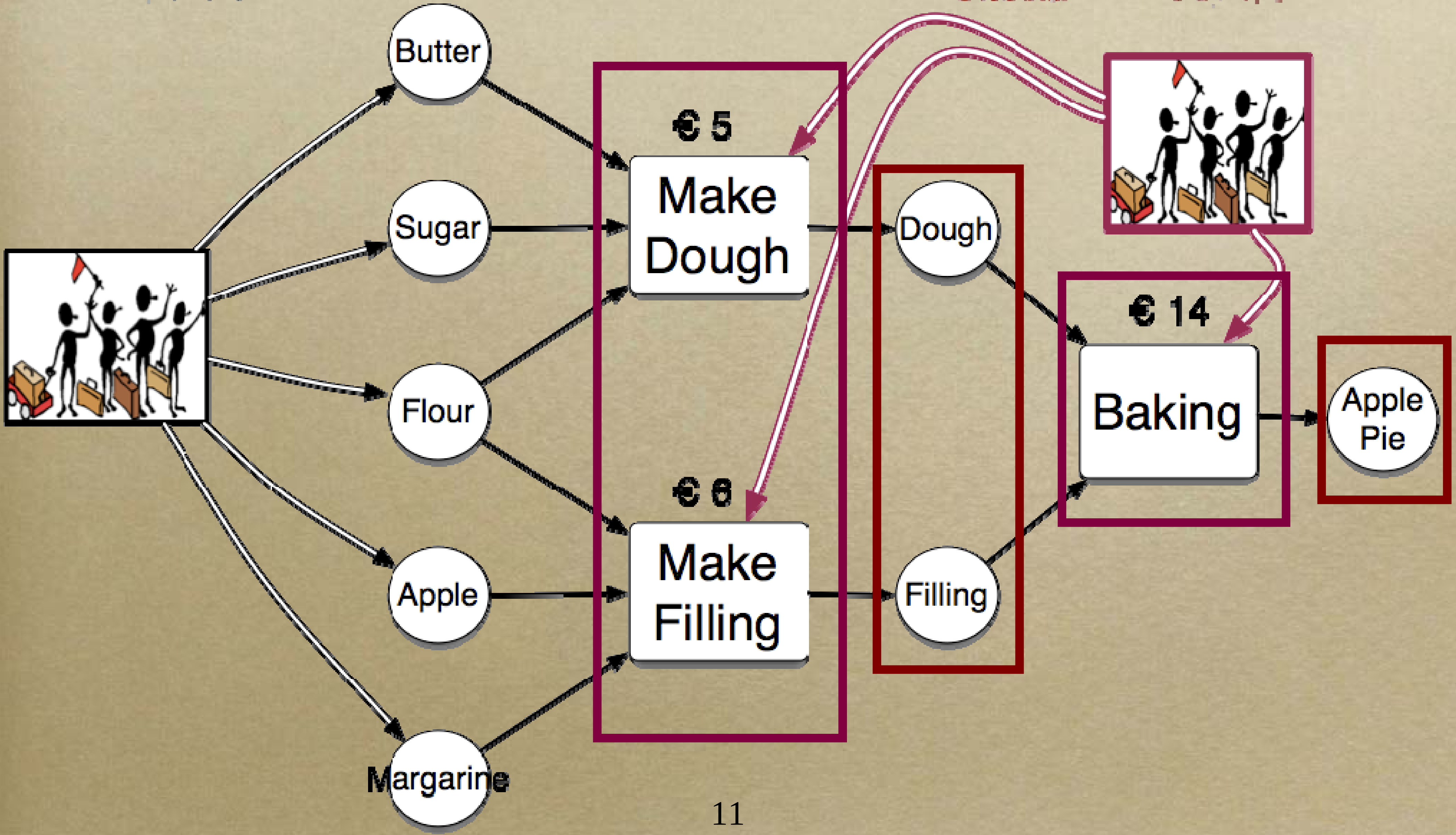
Make-or-Buy


Butter
€ 5
Sugar
Make Dough
Dough
€ 14
Flour
Baking
Apple Pie
€ 6
Apple
Make Filling
Filling
Margarine

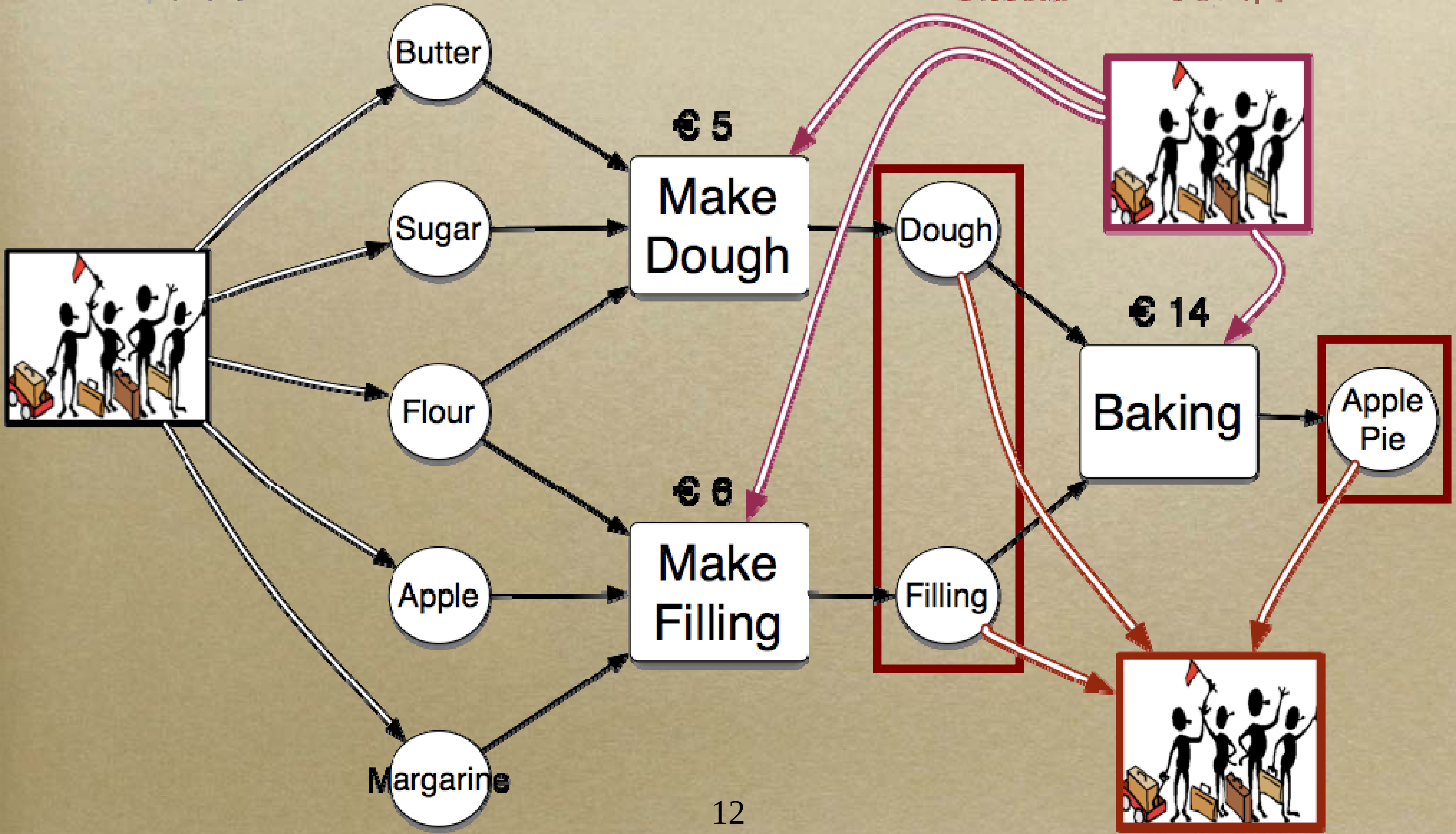
Procurement Stage



Make-or-Buy-or-Collaborate



Make-or-Buy-or-Collaborate



Make-or-Buy-or-Collaborate

- Mixed Multiunit Combinatorial Auctions (MMUCA)
- Automatically selects the best Make-or-Buy-or-Collaborate decisions

Overview

- MMUCA
  - Bidding Language (IJCAI 07)
  - Winner Determination Problem

    (1) Definition (IJCAI 07)

    (2) Solvers
    - Petri-Nets based (AAMAS 07)
    - Direct Integer Programming (IJCAI 07)
    - Connected Component Integer Program (AAMAS 08)
  - Empirical Evaluation (IJA 08)

Outline

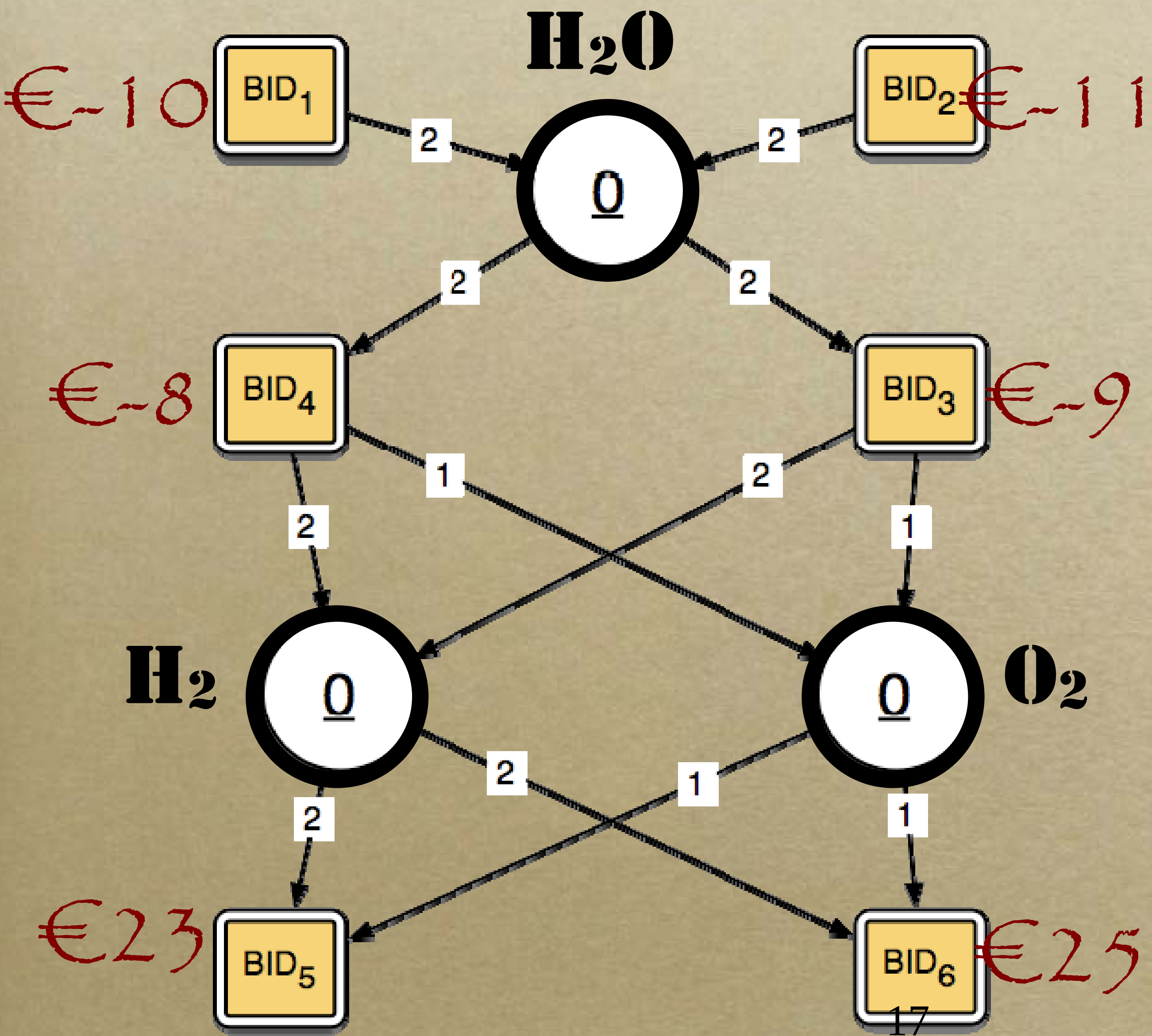
- Motivation
- Background (MMUCA)
- Limitations of WDP solvers for MMUCA
- The Improved Solver
- Empirical evaluation
- Future work

Mixed Multi-unit Combinatorial Auctions

- An extension of Combinatorial Auctions that provides:
  - A formal language to express preferences over operations across the supply chain
  - A formalisation of the optimisation problem that selects:
    - (1) The best business partners
    - (2)A feasible sequence of operations

Automatically selects the best
<u>Make-or-Buy-or-Collaborate</u> decisions

Mixed Multi-unit Combinatorial Auctions



FEASIBILITY

OPTIMALITY

Atomic Bid and Supply Chain Operation

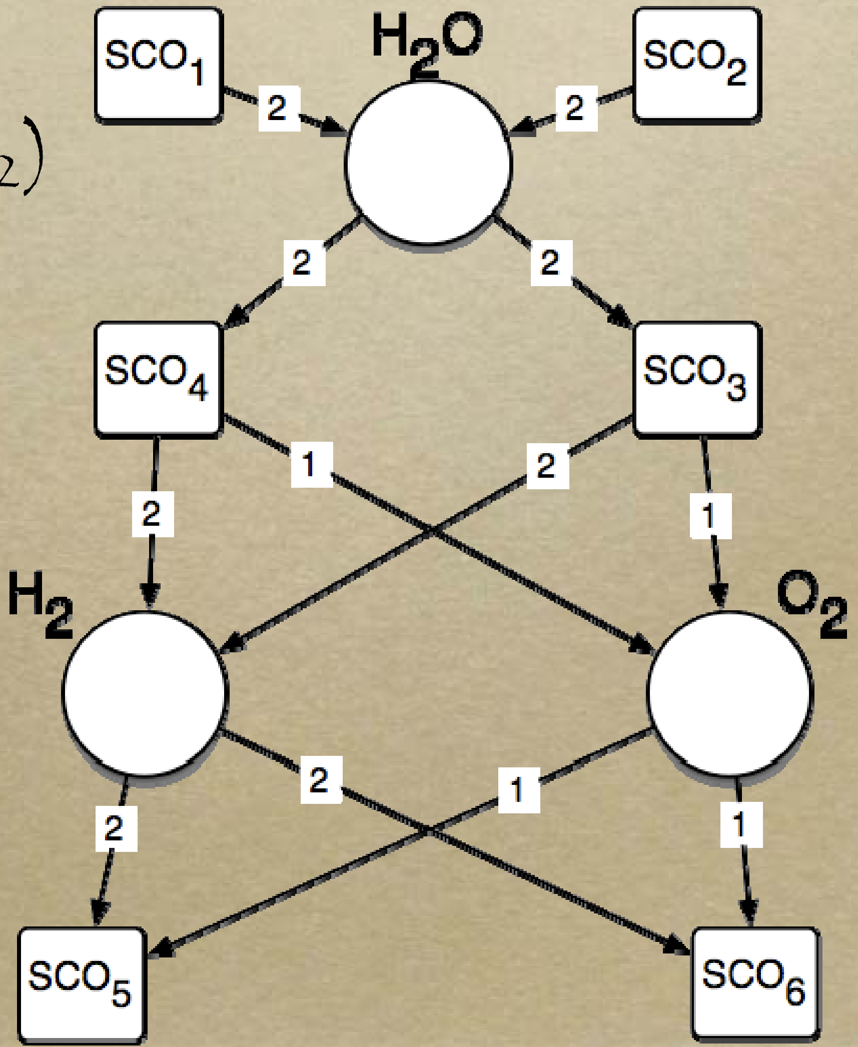
SCO=(Inputs, Outputs)

- SCO_4 = (2'H_2O , 1'O_2 + 2'H_2)
- SCO_5= (1'O_2+2'H_2, nothing)

BID_1=(1'SCO_1+2'SCO_2, -€2)

BID_1 XOR BID_2 XOR BID_3 XOR BID_4

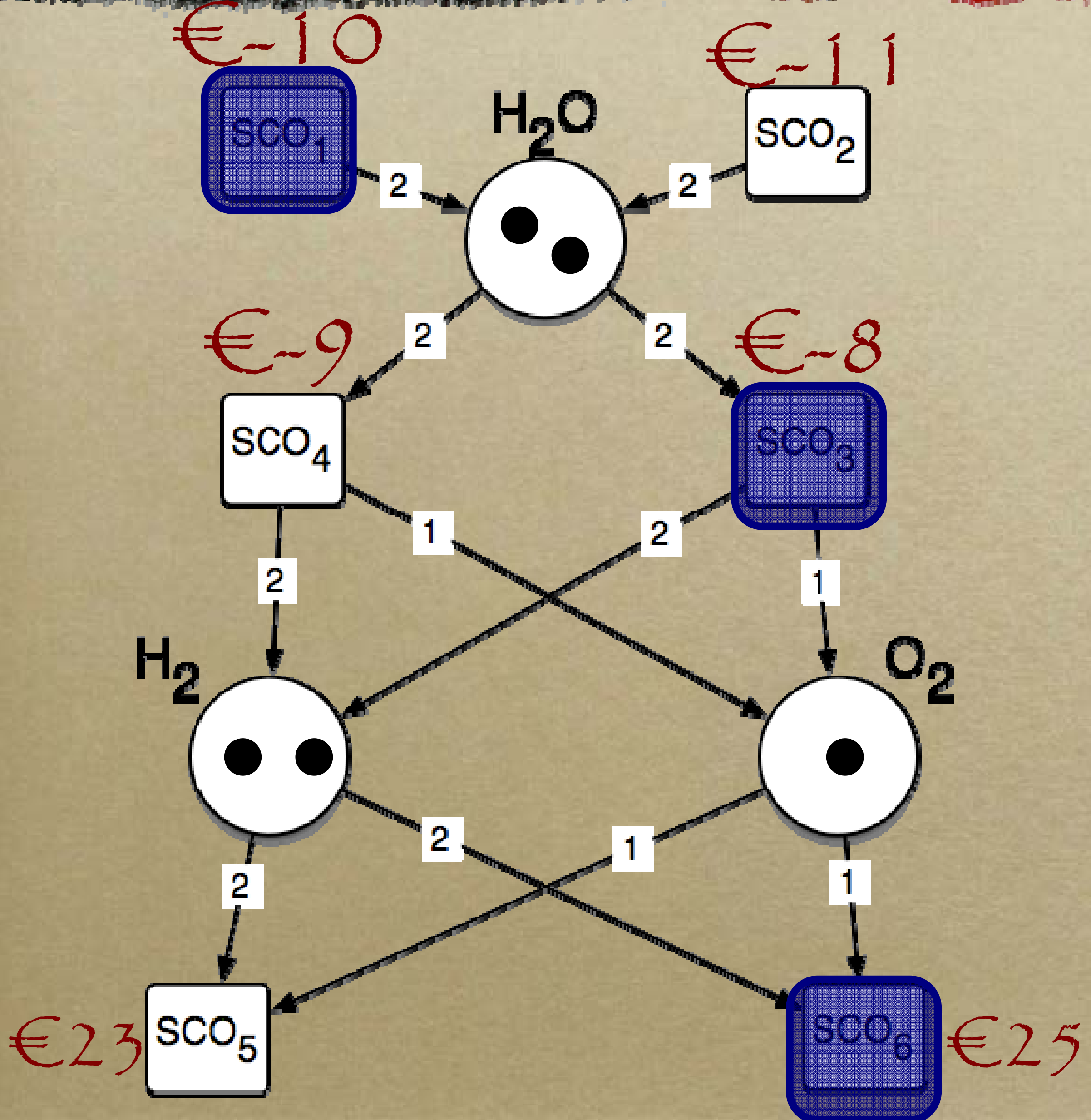
BID_1 OR BID_2 OR BID_3 OR BID_4



Bidding Language

- A bidder can express preferences over bundles of SCOs (Atomic Bid)
- A bidder can submit combinations of Atomic Bids (e.g. XOR, OR)
- Theorem: XOR is expressive enough to represent any valuation

MMUCA WDP



Solution:
<SCO_1>

Solution:
<SCO_1,SCO_3>

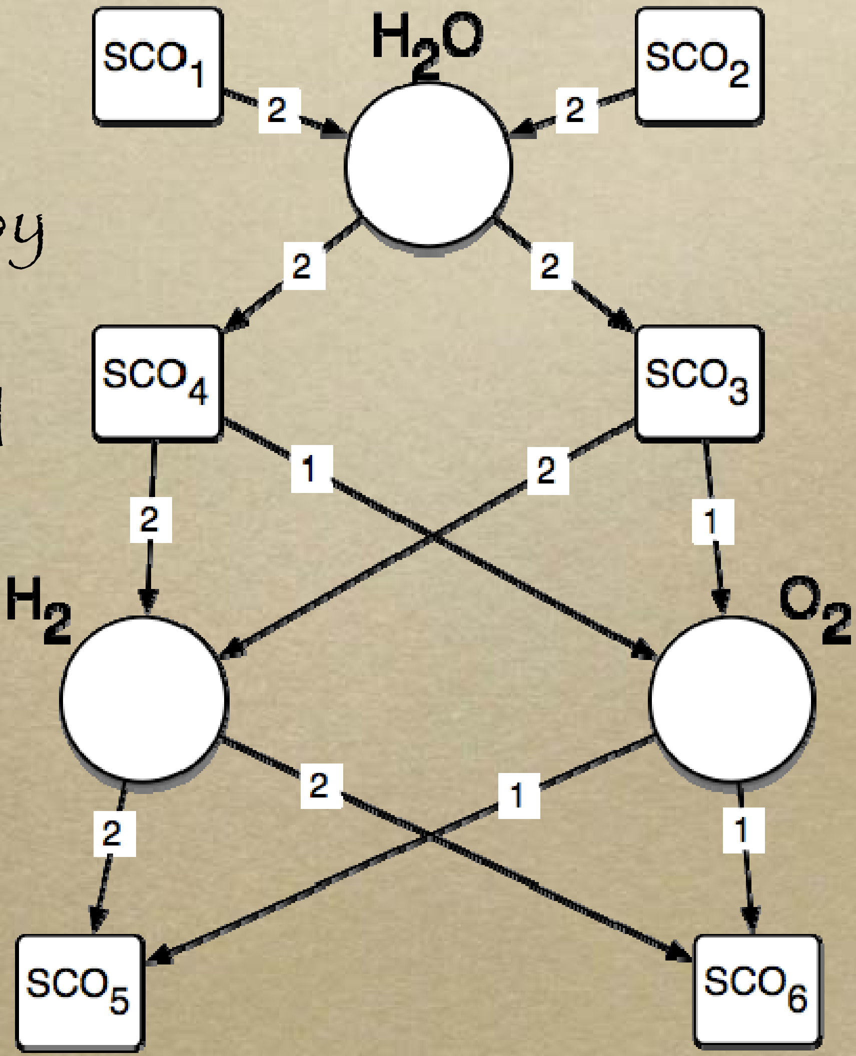
Solution:
<SCO_1,SCO_3,SCO_6>

Revenue:

-10 -8 +25= +7

Winner Determination Problem

- Compute a sequence of SCOs selected among the ones submitted by bidders such that:
  - it fulfils the constraints expressed by the bids
  - it is feasible
  - it maximises the auctioneer's revenue



Outline

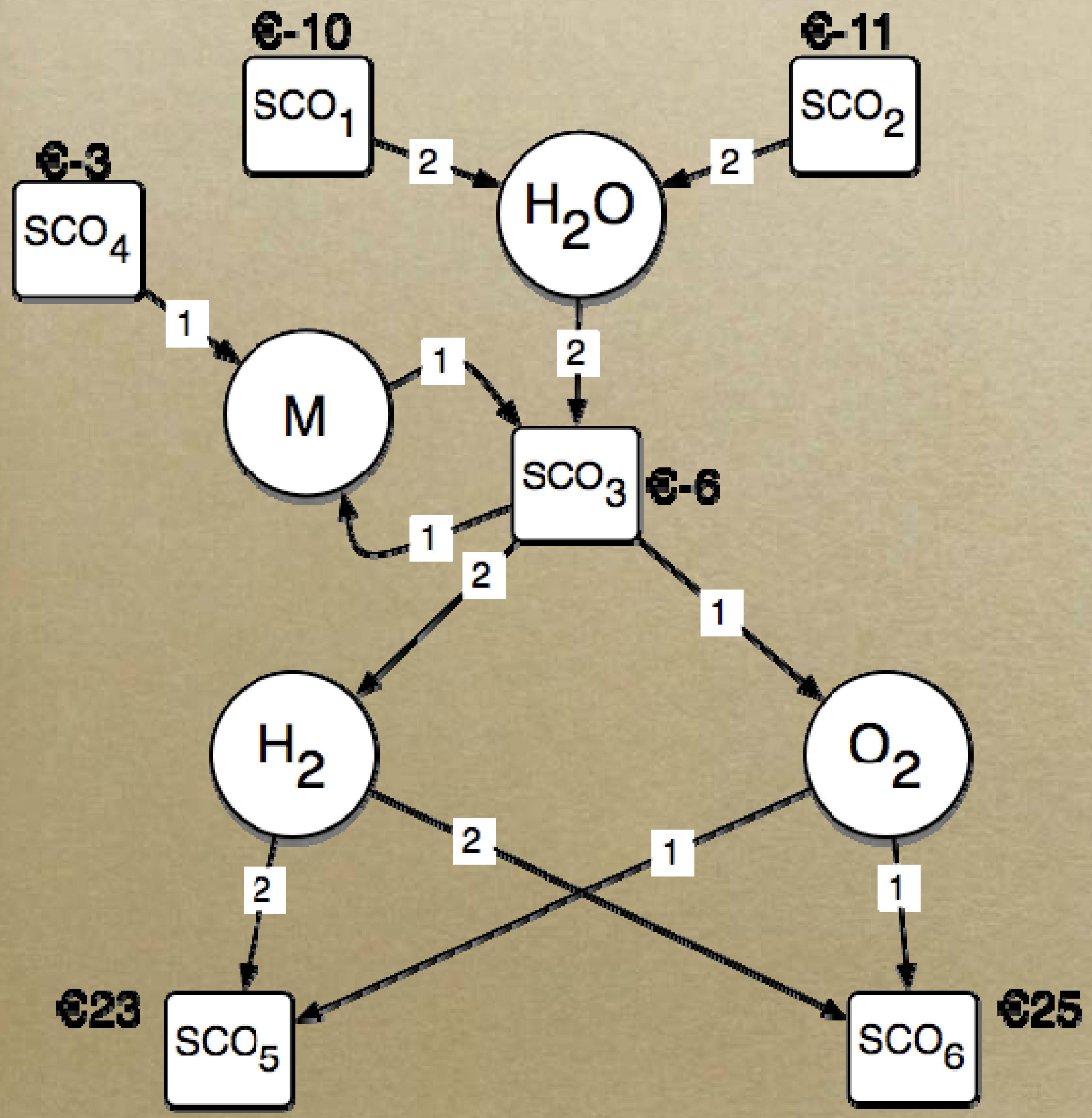
- Motivation
- Background (MMUCA)
- Limitations of WD solvers for MMUCA
- The Improved Solver
- Empirical evaluation
- Future work

Comparing solvers for MMUCA

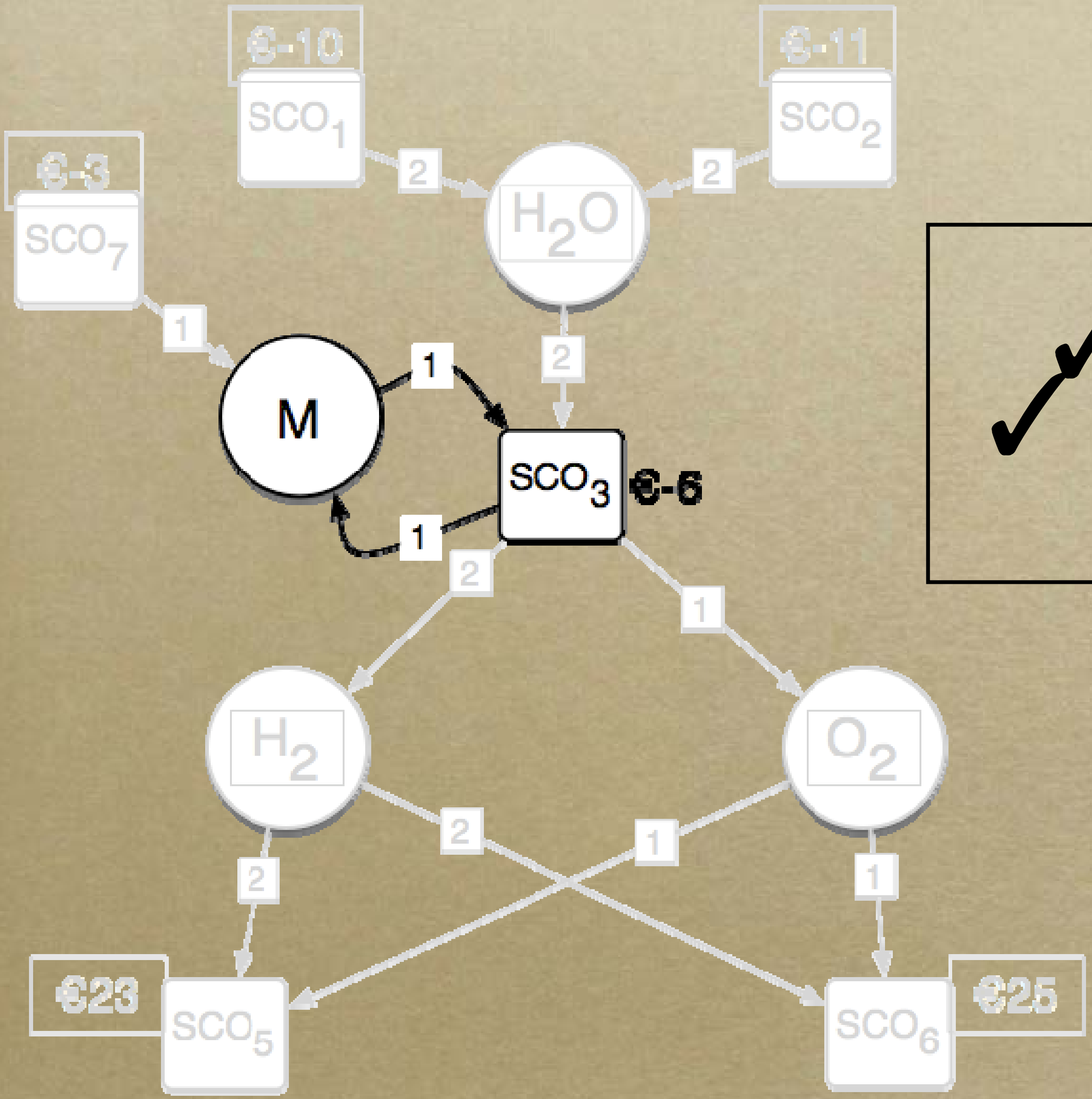
SOLVER	TOPOLOGY	#Decision Variables
Petri-Nets Based Integer Program	ACYCLIC	$O(N)$
Direct Integer Program	ANY	$O(N^2)$
Connected Components IP	ANY	$O(N) \leq ?? << O(N^2)$

N: overall number of Supply Chain Operations

Cyclic topologies



Cyclic topologies

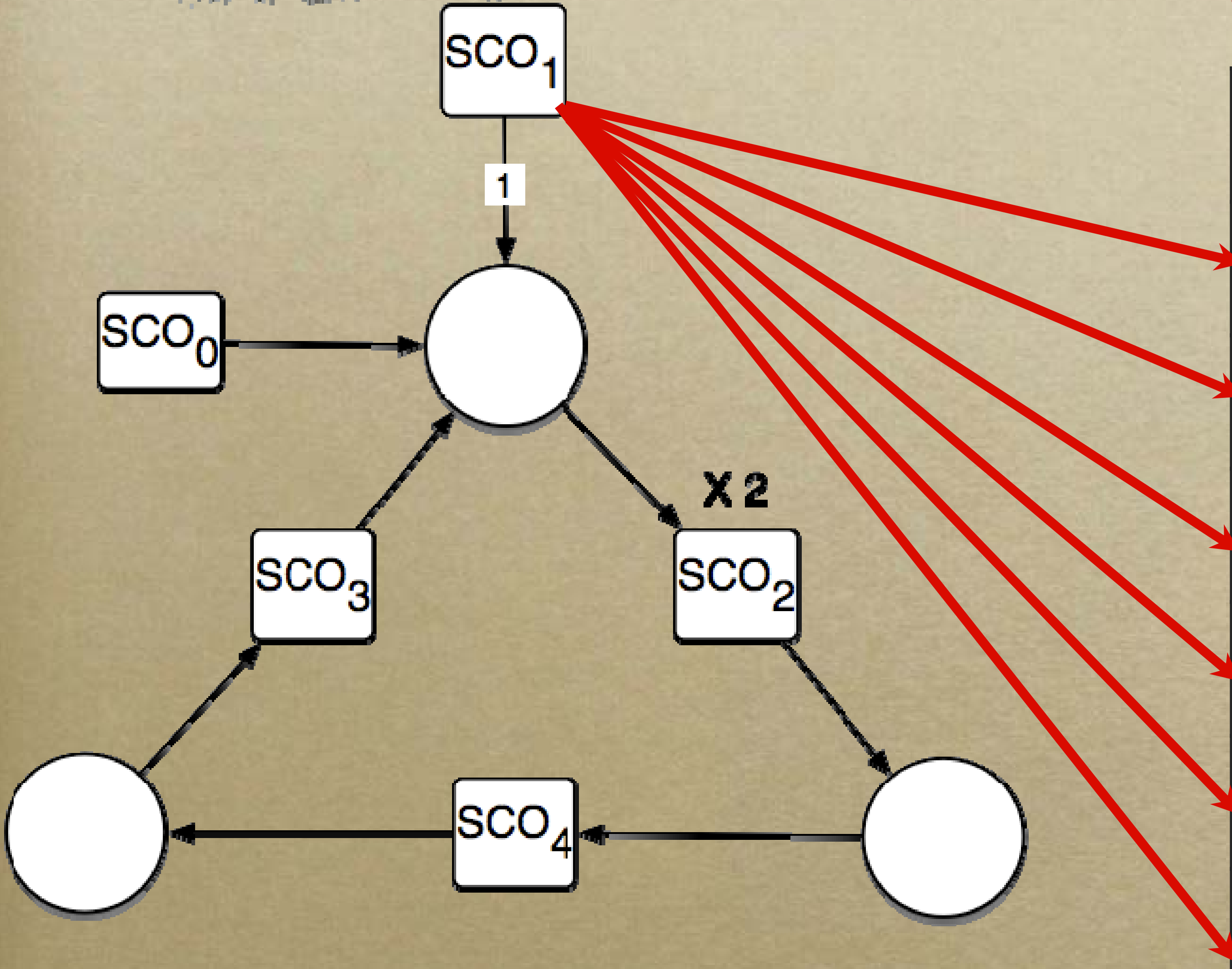


For instance

✓ Resource reuse

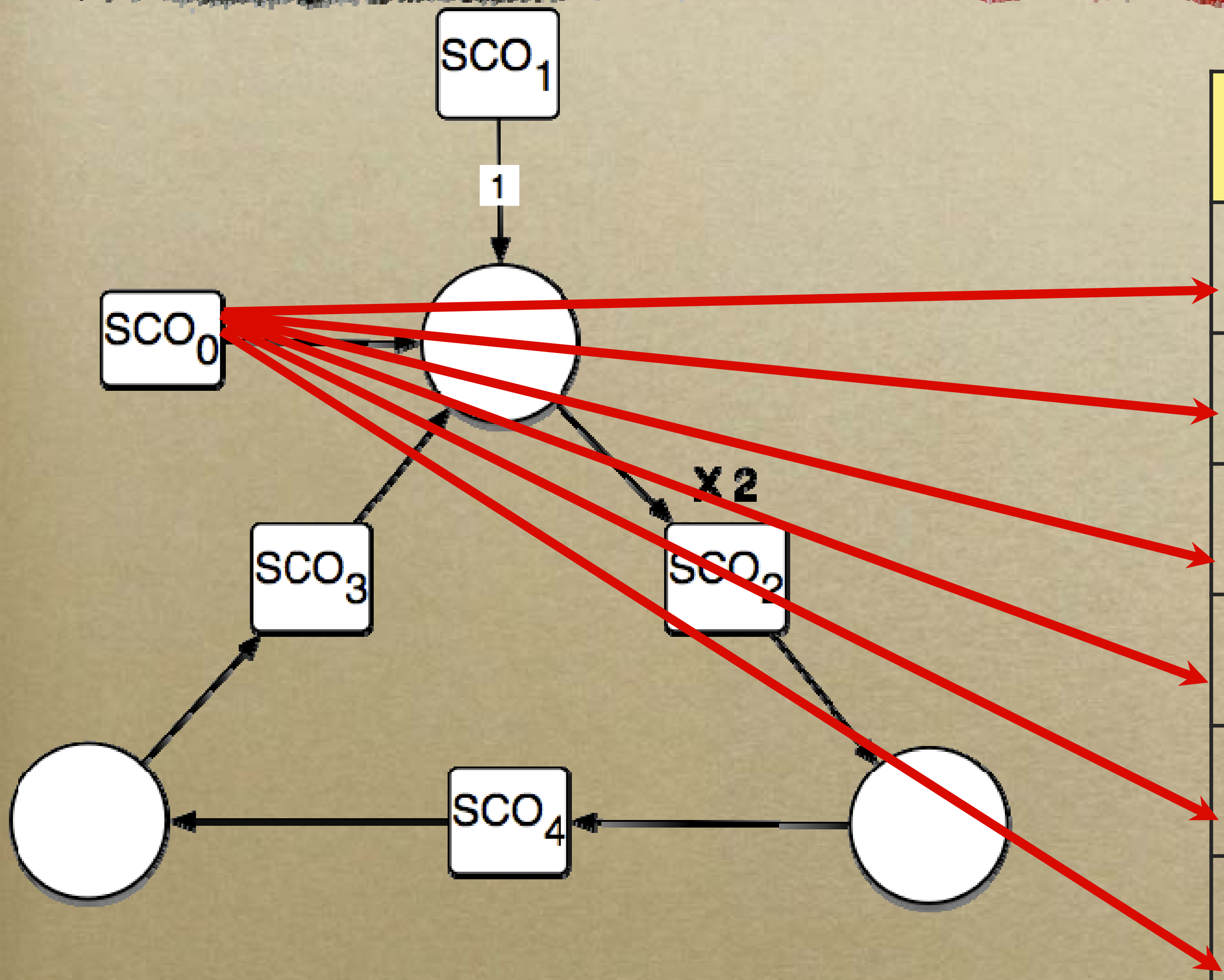
✓ Production Cycles

Direct Integer Program



Positions	Solution
1	SCO$_i$
2	SCO$_i$
3	SCO$_i$
4	SCO$_i$
5	SCO$_i$
6	SCO$_i$

Direct Integer Programming approach



Positions	Solution
1	SCO$_0$
2	SCO$_0$
3	SCO$_0$
4	SCO$_0$
5	SCO$_0$
6	SCO$_0$

DIP explained

Positions	1	2	3	4	5	6
SCOs	SCO$_1$ SCO$_2$ SCO$_3$ SCO$_4$ SCO$_5$ SCO$_6$	SCO$_1$ SCO$_2$ SCO$_3$ SCO$_4$ SCO$_5$ SCO$_6$	SCO$_1$ SCO$_2$ SCO$_3$ SCO$_4$ SCO$_5$ SCO$_6$	SCO$_1$ SCO$_2$ SCO$_3$ SCO$_4$ SCO$_5$ SCO$_6$	SCO$_1$ SCO$_2$ SCO$_3$ SCO$_4$ SCO$_5$ SCO$_6$	SCO$_1$ SCO$_2$ SCO$_3$ SCO$_4$ SCO$_5$ SCO$_6$

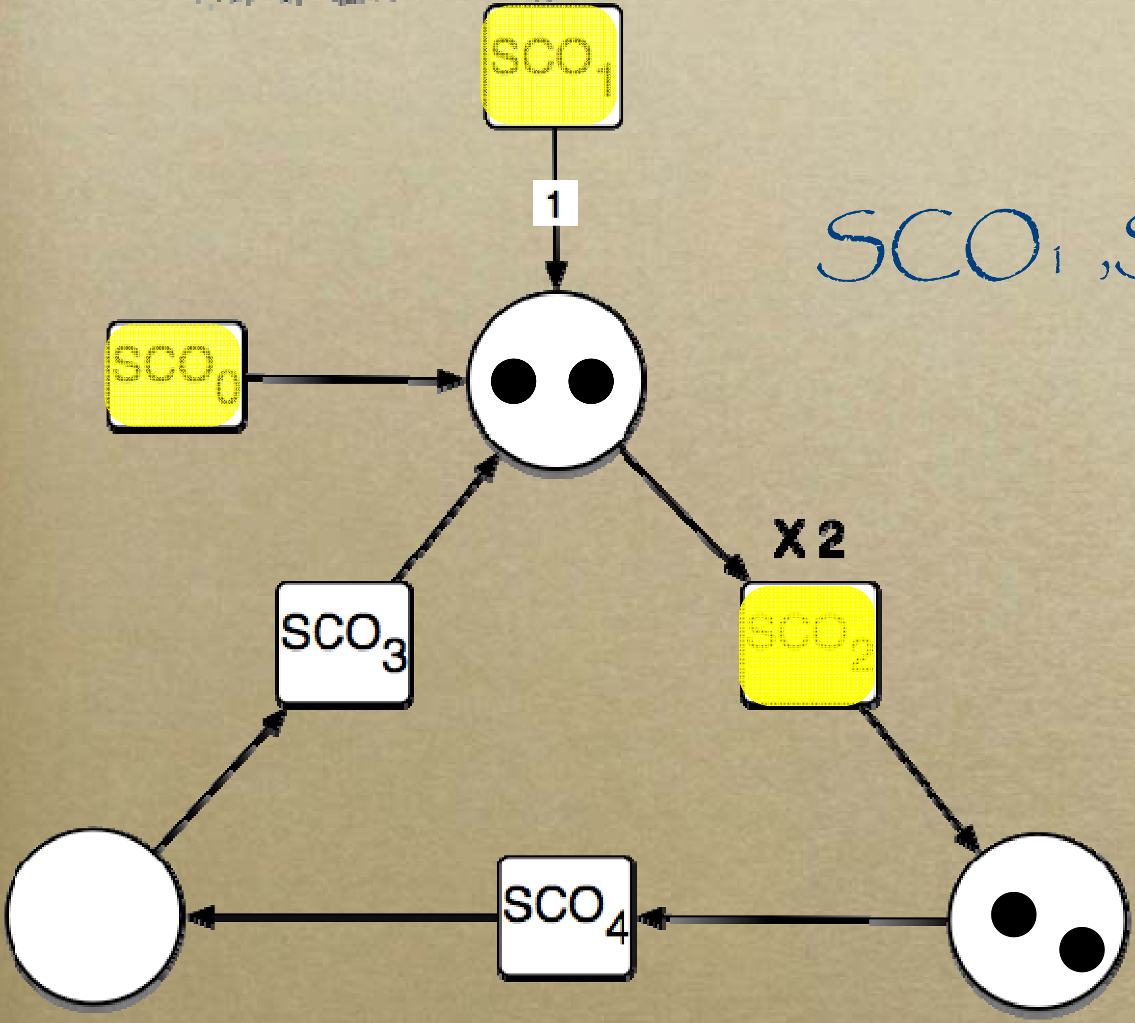
Problem

- The search space associated to DIP is big
- This affects the computational performance of DIP
- Can we reduce the associated search space?

Outline

- Motivation
- Background (MMUCA)
- Limitations of WD solvers for MMUCA
- The Improved Solver
- Empirical evaluation
- Future work

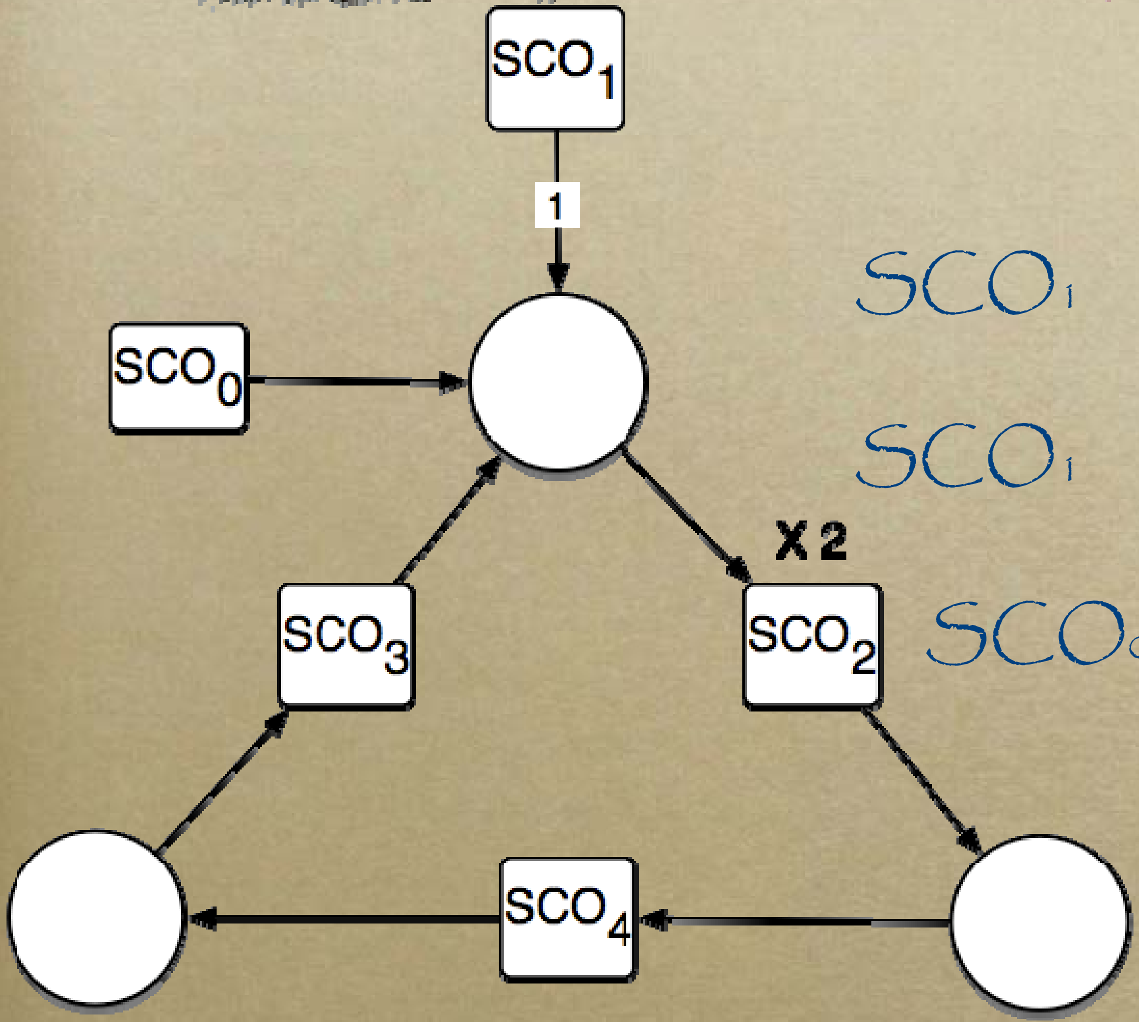
Equivalent Solutions



Solution sequence:

SCO_1 ,SCO_2 ,SCO_0 ,SCO_2

Equivalent Solutions



Solution sequence:

SCO$_1$,SCO$_2$,SCO$_0$,SCO$_2$

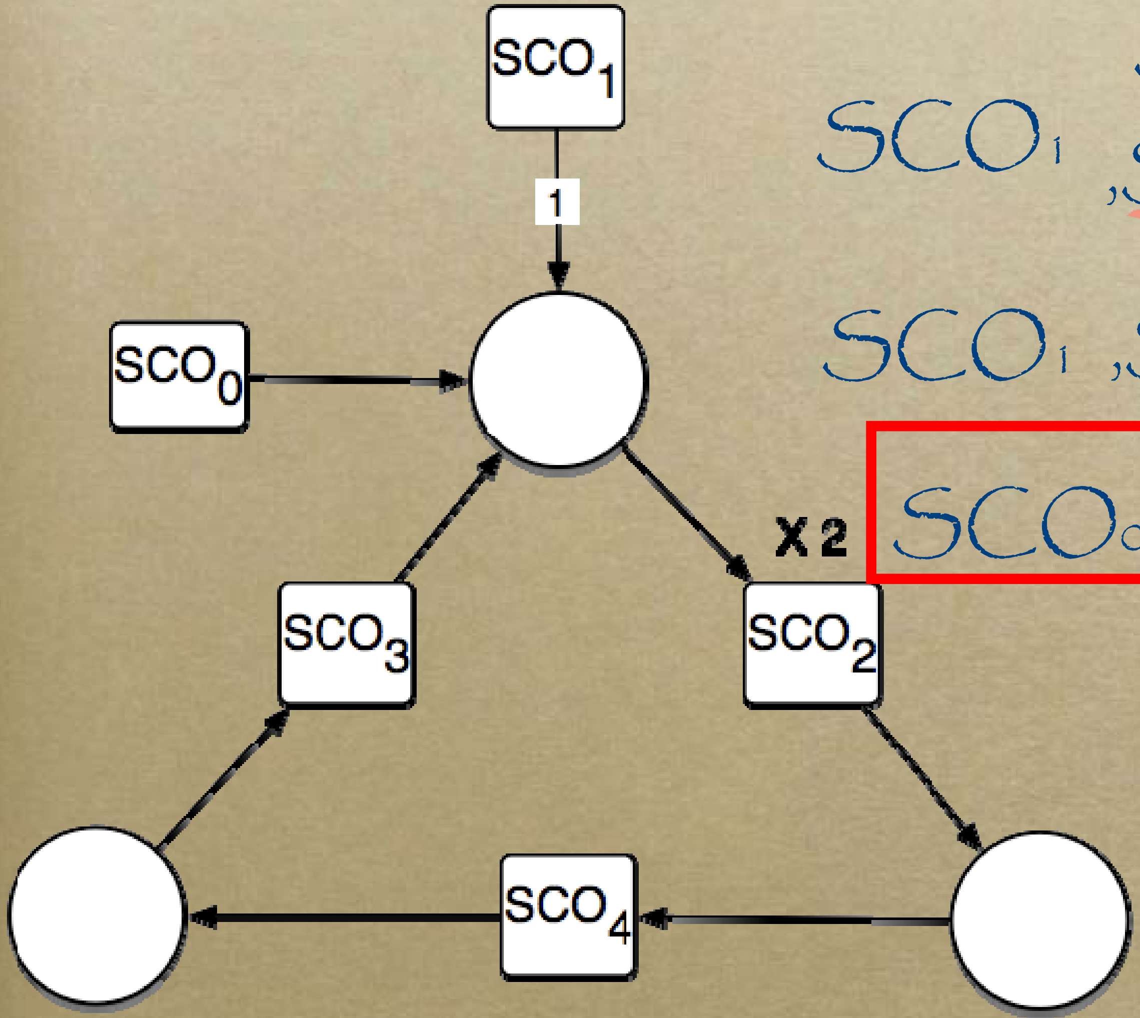
SCO$_1$,SCO$_0$,SCO$_2$,SCO$_2$

SCO$_0$,SCO$_1$,SCO$_2$,SCO$_2$

Reducing the search space

- Can we avoid considering re-orderings of the solution sequence?
- Indeed: Assume that the auctioneer doesn't care about the ordering of a solution sequence as long as enough goods are available for every SCO in the sequence

Equivalent Sequences



Solution sequence:

SCO_1 ~~,SCO_2 ,SCO_0~~ ,SCO_2

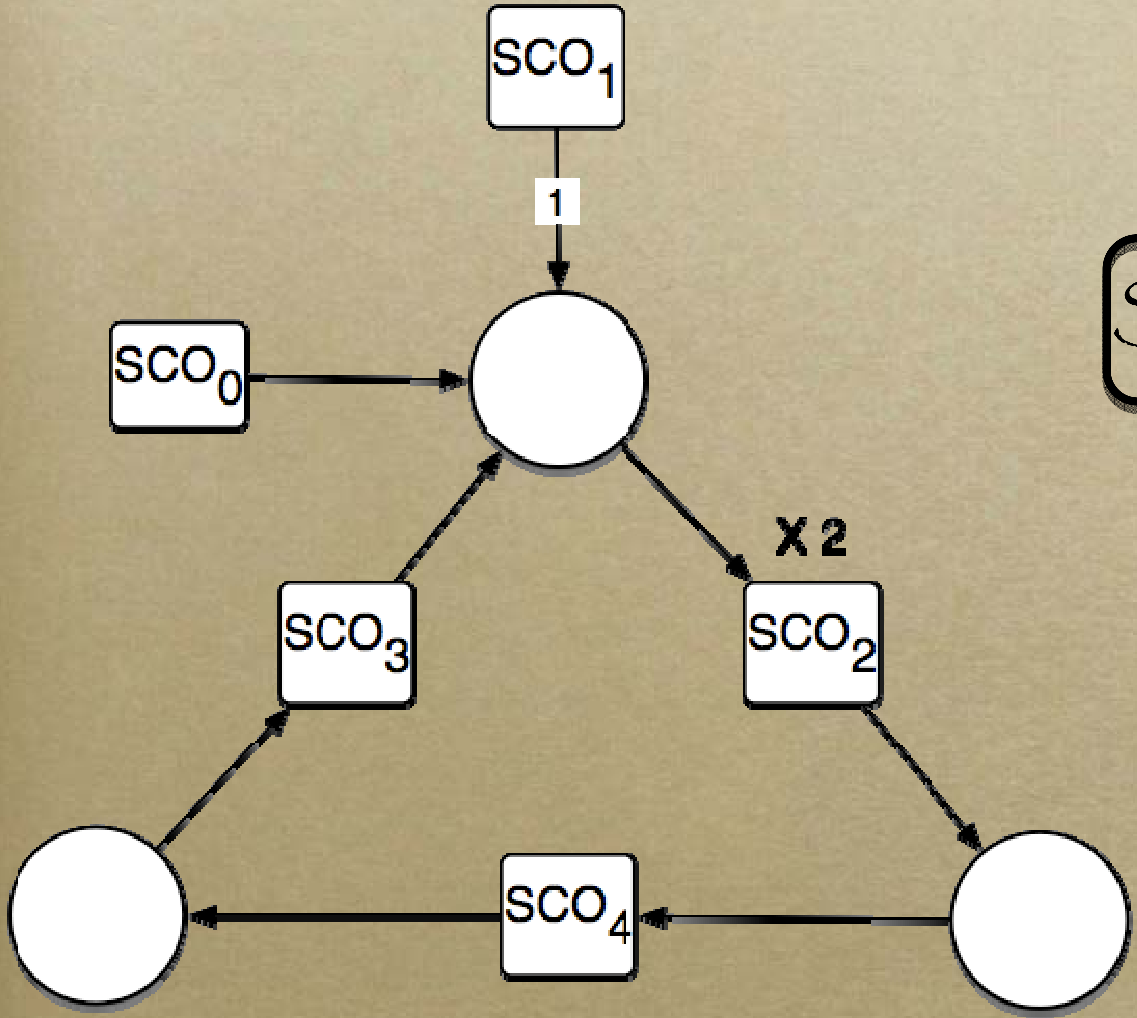
SCO_1 ~~,SCO_0 ,SCO_2~~ ,SCO_2

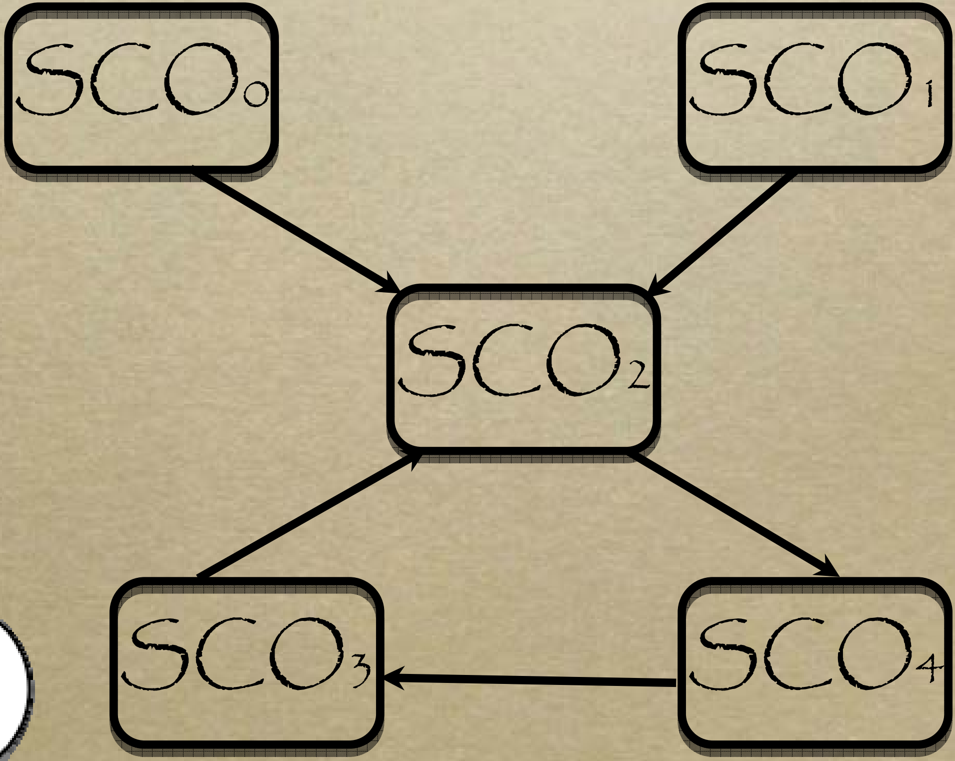
SCO_0 ,SCO_1 ,SCO_2 ,SCO_2

How to remove some sequences

- Each solution to the MMUCA WDP can be reordered into a solution that complies with a given TEMPLATE
- This template is built considering the dependency relationships among SCOs

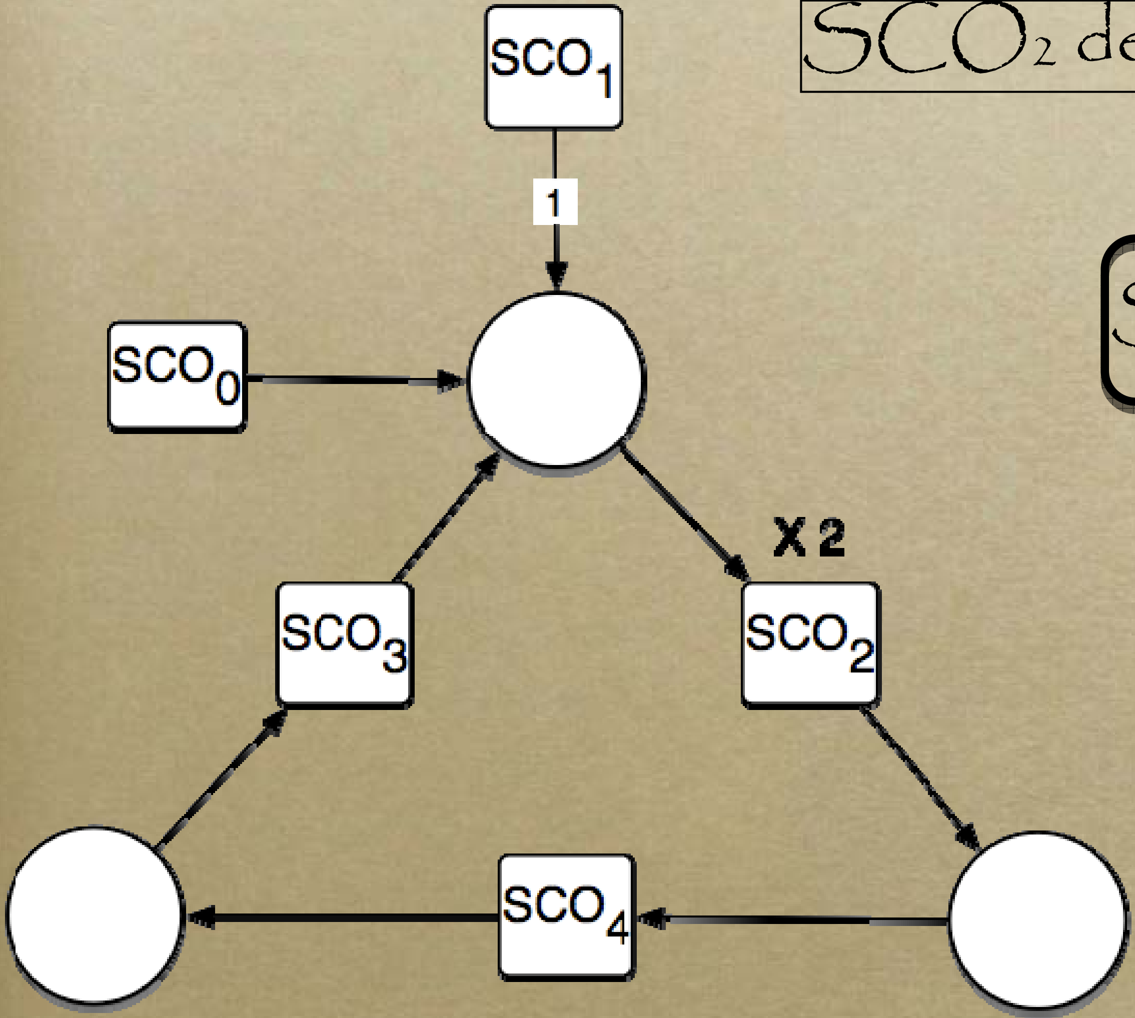
SCO Dependency Graph

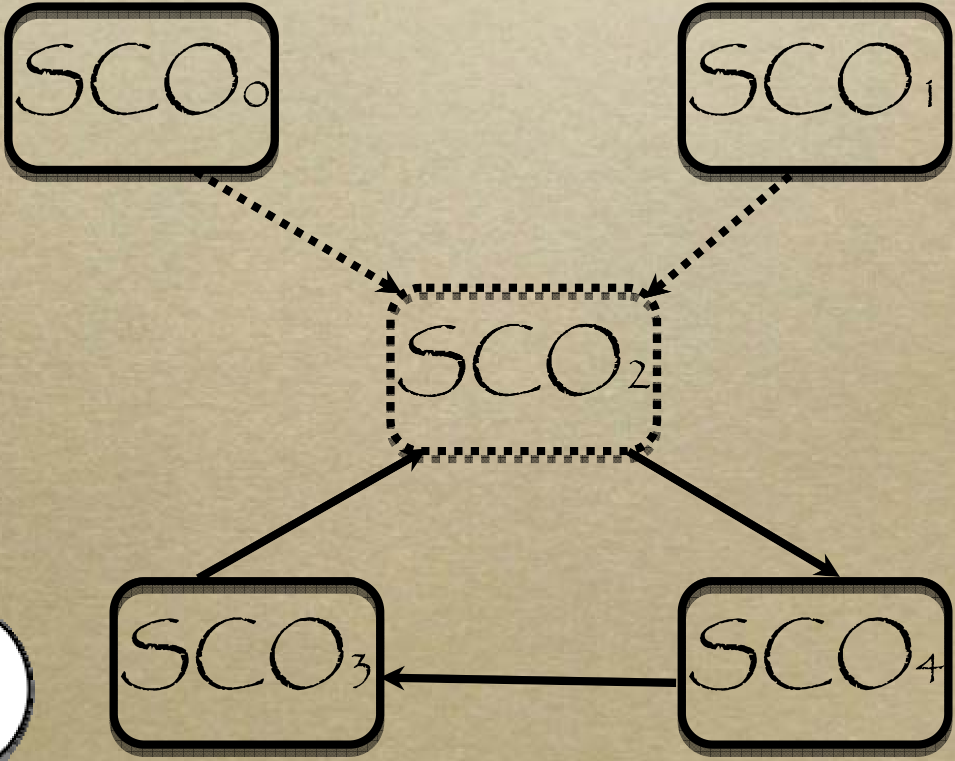




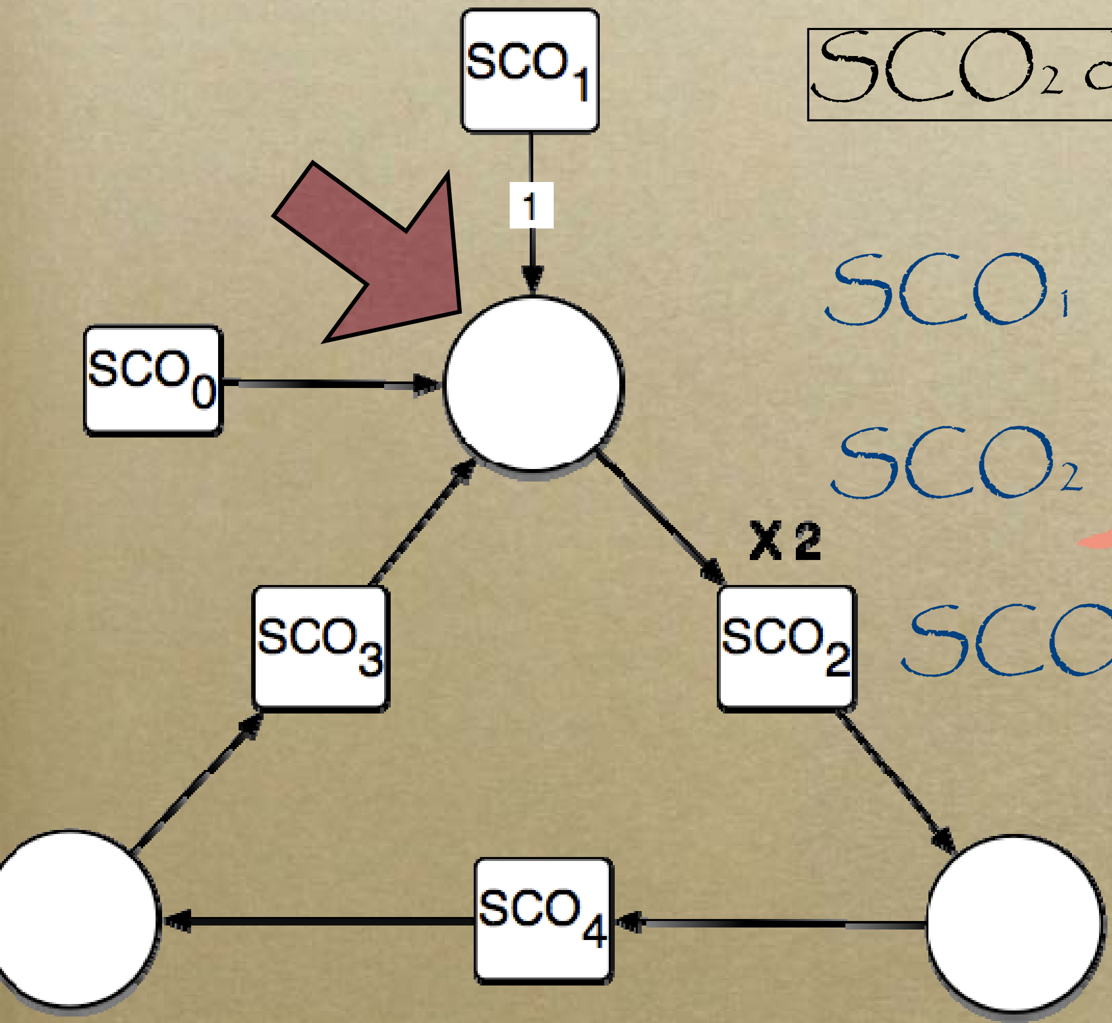
SCO Dependency Graph

SCO$_2$ depends on SCO$_0$,SCO$_1$





SCO Dependency Graph



SCO_2 depends on SCO_0,SCO_1

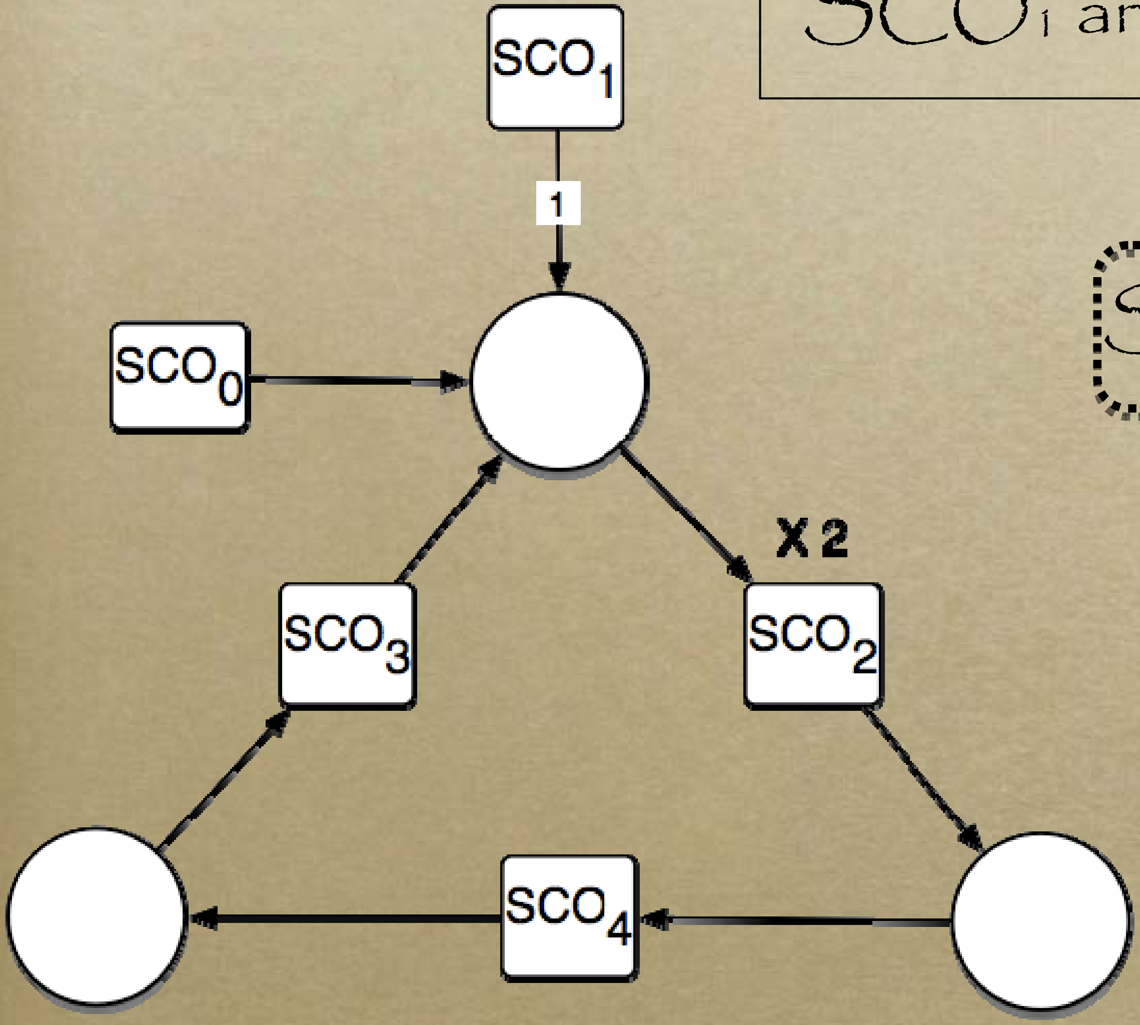
SCO_1 ,SCO_2 ,SCO_0 ,SCO_2

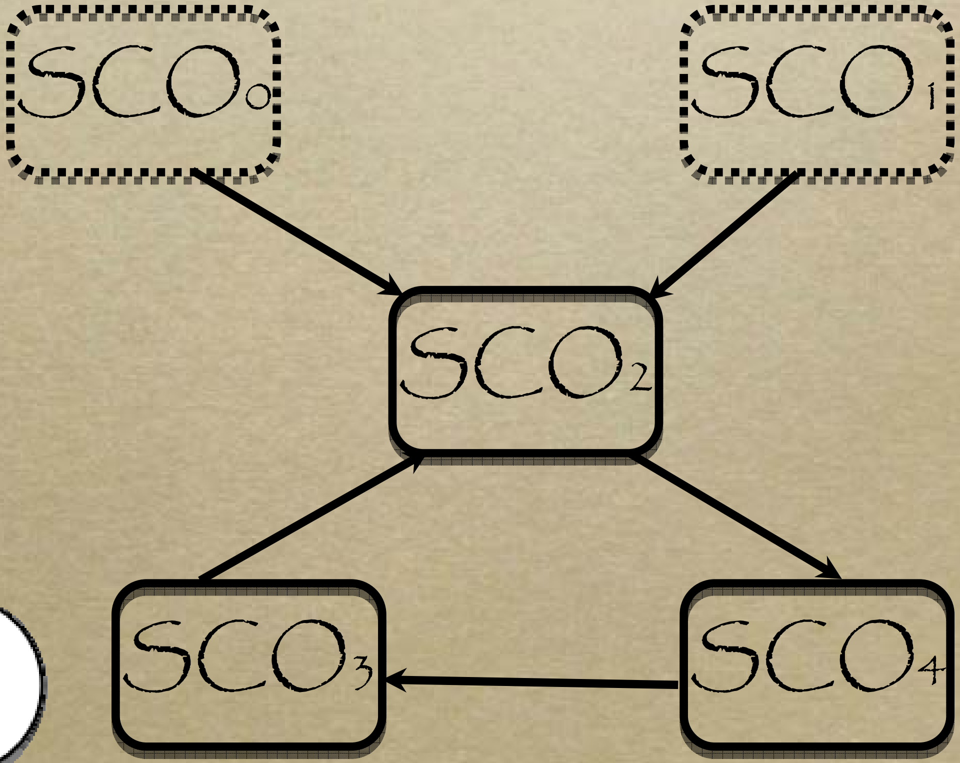
SCO_2 ,~~SCO ,SCO_0~~ ,SCO_2

SCO_0 ,SCO_1 ,SCO_2 ,SCO_2

SCO Dependency Graph

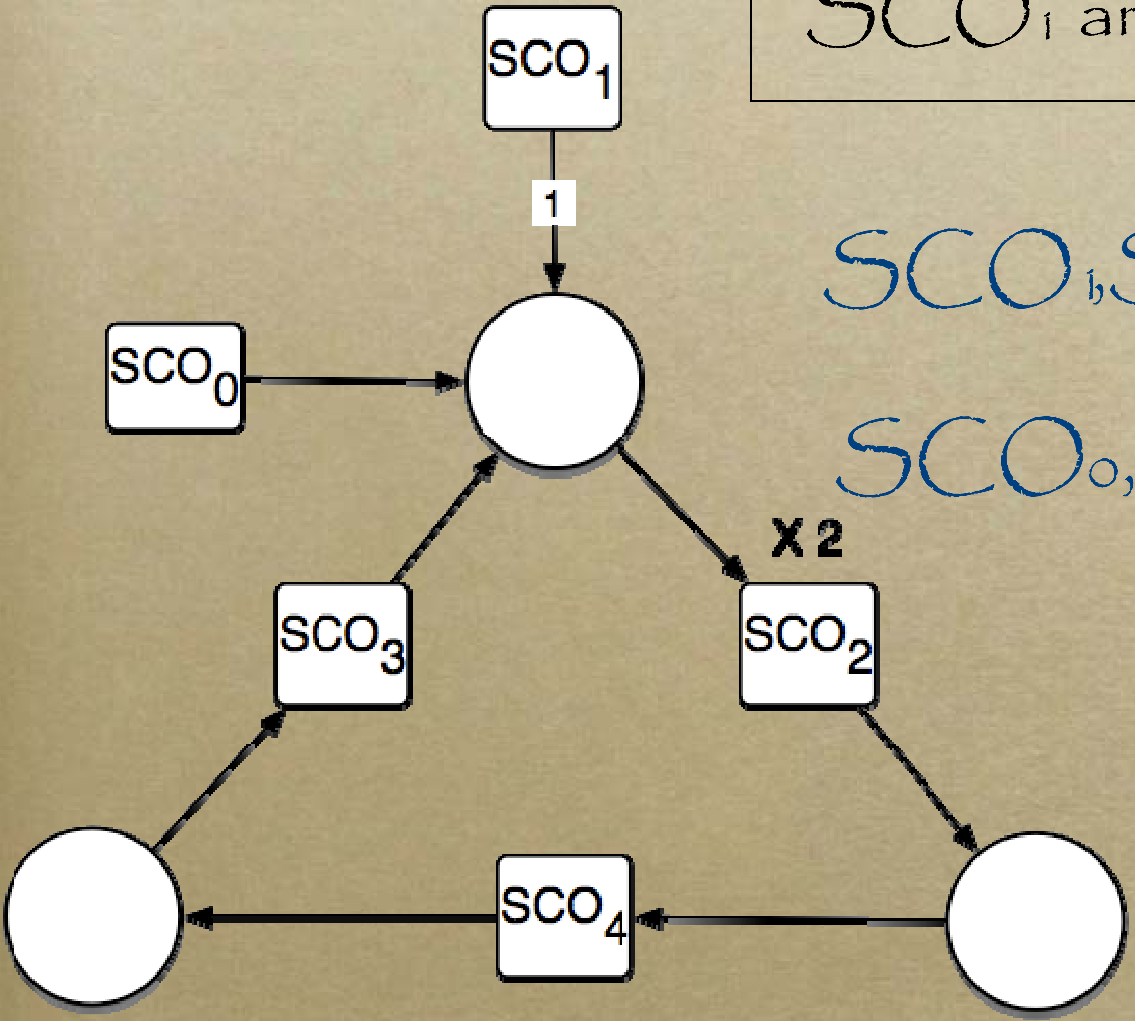
SCO$_1$ and SCO$_0$ are independent





SCO Dependency Graph

SCO_1 and SCO_0 are independent



SCO_1,SCO_2 ,SCO_0 ,SCO_2

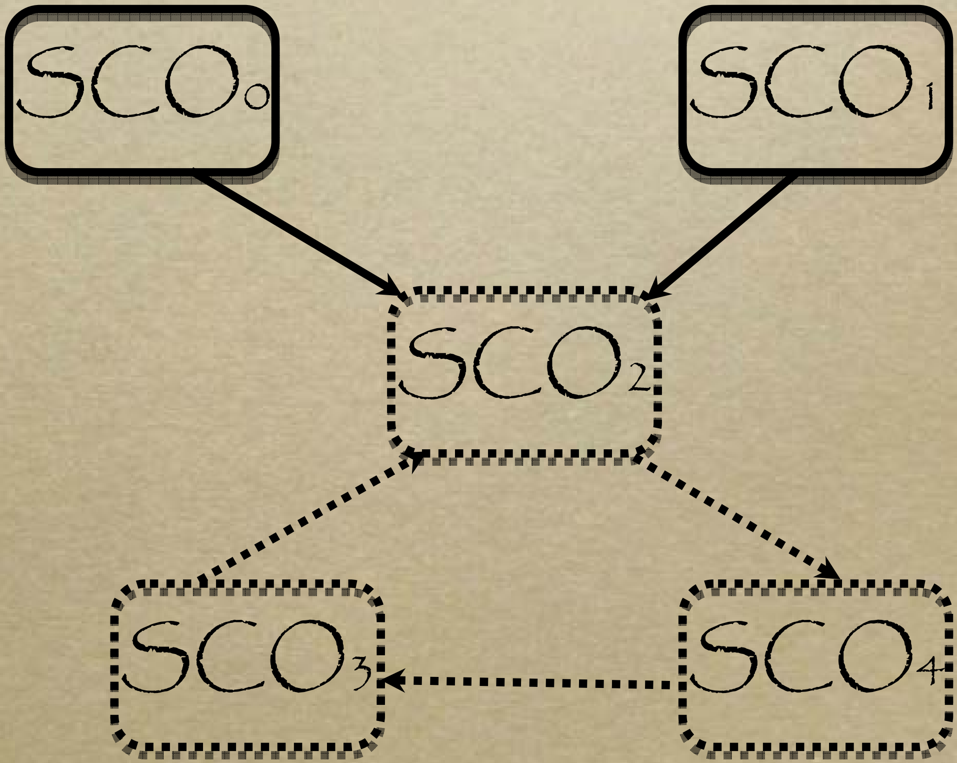
SCO_0,SCO_2 ,SCO_1 ,SCO_2

SCO Dependency Graph

SCO_4 depends on SCO_2

SCO_2 depends on SCO_4

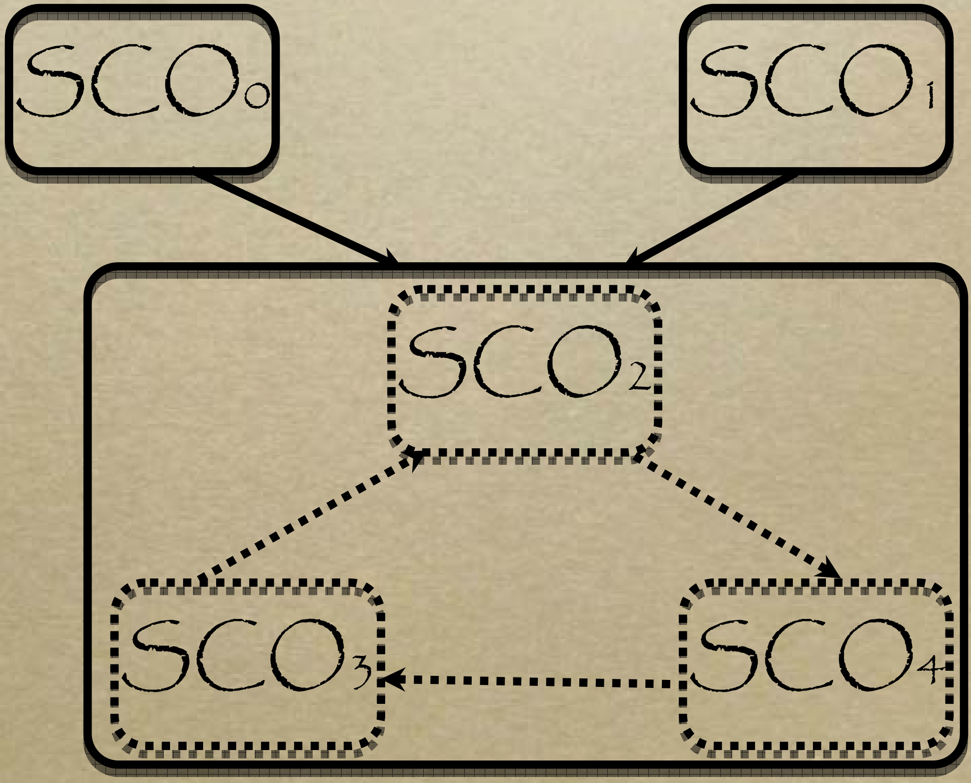
SCO_2,SCO_4 belong to a loop



Strongly Connected Components

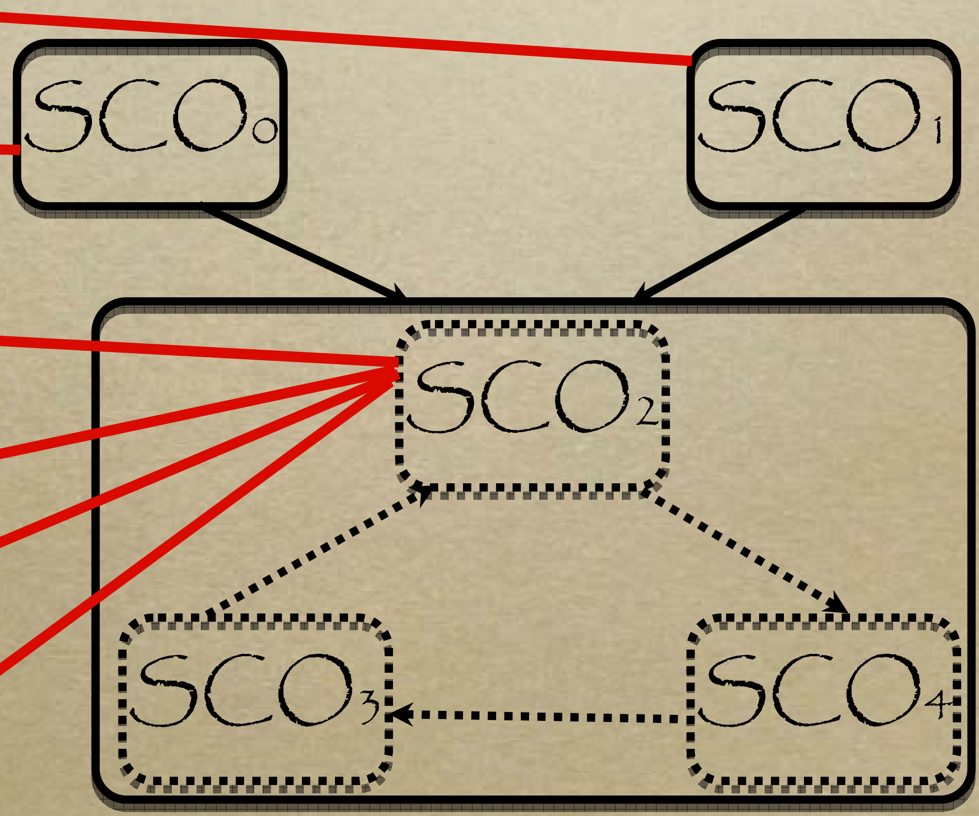
SCO_2, SCO_3, SCO_4 cannot be ordered among them

We group them: <u>SCCs</u>



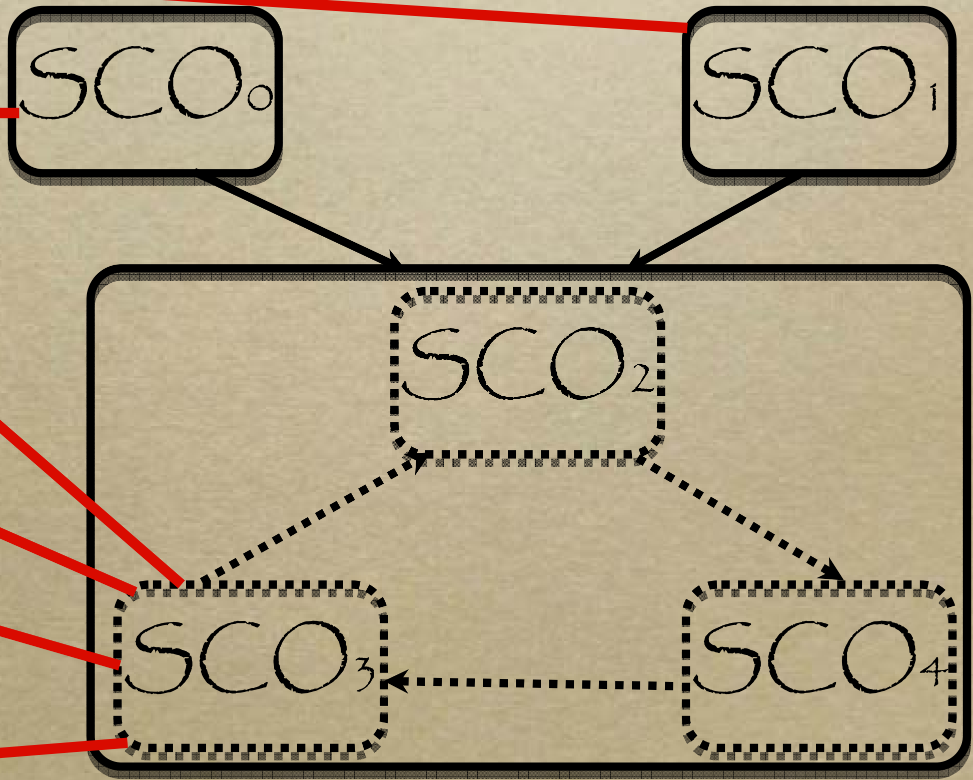
Strongly Connected Components

Pos	Solution
1	SCO_1
2	SCO_0
3	SCO_2,SCO_3,SCO_4
4	SCO_2,SCO_3,SCO_4
5	SCO_2,SCO_3,SCO_4
6	SCO_2,SCO_3,SCO_4



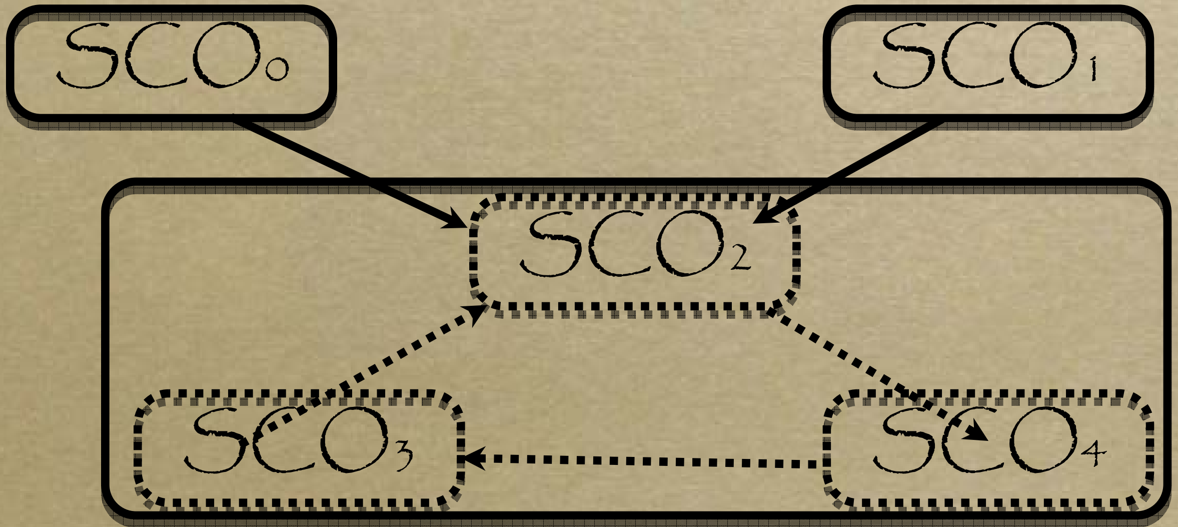
Strongly Connected Components

Pos	Solution
1	SCO_1
2	SCO_0
3	SCO_2,SCO_3,SCO_4
4	SCO_2,SCO_3,SCO_4
5	SCO_2,SCO_3,SCO_4
6	SCO_2,SCO_3,SCO_4



The Solution Template

Positions	1	2	3	4	5	6
Template	SCO_0	SCO_1	SCO_2 SCO_3 SCO_4	SCO_2 SCO_3 SCO_4	SCO_2 SCO_3 SCO_4	SCO_2 SCO_3 SCO_4
SEQA	SCO_0	SCO_1	SCO_3	SCO_2		
SEQB	$\underline{\text{SCO}_2}$	SCO_1	SCO_3	$\underline{\text{SCO}_0}$		



Proof of correctness

- THEOREM: "each solution to the MMUCA WDP can be reordered into an equivalent solution that fulfils the solution template"
- If we reduce the search space to the sequences fulfilling the solution template we do not to lose any solutions

Comparing DIP and CCIP

- The hypothesis behind DIP is that a SCO can hold any position within the solution sequence

$$\boxed{5 \times 6 = 30}$$

Positions	1	2	3	4	5	6
Template	SCO_0 SCO_1 SCO_2 SCO_3 SCO_4	SCO_0 SCO_1 SCO_2 SCO_3 SCO_4	SCO_0 SCO_1 SCO_2 SCO_3 SCO_4	SCO_0 SCO_1 SCO_2 SCO_3 SCO_4	SCO_0 SCO_1 SCO_2 SCO_3 SCO_4	SCO_0 SCO_1 SCO_2 SCO_3 SCO_4

Comparing DIP and CCIP

- The hypothesis behind CCIP is that a SCO can hold only the positions allowed by the template

14

Positions	1	2	3	4	5	6
Template	SCO_0	SCO_1	SCO_2 SCO_3 SCO_4	SCO_2 SCO_3 SCO_4	SCO_2 SCO_3 SCO_4	SCO_2 SCO_3 SCO_4

Comparing solvers

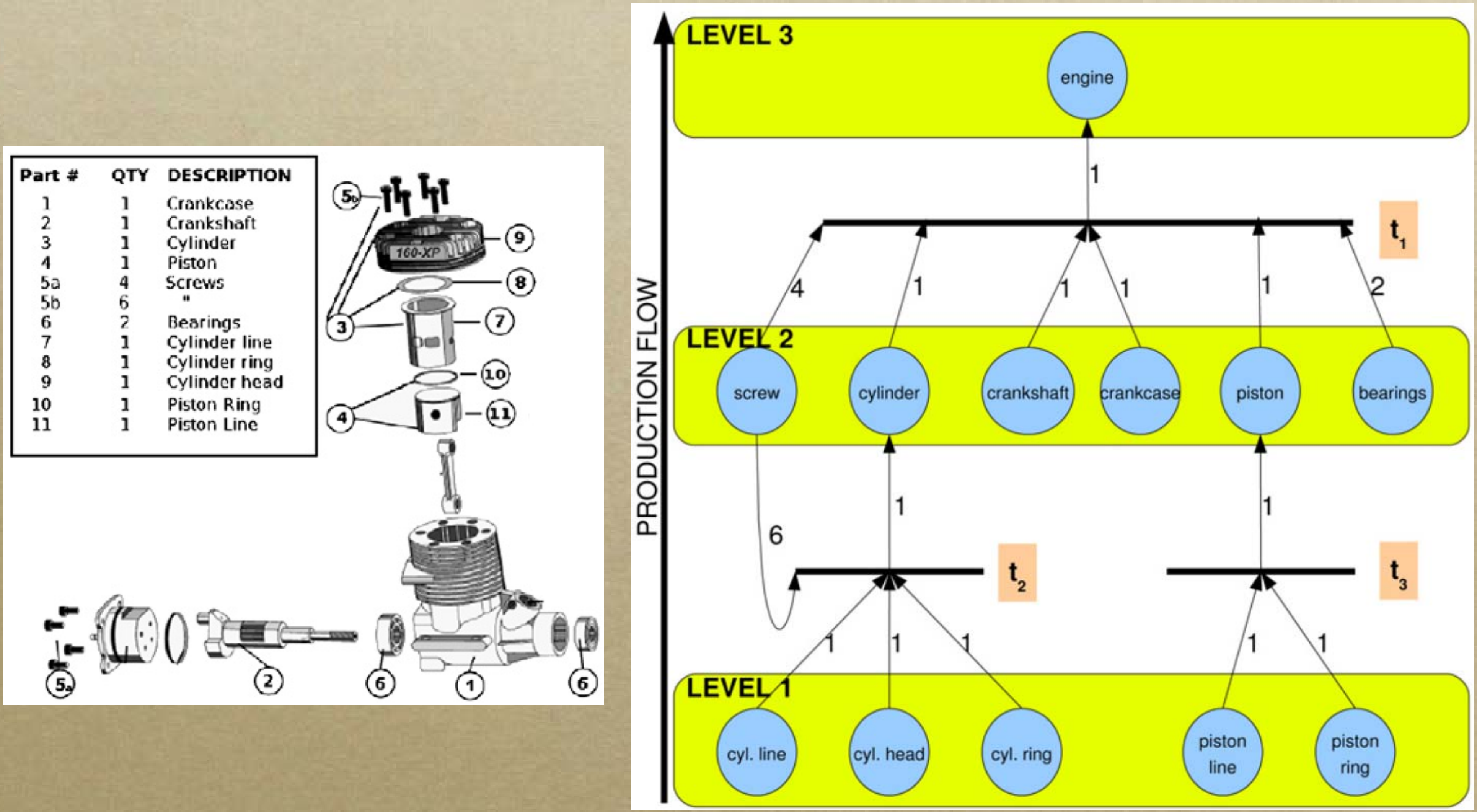
SOLVER	TOPOLOGY	#Decision Variables
Petri-Nets Based	ACYCLIC	O(N)
Direct Integer Program	ANY	$O(N^2)$
Connected Component Integer Program (CCIP)	ANY	$O(k^2 SCC)$

N: overall number of Supply Chain Operations

Outline

- Motivation
- Background (MMUCA)
- Limitations of WD solvers for MMUCA
- The Improved Solver
- Empirical evaluation
- Future work

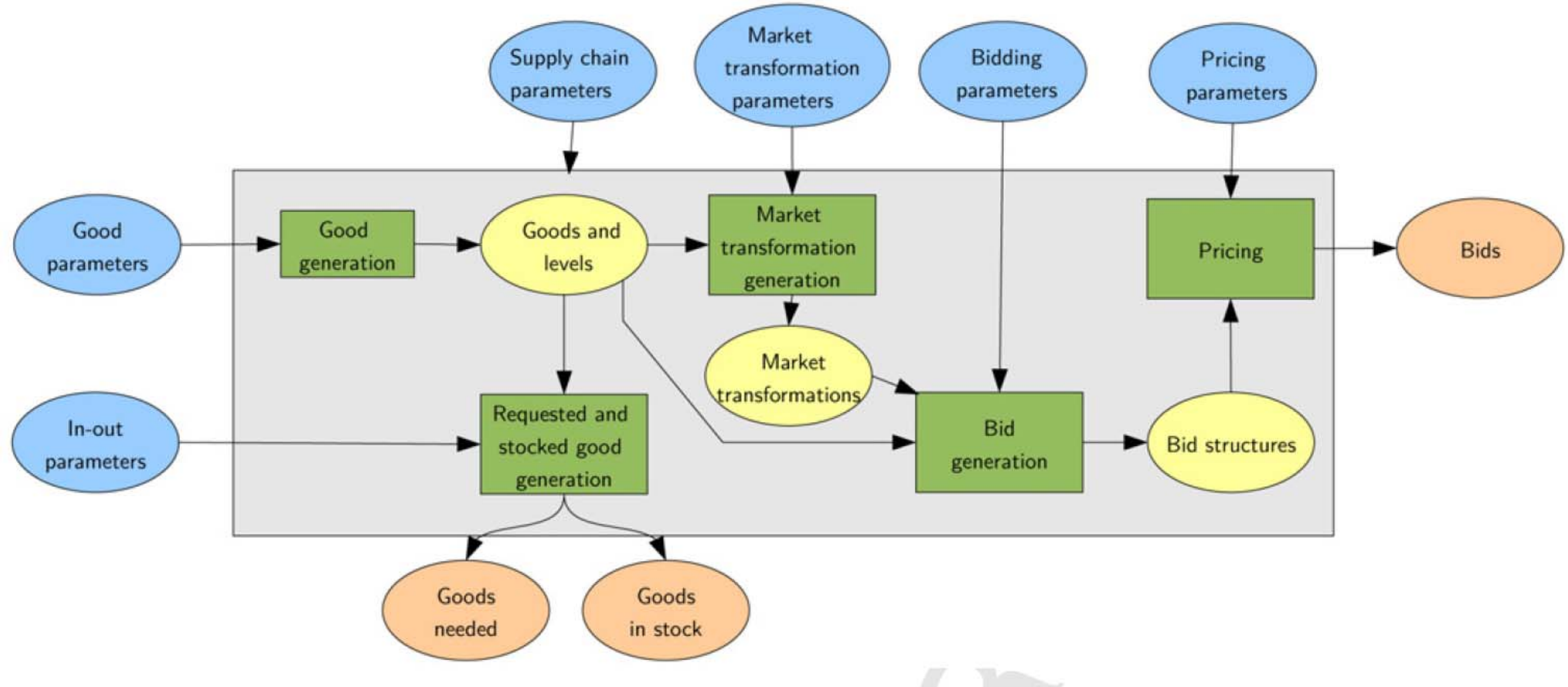
Empirical Evaluation

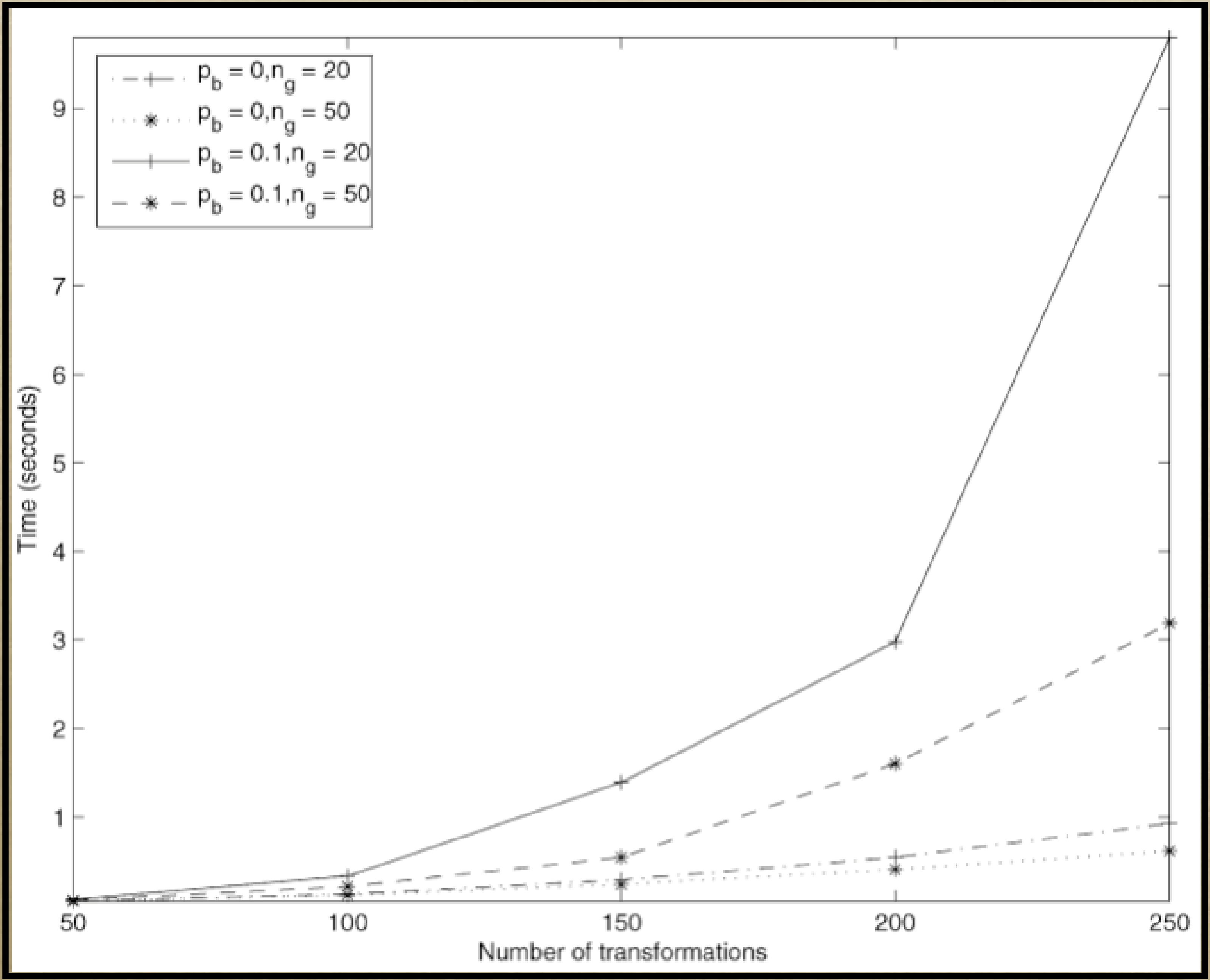


Part #	QTY	DESCRIPTION
1	1	Crankcase
2	1	Crankshaft
3	1	Cylinder
4	1	Piston
5a	4	Screws
5b	6	"
6	2	Bearings
7	1	Cylinder line
8	1	Cylinder ring
9	1	Cylinder head
10	1	Piston Ring
11	1	Piston Line

(a) Components of a car engine. (b) Supply chain for a car's engine.

MMUCA WDP Generator




$p_b = 0, n_g = 20$
$p_b = 0, n_g = 50$
$p_b = 0.1, n_g = 20$
$p_b = 0.1, n_g = 50$
Time (seconds)
1
2
3
4
5
6
7
8
9
50
100
150
200
250
Number of transformations

Conclusions

- The scalability of an IP implementation of MMUCA is affected by the size of the largest connected components
- When there is a "natural" flow in the supply chain, CCIP scales reasonably well wrt number of transformations and goods

Outline

- Motivation
- Background (MMUCA)
- Limitations of WD solvers for MMUCA
- The Improved Solver
- Empirical evaluation
- Future work

Future Work

- Incorporate time
  - time to perform operations
  - time to finish before a deadline
- Incorporate uncertainty
  - bidders may fail
  - maximise the expected value
- Study connections to Planning