Lecture 6: Cooperation in MAS (II) – Coalition formation

Multi-Agent Systems

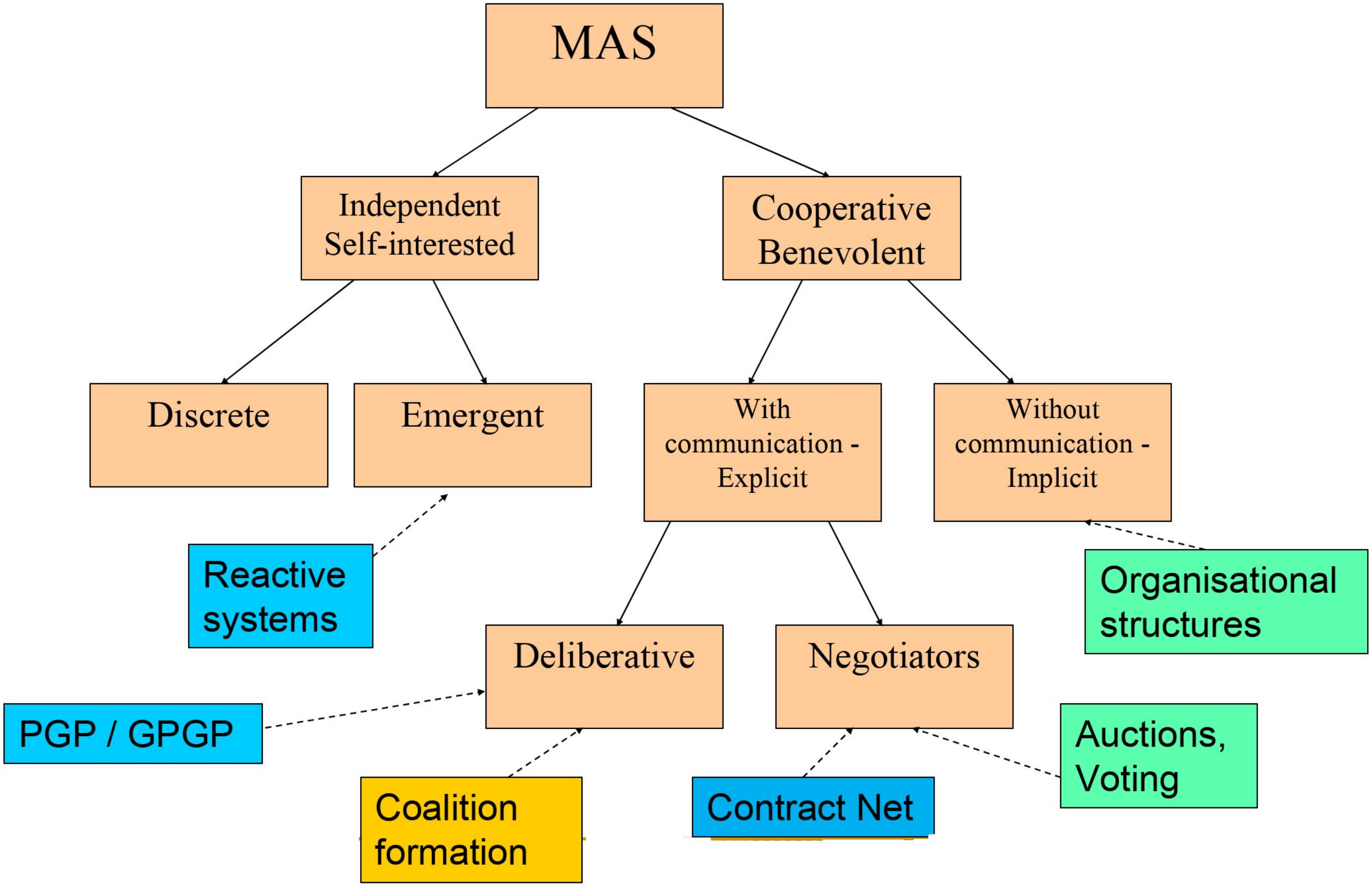
Universitat Rovira i Virgili

MESIIA – Master's Degree in Computer Security Engineering and Artificial Intelligence
MAI – Master's Degree in Artificial Intelligence

Cooperation between deliberative agents

- Situation in which **collaborative agents actively exchange explicit messages** in order to coordinate their actions and pursue a global goal in an efficient way
- Last lecture: distributed planning mechanism via (Generalised) Partial Global Planning
- This lecture: coalition formation

Cooperation hierarchy



Outline

- Coalition formation
  - What is a coalition?
  - External/Internal algorithms
  - Task allocation via coalition formation
    - Disjoint coalitions
    - Overlapping coalitions
- Example: RETSINA

What is a Coalition?

- Coalitions are (temporary) collections of individuals working together for the purpose of achieving a task
- Coalition formation is the process whereby an agent decides to cooperate with other agents, because
  - The task cannot be performed by a single agent
  - The task can be performed more efficiently by several agents working together

Interesting aspects of coalitions

Agents usually bring different, complementary abilities/resources to the coalition –

They are not clones !

When the task is completed, the pay-off is distributed, the coalition is disbanded and agents continue to pursue their own agendas

Issues in coalition formation

- Given a set of tasks and a set of agents, which coalitions should an agent attempt to form?
- What mechanism can an agent use for coalition formation?
- What guarantees regarding efficiency and quality can the mechanism provide?
- Once a coalition has been defined, how should its members distribute the work/pay-off?

Solution types

- Mechanisms for benevolent agents are usually much simpler than those for self-interested agents
- A centralised design of coalitions is usually much simpler to execute than a distributed one, but it may also be more computationally expensive
- Coalition formation may be *external* or *internal*

External coalition formation

- By imposition: an external agency makes decisions
- Agents advertise skills (capabilities) and prices (cost)
- Requester defines properties of coalition to the external agency
- An entity external to the MAS computes the optimal coalition

Internal coalition formation

- By self-organisation: coalitions are established by group interactions
- Multi-lateral negotiation of tasks and outcomes
- Identification of tasks to be solved
  - Can be static (given by the user at the beginning) or dynamic (generated by the own agents at run-time)

Coalition formation activities

- Coalition value calculation
  - Calculate the benefit of each coalition for each task
- Coalition structure generation
  - Decide the coalitions to form, maximizing the coalition value
- Pay-off distribution
  - Deciding how to distribute the pay-off between coalition members (equally, outputs, role)

Coalition structure generation

- Partition the set of agents into exhaustive [disjoint] coalitions
- Given 3 agents {a1, a2, a3}, there are seven possible coalitions

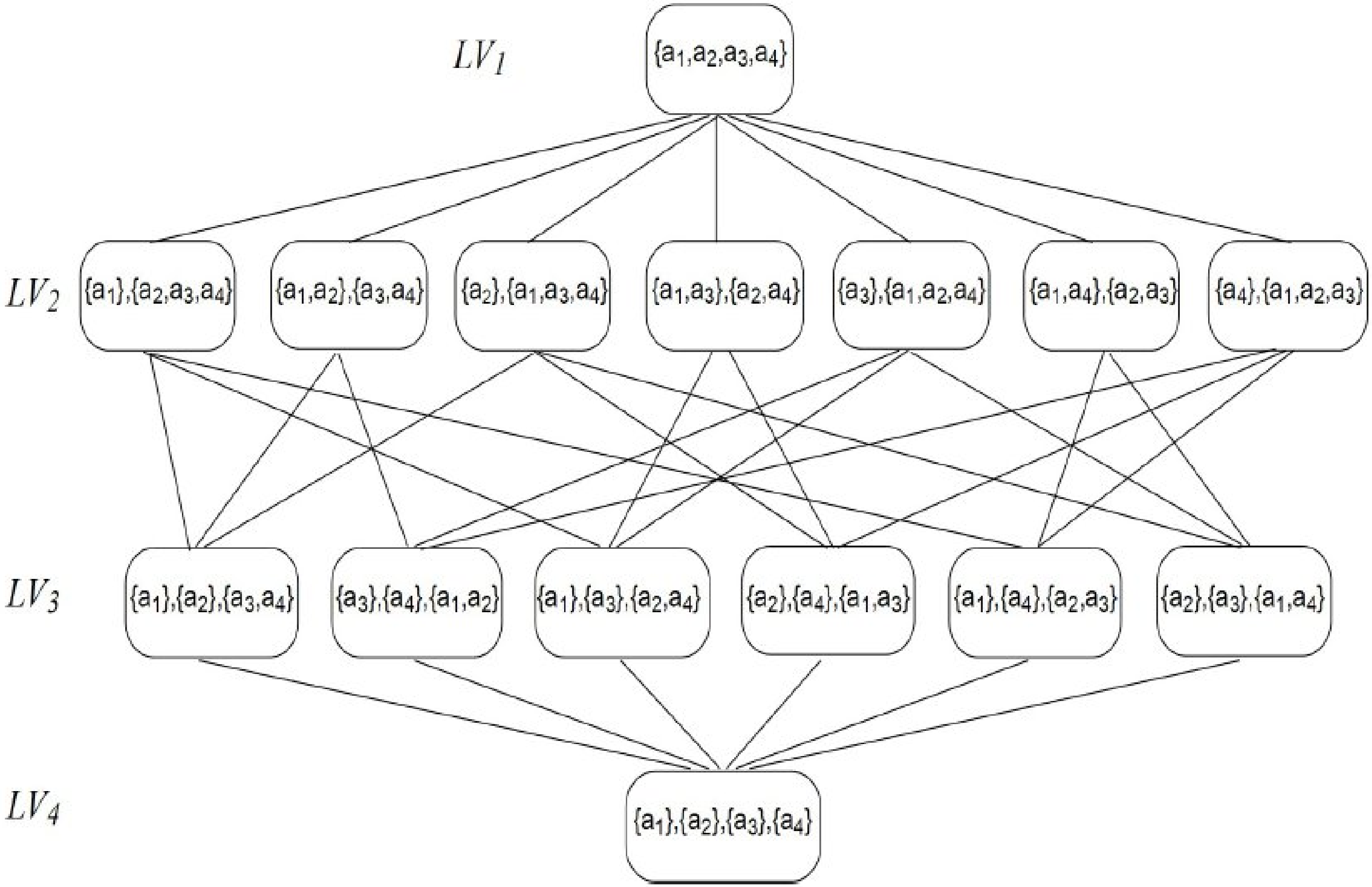
{a1}, {a2}, {a3}, {a1, a2}, {a1, a3}, {a2, a3}, {a1, a2, a3}

and five coalition structures

{{a1, a2, a3}}, {{a1}, {a2, a3}}, {{a2}, {a1, a3}},
{{a3}, {a1, a2}}, {{a1}, {a2}, {a3}}

- It may not be necessary or appropriate to generate all coalition structures in advance, depending on the application domain

Coalition structure generation



20 agents? -> 51,724,158,235,372 coalition structures

Task allocation via coalition formation

- There is a set of tasks that have to be distributed among a group of agents
- Each task has a certain set of requirements, and each agent has a certain set of capabilities
- The aim is to find an efficient distribution of tasks to subgroups of agents

"Optimal" distribution of tasks?

- Optimality could be defined in different ways
  - Minimum number of coalitions
  - Minimum size of coalitions
  - Best fit between tasks requirements and the capabilities of coalitions
  - Maximum benefit obtained by coalitions
  - Assignment of all tasks
  - Balanced work of all agents
  - …

Definition of capabilities in agents

- A is a set of *n* agents: $A=\{A_1,A_2,\ldots,A_n\}$
- Each agent A_i has a positive vector of *r* capabilities

  $$B_i=\langle b^i_1,\ldots,b^i_r\rangle$$

- Each capability is a *property* that *quantifies* the agent's ability in some aspect
- Capabilities may be expendable (e.g., amount of material of a certain type) or non-expendable (e.g., ability to perform an action)
- There is an evaluation function for each capability, that transforms it into monetary units

Definition of capabilities for tasks

- Set of *m* independent tasks $T=\{t_1, \ldots, t_m\}$
- There is a vector of *r* capabilities needed to perform each task t_l, $B_l=\{b^l_1, \ldots, b^l_r\}$
- The benefit gained from performing each task depends on the capabilities required for its performance
- To simplify the problem, the benefit can be computed with a linear function on the amount of resources

Simplifying conditions on coalitions

- A coalition is a group of agents that decide to cooperate to achieve a common task
- A coalition can work on a single task at a time
- Each agent can only belong to one coalition at a time
- A coalition C has a vector of capabilities B_c (sum of the capabilities of the agents in the coalition)
- A coalition *C* can perform a task *t* iff

for all $1 <= i <= r \quad b^t_i <= b^C_i$

Coalitional cost

- **For each coalition C and specific task *t***, it is possible to calculate the *coalitional value* *V*, that measures the joint utility that the members of C can reach if they cooperate to satisfy *t*. This value depends on the capabilities contributed by the team members and the number of coalition members
- The coalitional cost *c* is the *reciprocal* of the coalitional value
- The aim is to maximise the coalitional value (i.e., minimise the cost)

Task allocation process

- Heuristic: prefer small-sized coalitions, to reduce the number of coalitions to consider and the communication and coordination costs
  - Maximum coalitional size allowed: *k*
- Initial coalitional state: *n* single agents
- Step by step, formation of coalitions
  - 1 new coalition in each iteration
- When an agent joins a coalition, it quits the coalition formation process

*Problem: calculate, **for each possible coalition and task**, the coalitional value*

Algorithm of task allocation

- Preliminary stage: all possible coalitions are distributed among all agents
  - Collective, not centralised process
- Iterative stage: assign a task to the best coalition
  - Coalitional values for each pair <coalition,task> are (re)calculated
  - One coalition C is formed
  - Agents in coalition C quit the coalition formation process

Stage 1: initial calculations of agent A_i (I)

- L_i = empty_set
  This list will contain, at the end of stage 1, the coalitions whose value agent A_i should calculate
- P_i = all combinations up to *k* agents containing agent A_i
- *Those are all the possible coalitions in which agent i could participate*

Stage 1: initial calculations of agent A_i (II)

- For each combination in P_i do
  - Contact agent *j*, which is in the combination (i.e., *j* is in some possible coalition(s) with *i*)
  - In the first contact, request capabilities B_j
  - Commit to calculate the value of some of the coalitions including *i* and *j* = S_{ij}
  - $P_i = P_i - S_{ij}$
  - $L_i = L_i + S_{ij}$
- End_for

[at the same time]

For each agent *k* that contacts *i*, $P_i = P_i - S_{ki}$

Stage 1: End

- All the agents have (collectively) distributed all possible coalitions
- Each agent *i* has a list L_i of coalitions whose value it has to (repeatedly) calculate

A1

A1
A1,A2
A1,A3
A1,A4
A1,A2,A3
A1,A2,A4
A1,A3,A4
A1,A2,A3,A4

A2

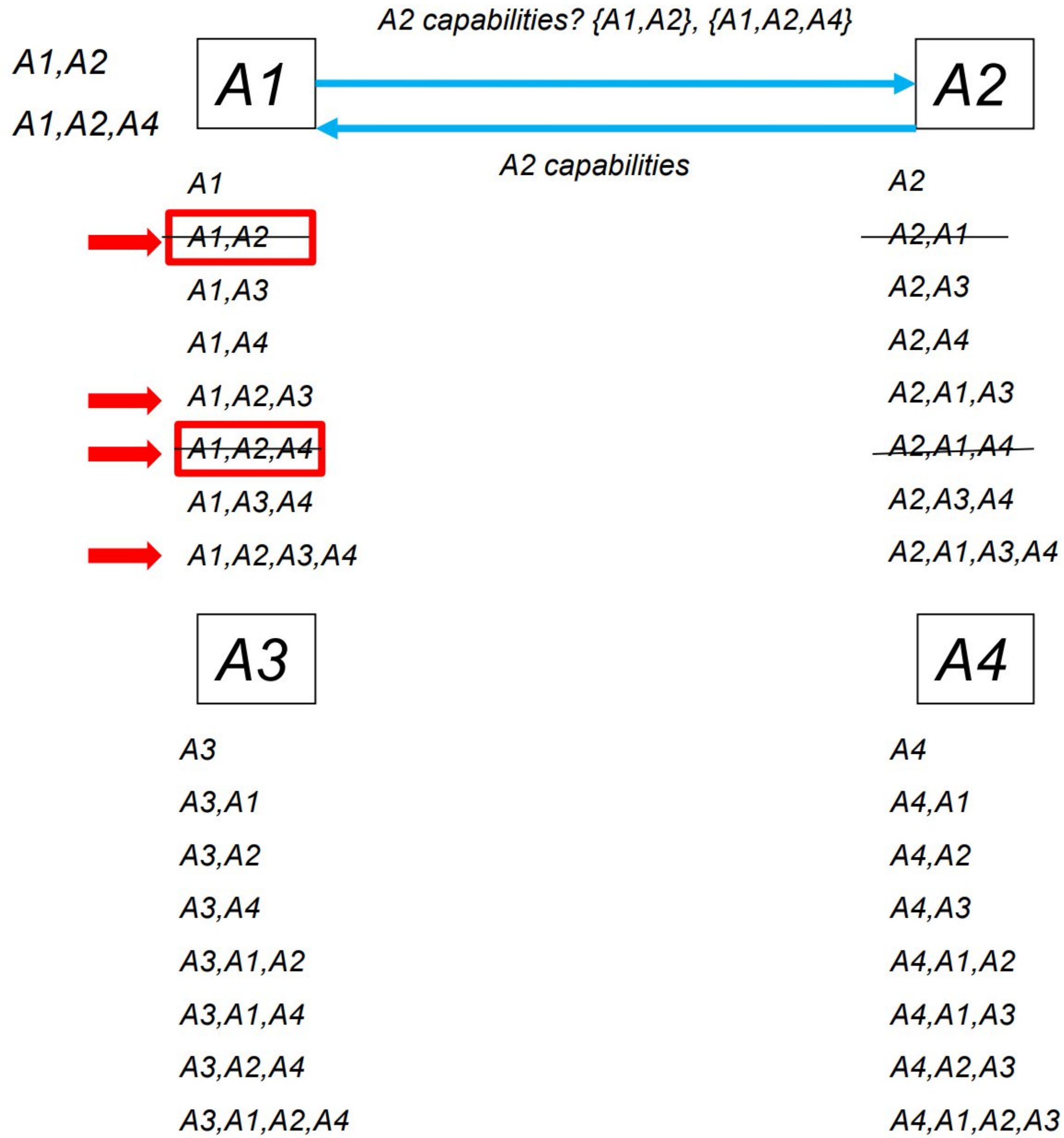
A2
A2,A1
A2,A3
A2,A4
A2,A1,A3
A2,A1,A4
A2,A3,A4
A2,A1,A3,A4

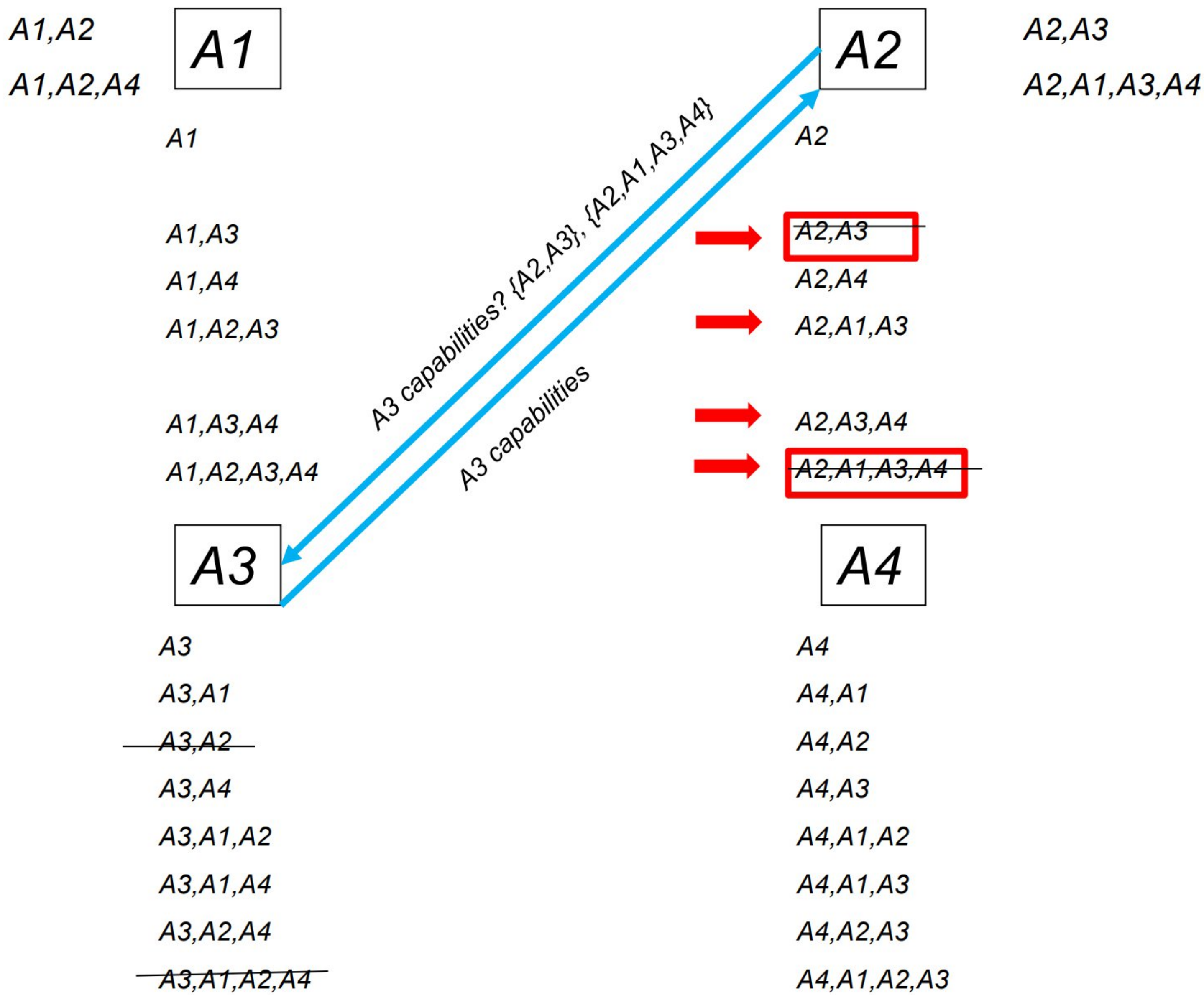
A3

A3
A3,A1
A3,A2
A3,A4
A3,A1,A2
A3,A1,A4
A3,A2,A4
A3,A1,A2,A4

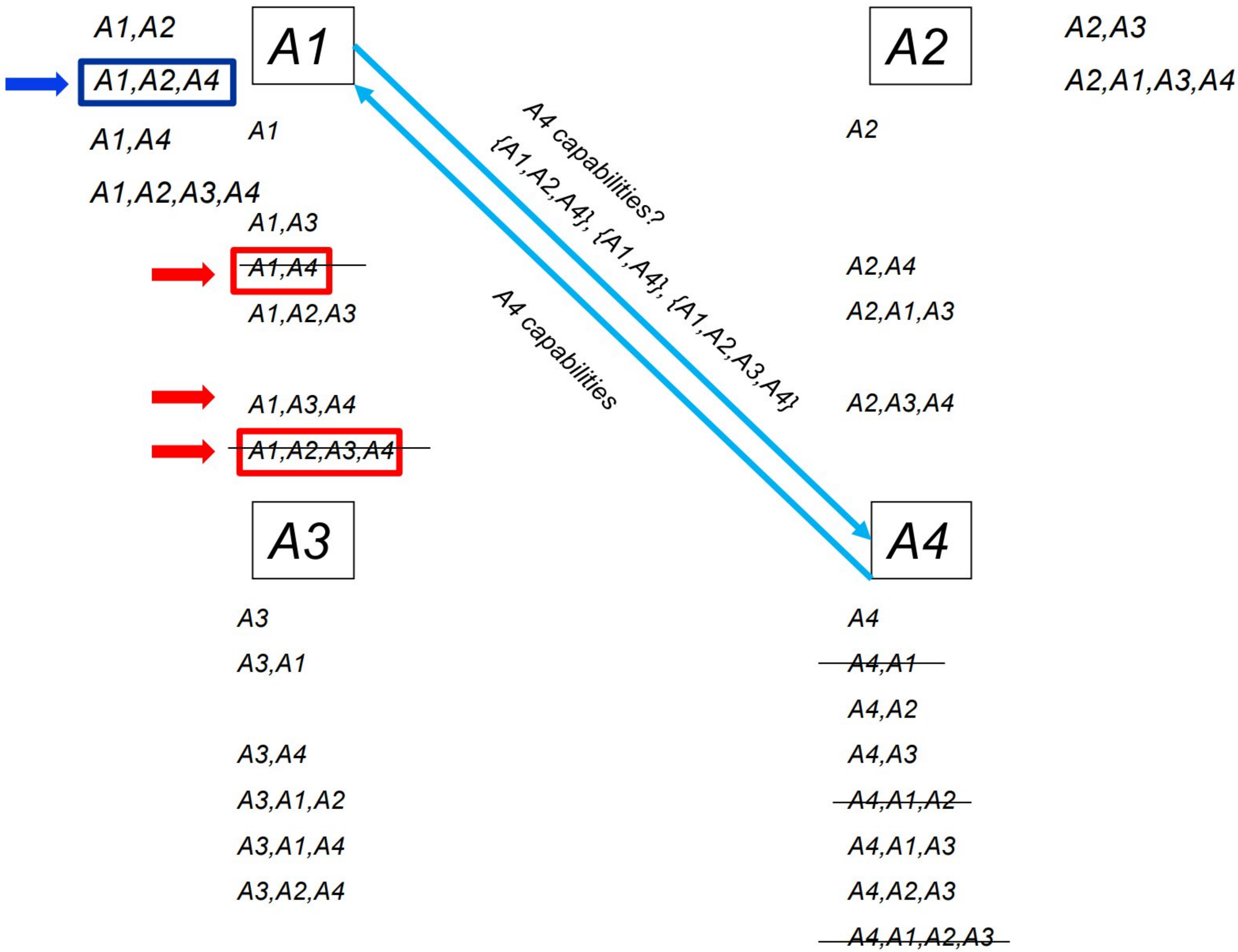
A4

A4
A4,A1
A4,A2
A4,A3
A4,A1,A2
A4,A1,A3
A4,A2,A3
A4,A1,A2,A3


A2 capabilities? {A1,A2}, {A1,A2,A4}
A1,A2
A1,A2,A4
A1
A2
A2 capabilities
A1
~~A1,A2~~
A1,A3
A1,A4
A1,A2,A3
~~A1,A2,A4~~
A1,A3,A4
A1,A2,A3,A4
A2
~~A2,A1~~
A2,A3
A2,A4
A2,A1,A3
~~A2,A1,A4~~
A2,A3,A4
A2,A1,A3,A4
A3
A4
A3
A3,A1
A3,A2
A3,A4
A3,A1,A2
A3,A1,A4
A3,A2,A4
A3,A1,A2,A4
A4
A4,A1
A4,A2
A4,A3
A4,A1,A2
A4,A1,A3
A4,A2,A3
A4,A1,A2,A3



A1,A2
A1,A2,A4
A1
A1,A3
A1,A4
A1,A2,A3
A1,A3,A4
A1,A2,A3,A4
A2,A3
A2,A1,A3,A4
A2
~~A2,A3~~
A2,A4
A2,A1,A3
A2,A3,A4
~~A2,A1,A3,A4~~
A3 capabilities? {A2,A3}, {A2,A1,A3,A4}
A3 capabilities
A3
A3,A1
~~A3,A2~~
A3,A4
A3,A1,A2
A3,A1,A4
A3,A2,A4
~~A3,A1,A2,A4~~
A4
A4,A1
A4,A2
A4,A3
A4,A1,A2
A4,A1,A3
A4,A2,A3
A4,A1,A2,A3



A1
A2
A3
A4
A1,A2
A1,A2,A4
A1,A4
A1,A2,A3,A4
A1
A1,A3
~~A1,A4~~
A1,A2,A3
A1,A3,A4
~~A1,A2,A3,A4~~
A2,A3
A2,A1,A3,A4
A2
A2,A4
A2,A1,A3
A2,A3,A4
A4 capabilities?
{A1,A2,A4}, {A1,A4}, {A1,A2,A3,A4}
A4 capabilities
A3
A3,A1
A3,A4
A3,A1,A2
A3,A1,A4
A3,A2,A4
A4
~~A4,A1~~
A4,A2
A4,A3
~~A4,A1,A2~~
A4,A1,A3
A4,A2,A3
~~A4,A1,A2,A3~~

A1,A2
A1,A2,A4

A1,A4
A1,A2,A3,A4

A1

A1

A1,A3

A1,A2,A3

A1,A3,A4

A2

A2,A3
A2,A1,A3,A4

A2

A2,A4
A2,A1,A3

A2,A3,A4

A3

A3
A3,A1

A3,A4
A3,A1,A2
A3,A1,A4
A3,A2,A4

A4

A4

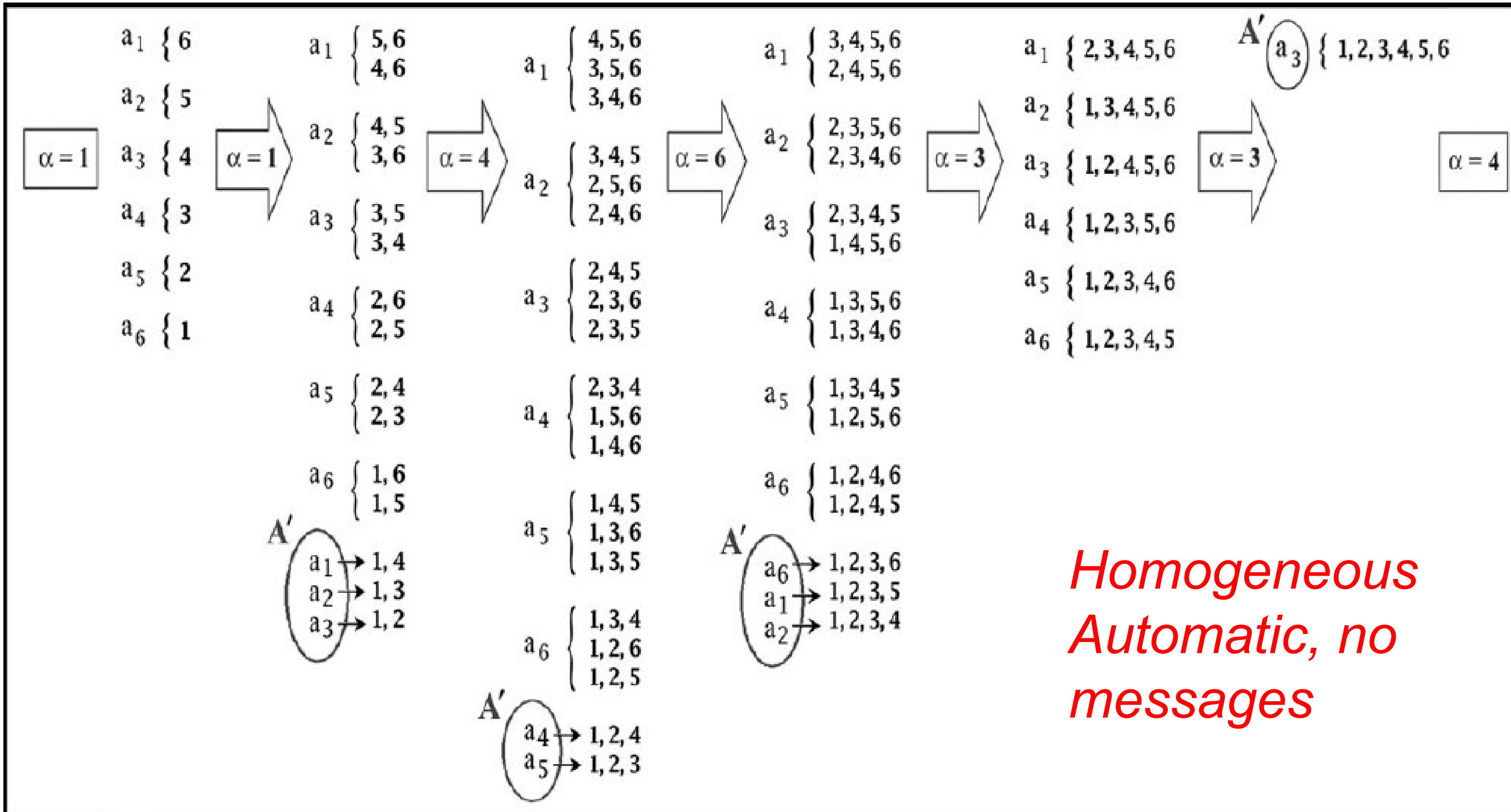
A4,A2
A4,A3

A4,A1,A3
A4,A2,A3

Drawbacks of this distribution

- It may not be homogeneous
  - An agent may have many more coalitions to analyse than another.
- The distribution may not be perfect
  - For example, the same coalition may be taken by two agents at the same time.
- It requires a heavy exchange of messages among the agents, who have to select coalitions iteratively

Other distribution possibilities – Rahwan & Jennings 07



Homogeneous
Automatic, no messages

Stage 2a: calculate coalitional values (I)

- $L_i^{cr} = L_i$ (in the first iteration)
- For each coalition C in L_i^{cr} do

  E_c = empty set

  $B_c = \text{sum}(B_k)$ for all agents *k* in *C*

  [potential capability of coalition C]

  For each pending task t_j, do

    Check if coalition *C* can do task t_j ($B_c >= B_j$)

    If it can, calculate the net benefit of t_j for *C*, e_j

      e_j = sum of market value of capabilities needed in t_j

      – sum of the capabilities costs

      – internal coordination costs

    $E_c = E_c + \{(t_j, e_j)\}$

  End for

Stage 2a: calculate coalitional values (II)

E_c contains the benefits that coalition C can gain from each of the tasks it is capable of performing

(t_c^{best}, V_c) = max-value(E_c)

Coalitional value of C and its related task

$c_c = 1 / V_c$ *Coalitional cost of C*

End for

- At the end of this step, each agent has calculated the best coalitional value/cost for each coalition, and it knows the most profitable task for each coalition

For each coalition C, $[t_c^{best}, V_c, c_c]$

Example

- Let's take
- $L_1 = \{ \{A_1, A_2\}, \{A_1, A_4\}, \{A_1, A_2, A_4\}, \{A_1, A_2, A_3, A_4\}\}$

- A_1 calculates the best value (and cost) of each coalition in L_1, and its associated task
  - $V_{12}, V_{14}, V_{124}, V_{1234}$
  - $c_{12}, c_{14}, c_{124}, c_{1234}$
  - $t_{12}^{best}, t_{14}^{best}, t_{124}^{best}, t_{1234}^{best}$

Example

	t1	*t2*	*t3*	*v*	*c*	*best*
A1,A2						
A1,A2,A4						
A1,A4						
A1,A2,A3,A4						

Example

	t1	t2	t3	v	c	best
A1,A2	-	10	5			
A1,A2,A4	15	18	27			
A1,A4	-	-	20			
A1,A2,A3,A4	18	15	17			

Example

	t1	t2	t3	v	c	best
A1,A2	-	10	5	10	1/10	t2
A1,A2,A4	15	18	27	27	1/27	t3
A1,A4	-	-	20	20	1/20	t3
A1,A2,A3,A4	18	15	17	18	1/18	t1

Stage 2b: form one coalition (I)

- The *weight* w_p of a coalition C_p is defined as
- $$w_p = c_p / |C_p|$$
- C_i = coalition of L_i with minimum weight

  Best coalition for agent *i*
- Each agent announces publicly the weight of its best coalition
- w_{low} = minimum of the announced weights

  Best coalition for all agents $[C_{low}, t_{low}^{best}]$

Stage 2b: form one coalition (II)

- If I am an agent in C_{low}, join the other agents in C_{low} to perform task t_{low}^{best}
- Delete the members of C_{low} from the list of candidates to future coalitions
- $L_i = L_i$ – coalitions with agents in C_{low}
- $T = T - t_{low}^{best}$
- L_i^{cr} = coalitions of L_i whose value has to be recalculated
  - Those coalitions whose previous best value was making t_{low}^{best}

Example

	t1	t2	t3	v	c	best	weight
A1,A2	-	10	5	10	1/10	t2	1/20
A1,A2,A4	15	18	27	27	1/27	t3	1/81
A1,A4	-	-	20	20	1/20	t3	1/40
A1,A2,A3,A4	18	15	17	18	1/18	t1	1/72

- *For agent 1, its best coalition is {A1,A2,A4}, with a cost 1/81 to solve t3*
- *Each agent would make public its best coalition*
- *If agent 1 wins (the cost 1/81 is smaller than the one found by the other agents for their coalitions), agents A1,A2 and A4 will work together in task 3*
- *The rest of the agents would now have to decide how to make coalitions to solve the remaining tasks*

Example

	t1	t2	t3	v	c	best	weight
A1,A2	-	10	5	10	1/10	t2	1/20
A1,A2,A4	15	18	27	27	1/27	t3	1/81
A1,A4	-	-	20	20	1/20	t3	1/40
A1,A2,A3,A4	18	15	17	18	1/18	t1	1/72

- *Imagine that agent 1 is not the winner, because agent A3 has found out that it can solve task t3 on its own (in the singleton coalition {A3}) with a better cost*

	t1	t2	v	c	best	weight
A1,A2	-	10	10	1/10	t2	1/20
A1,A2,A4	15	18	18	1/18	t2	1/54
A1,A4	-	-	-	-	-	-

  - *Now the best option for agent 1 would be to work together with agents 2 and 4 to solve task t2*

General comments

- Decentralised mechanism, but …
- It requires many (large) messages between the agents (an agent may need to contact another agent several times)
- The value of the same coalition may be calculated by different agents
- Large space of memory needed to store all possible coalitions of an agent
- No guarantees on the quality of the distribution. Tasks could be left unassigned !

Optimality?

- 2 tasks: t1, t2. 5 agents: a1, a2, a3, a4, a5
- Benefit for task 1:
  - {a1, a2, a3} => 30, {a1, a2, a4} => 29
- Benefit for task 2:
  - {a4, a5} => 10, {a3, a5} => 20
- Greedy algorithm: {a1, a2, a3} for task 1, {a4, a5} for task 2, 30 + 10 = 40 total benefit
- Optimal solution: {a1, a2, a4} for task 1, {a3, a5} for task 2, with a total benefit 29+20=49

Overlapping coalitions

- The previous algorithm formed disjoint coalitions because agents left the allocation process once they had been assigned to a coalition
- It could be possible to adapt the same task allocation algorithm to allow overlapping coalitions (an agent could participate in several coalitions, as long as it has enough resources)

Stage 2a': calculate coalitional values (I)

- $L_i^{cr} = L_i$ (in the first iteration)
- For each coalition C in L_i^{cr} do

  E_c = empty set

  $B_c = \text{sum}(B_k)$ for all agents *k* in *C*

  [potential capability of coalition C]

  For each pending task t_j, do

    Check if coalition *C* can do task t_j ($B_c >= B_j$)

    If it can, calculate the net benefit of t_j for C, e_j

      e_j = sum of market value of capabilities needed in t_j

      – sum of the capabilities costs

      – internal coordination costs

    $E_c = E_c + \{(t_j, e_j)\}$

  End for

Stage 2a': calculate coalitional values (II)

E_c contains the benefits that coalition C can gain from each of the tasks it is capable of performing

(t_c^{best}, V_c) = max-value(E_c)

Coalitional value of C and its related task

$c_c = 1 / V_c$ *Coalitional cost of C*

End for

- At the end of this step, each agent has calculated the best coalitional value/cost for each coalition, and it knows the most profitable task for each coalition

For each coalition C, $[t_c^{best}, V_c, c_c]$

Stage 2b': form one coalition (I)

- C_i = coalition of L_i with minimum cost

  Best coalition for agent *i*
- Each agent announces publicly the cost of its best coalition
- w_{low} = minimum of the announced costs

  Best coalition for all agents $[C_{low}, t_{low}^{best}]$

Stage 2b': form one coalition (II)

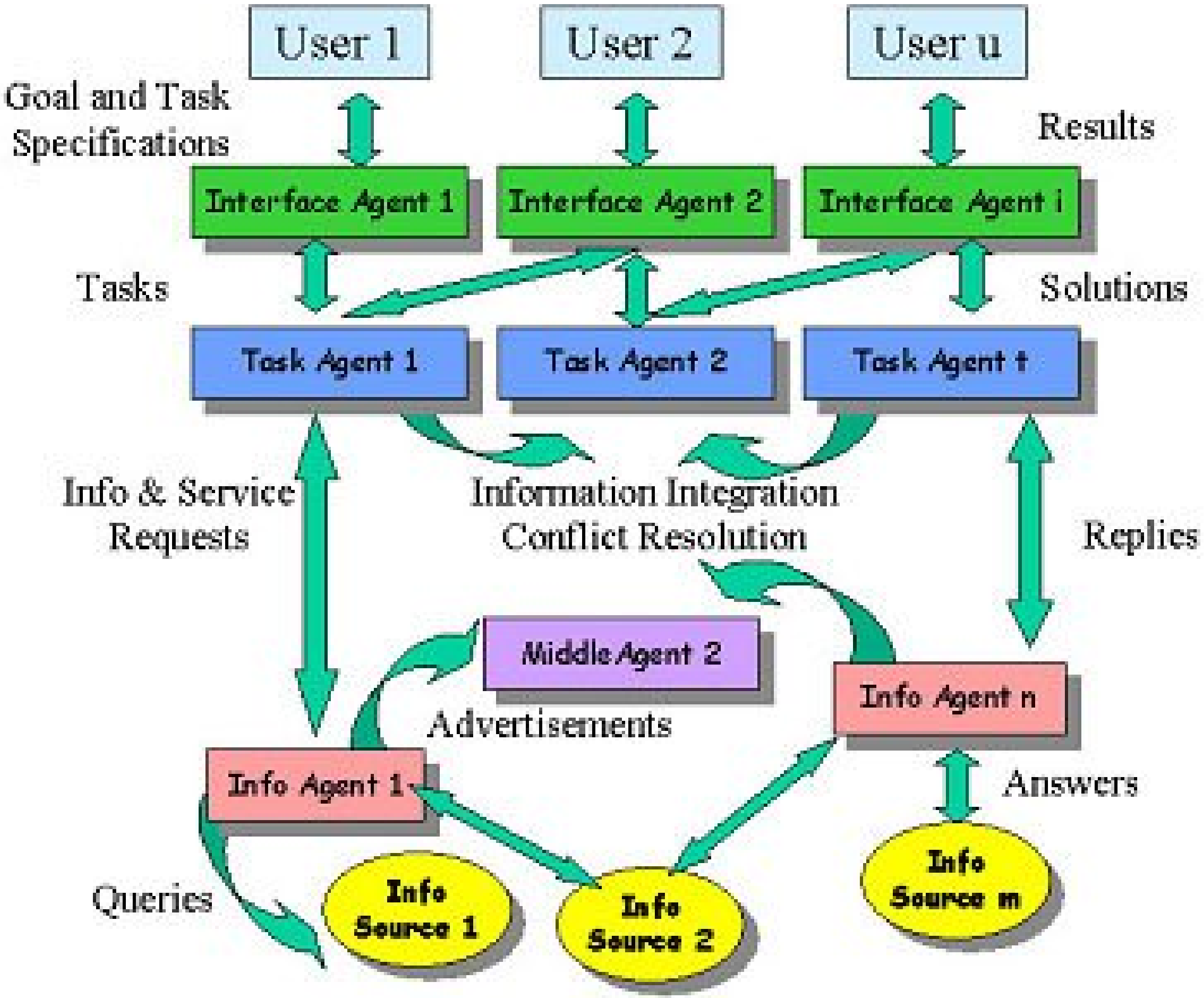
- If I am an agent in C_{low}, join the other agents in C_{low} to perform task t_{low}^{best}
- ~~Delete the members of C_{low} from the list of candidates to future coalitions~~
- ~~$L_i = L_i$ – coalitions with agents in C_{low}~~
- $T = T - t_{low}^{best}$
- Update the capability-vectors of all the members of C_{low} (expendable capabilities) according to their contribution to the execution of the task t_{low}^{best}
- L_i^{cr} = coalitions of L_i whose value has to be recalculated
  - Those coalitions that have an agent with changed capabilities or whose previous best value was making t_{low}^{best}

Example

	t1	t2	t3	v	c	best
A1,A2	-	10	5	10	1/10	t2
A1,A2,A4	15	18	27	27	1/27	t3
A1,A4	-	-	20	20	1/20	t3
A1,A2,A3,A4	18	15	17	18	1/18	t1

- In the previous example, in the first iteration the coalition {A1,A2,A4} would be assigned to task 3
- After eliminating the resources spent in this task, these agents could still participate in other coalitions to solve tasks 1 or 2

Example: RETSINA



Example: RETSINA – New task

- New task t_j arrives at Task Agent A_i (from a user or from another agent)
- Adds t_j to its task list, T_i.
- Decomposes t_j in subtasks t_{j1}, t_{j2}, ..., t_{jl} (using a predefined *task reduction library*)
- Asks a Middle Agent about the agents that can solve each subtask

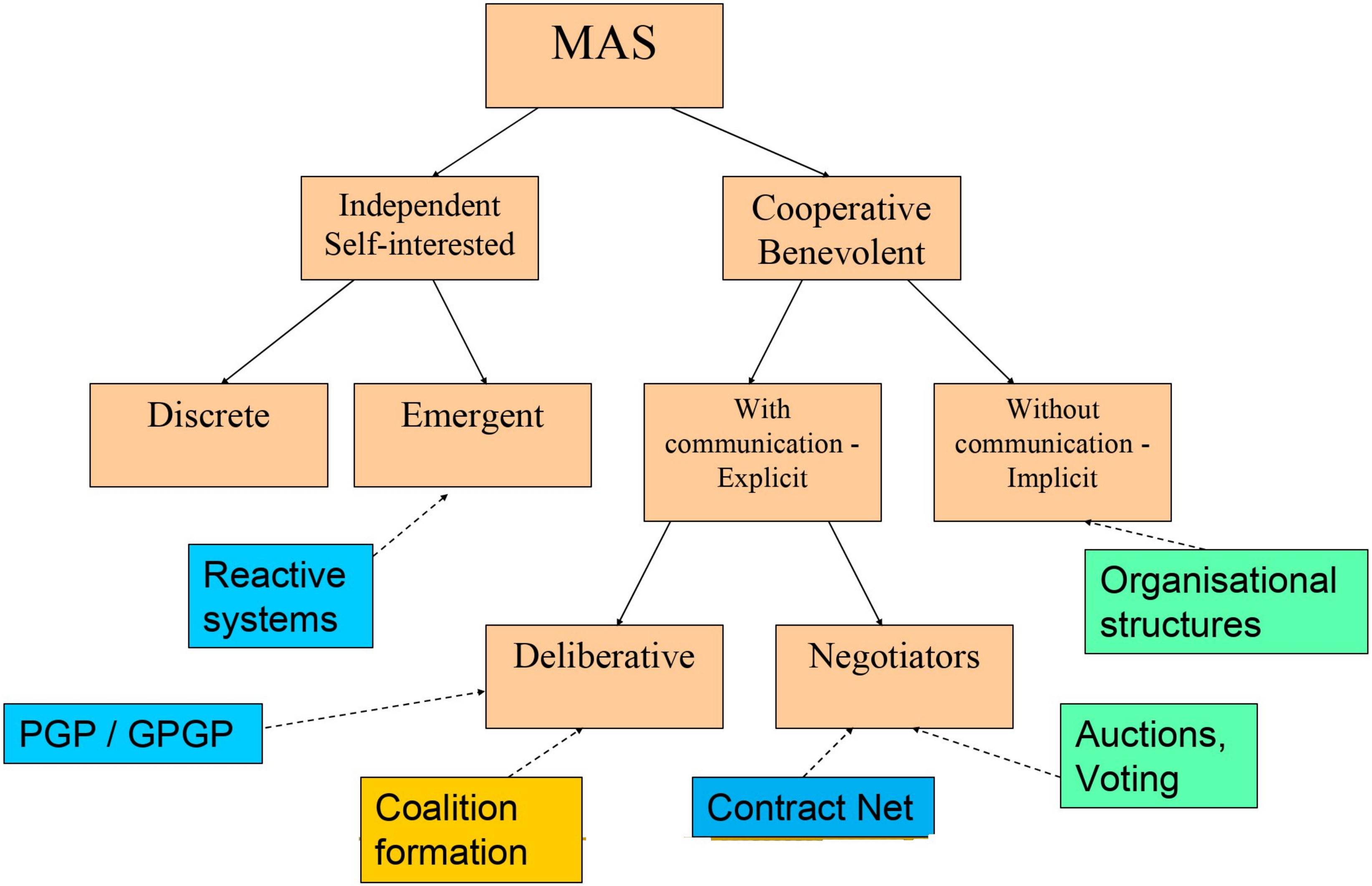
Example: RETSINA – Coalition formation

- For each task t_j in T_i, Task Agent A_i considers the possible coalitions that can be formed to perform it (formed by the agents that can solve the subtasks) and computes their coalition value
- The coalition C_j with the maximum value is sent to the Middle Agent, and the agents in this coalition are contacted to work together and solve the associated subtasks

Example: RETSINA – Coalition formation

- Once all the agents in the coalition have completed their subtasks and reported to A_i, A_i removes task t_j from its list of tasks T_i
- Each of the participating agents reports any change in their expendable capabilities to the Middle Agent
- Agent A_i asks to the Middle Agent about any change in the capabilities of the other agents, and, if T_i is not empty, it starts again the coalition formation process

Cooperation hierarchy



Proposed readings

- Article "*Methods for task allocation via agent coalition formation*". Shehory, Kraus.
- Article "*Coalition structure generation: a survey*" (2015). Rahwan, Michalak, Wooldridge, Jennings (also Rahwan video)
- Chapter 13 of the book *An introduction to MultiAgent Systems* (M. Wooldridge), 2nd ed.
- PhD thesis: *Algorithms for coalition formation in MAS* (Rahwan, 2007)