CBAA: Consensus-Based Auction Algorithm:

N_t : number of tasks

N_u : number of agents

x_{ij} = local task assignment to the agent :

y_{ij} = local winning bid list

c = agent's bid on each resource. → euclidean distance between agent and task.

Select Task : Phase 1 → Allocation:

→ For each Agent loop.

if sum(x_{ij}) = 0 : ▹ meaning the agent is to assigned yet

h_i = c > y_{ij} ▹ h_i is the list of valid tasks, they are valid if the bid is smaller than the winning bid.

if $h_i \neq 0$: ▹ meaning if there is at least one true element in h_i

J : argmax c · h_i ▹ selecting the task with the highest bid.

x_{ij}[J] = 1 ▹ assigning to the agent the task J.

y_{ij}[J] = c[J] ▹ updating the winning bid list with the highest bid in c at index J.

Update task : Phase 2 : Consensus Process:

→ For each agent Loop.

→ Given a topology, agents communicate or not. (star, fc, square, central etc...)

send Y_i to k if agents are connected.

receive y_k if agents are conected.

```
x_prev = xij.copy()
neigh_ids = list(y.keys())
neigh_ids.insert(0, self.id).

all_bids = np.array(list(y.values()))

yij = all_bids.max(0)

winner_agent_id = np.argmax(all_bids[:, j])
z = neighbor_id[winner_agent_id]

if z ≠ self:
    xij[j] = 0 // unassign

if x_prev = xij => converged → all tasks are assigned.
```

Auction

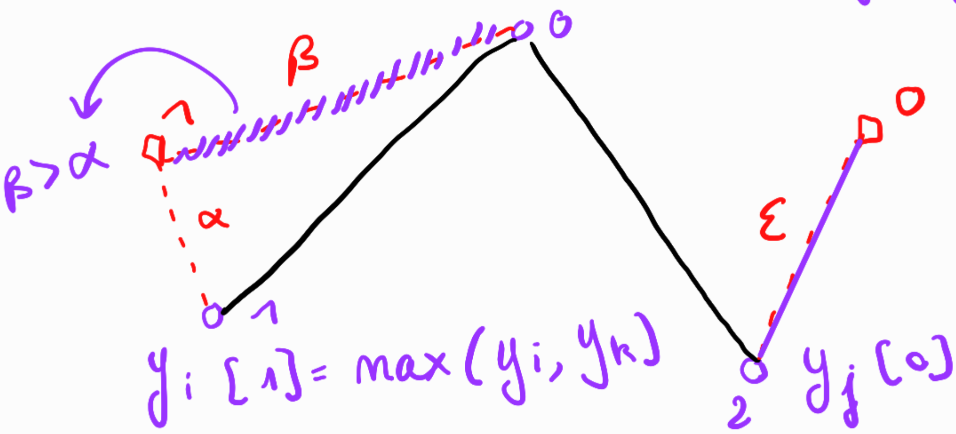
Consensus

$c_k = \beta \quad y_k = [\varepsilon, \beta]$

$x_k = [0, 1]$

$y_k[1] = \max(y_k, y_j, y_i)$

□ Tasks
○ Agents



$y_i[1] = \max(y_i, y_k)$ $\qquad$ $y_j[0] = \max(y_k, y_j)$

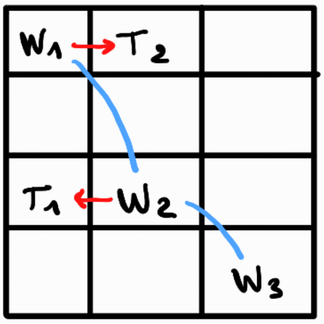
$x_i = [0, 1]$ $\qquad$ $x_j = [1, 0]$

$y_i = [0, \beta]$ $\qquad$ $c_j = \varepsilon$

$c_i = \alpha$ $\qquad$ $y_j = [\varepsilon, 0]$

$\text{all\_bid} = [\beta, \alpha, \varepsilon]$, $\text{winner\_agent} = \text{argmax}[\text{all\_bid}]$

CBAA Toy Example: Single Assignement



— : communication

environment:

Let W be the set of agents

$W = \{W_1, W_2, W_3\}$

Let T be the set of tasks

$T = \{T_1, T_2\}$

Let x_{ij} the assignement matrix :

x_{ij}	T_1	T_2
w_1	0	0
w_2	0	0
w_3	0	0

$x_{ij} = 1$ if agent i is assigned to task j

Let y_{ij} be the winning bid matrix:

y_{ij}	T_1	T_2
w_1	-inf	-inf
w_2	-inf	-inf
w_3	-inf	-inf

Let c_{ij} be the local bid matrix:

c_{ij}	T_1	T_2
w_1	-2	-1
w_2	-1	-2
w_3	-3	-4

(we suppose that the agents are fixed in space)

- c_{ij} is the manhattan distance between agent i and task j.

Objective function:

$$\max \sum_{i=1}^{|W|} \left(\sum_{j=1}^{|T|} c_{ij}(x_i, p_i)\, x_{ij} \right)$$

p: only in multi-assignement

- $\sum_{j=1}^{|W|} x_{ij} \leq 1, \ \forall_T \in T$

- $x_{ij} \in \{0, 1\}, \ \forall (i,j) \in T \times W$

i.e. we want to minimize the cost of task assignement for an agent. (here the manhattan distance).

Let G_{ik}, the adjacency matrix of the communication network.

G_{ik}	W_1	W_2	W_3
w_1	1	1	0
w_2	1	1	1
w_3	0	1	1

(we suppose G_{ik} to be constant in time)

$G_{ik} = 1$ if w_i and w_k are conected

CBAA Algorithmic trace:

Local agent information:

W_1 (T_1, T_2)

$x_1 = <0, 1>$

$y_1 = <-inf, -1>$

$c_1 = <-2, -1>$

W_2

$x_2 = <1, 0>$

$y_2 = <-1, -inf>$

$c_2 = <-1, -2>$

W_3

$x_3 = \boxed{<1, 0>}$

$y_3 = <-3, -inf>$

$c_3 = <-3, -4>$

$$h_{ij} = \begin{cases} 1 & \text{if } c_{ij} > y_{ij} \\ 0 & \text{else} \end{cases}$$

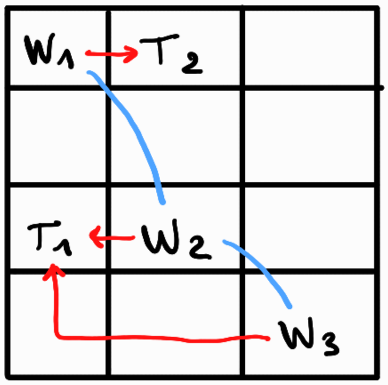
h_{ij}	T_1	T_2
W_1	1	1
W_2	1	1
W_3	1	1

1) $J_i = \text{argmax } h_{ij} \cdot c_{ij}$

$W_1 \to T_2$

$W_2 \to T_1$

$W_3 \to T_1$



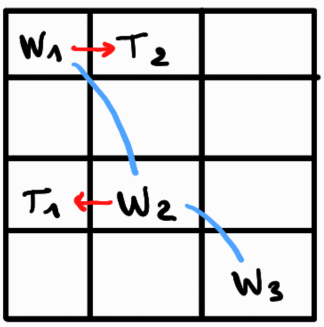
2) Consensus Phase: 2

W_1	W_2	W_3
$y_1 = <-inf, -1>$	$y_2 = <-1, -inf>$	$y_3 = <-3, -inf>$
$y_2 = <-1, -inf>$	$y_1 = <-inf, -1>$	$y_2 = <-1, -inf>$
$J_1 = T_2$	$y_3 = <-3, -inf>$	$J_3 = T_1$
	$J_2 = T_1$	

$y_1 = <-1, -1>$ | $y_2 = <-1, -1>$ | $y_3 = <-1, -inf>$

$z_1 = W_1$ | $z_2 = W_2$ | $z_3 = W_2$

$x_1 = <0, 1> = 1$ | $x_2 = <1, 0> = 1$ | $x_3 = <0, 0> =$



→ Consensus-Based Auction Algo.

→ Consensus-Based Bundle Algo.