Consensus-Based Decentralized Auctions for Robust Task Allocation

Written by: Han-Lim Choi, Luc Brunet, Jonathan P. How

Presented by: David Akagi

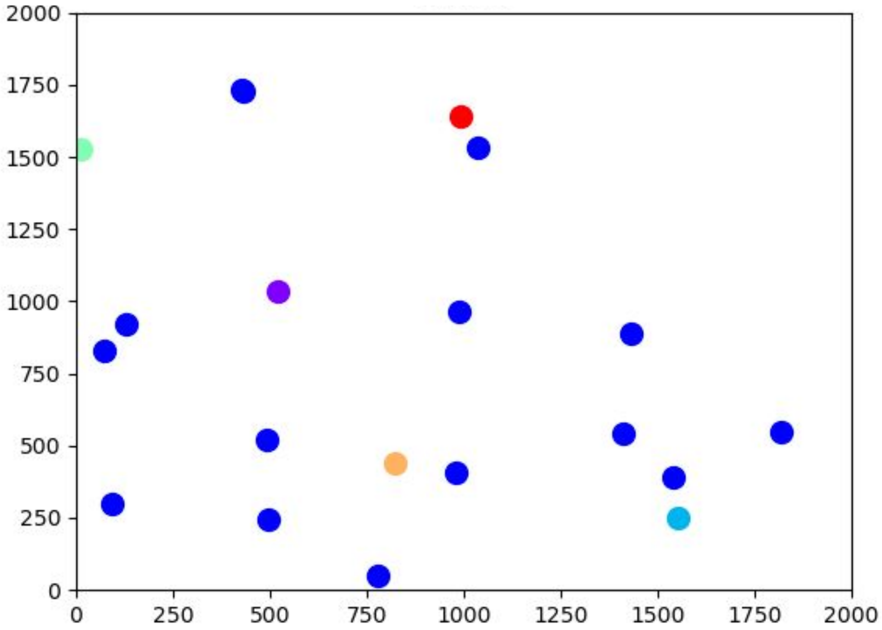
H. -L. Choi, L. Brunet and J. P. How, "Consensus-Based Decentralized Auctions for Robust Task Allocation," in *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912-926, Aug. 2009, doi: 10.1109/TRO.2009.2022423

Outline

- What is Task Allocation?
- Existing Approaches
- CBAA & CBBA
- Convergence Guarantees
- Optimality Guarantees
- Simulation Results

Task Allocation

- Multiple agents
- Multiple tasks to assign
- Find optimal (ish), conflict-free assignments
- Don’t take too long



Task Allocation Approaches

Centralized planner - all agents communicate with central hub

- Heavy processing done off-board agents
- Single point of failure
- Limited allowable distance from hub

C. Schumacher, P. Chandler, and S. Rasmussen, “Task allocation for wide area search munitions,” in Proc. Amer. Control Conf., 2002, pp. 1917– 1922

Decentralized planner - centralized planner on each agent

- No single point of failure
- Limited communication b/t agents can create conflicting assignments

T. W. McLain and R. W. Beard, “Coordination variables, coordination functions, and cooperative-timing missions,” *J. Guid., Control, Dyn.*, vol. 28, no. 1, pp. 150–161, 2005

Task Allocation Approaches

Consensus-based allocation - agents converge to agreement on situation before assigning tasks

- Robust to differences in situational awareness
- May require perfect consensus for conflict-free assignments
- Time intensive

D. Dionne and C. A. Rabbath, "Multi-UAV decentralized task allocation with intermittent communications: The DTC algorithm," in Proc. Amer. Control Conf., 2007, pp. 5406–5411.

Auction-based allocation - agents' bids for tasks are gathered by single "auctioneer"

- Sub-optimal assignments, but always conflict-free
- Robust to different situational awarenesses
- Agents must all be connected to auctioneer - limited possible topologies

B. Gerkey and M. Mataric, "Sold!: Auction methods for multirobot coordination," IEEE Trans. Robot. Autom., vol. 18, no. 5, pp. 758–768, Oct. 2002

Consensus-Based Auction (CBAA) and Consensus-Based Bundle Algorithms (CBBA)

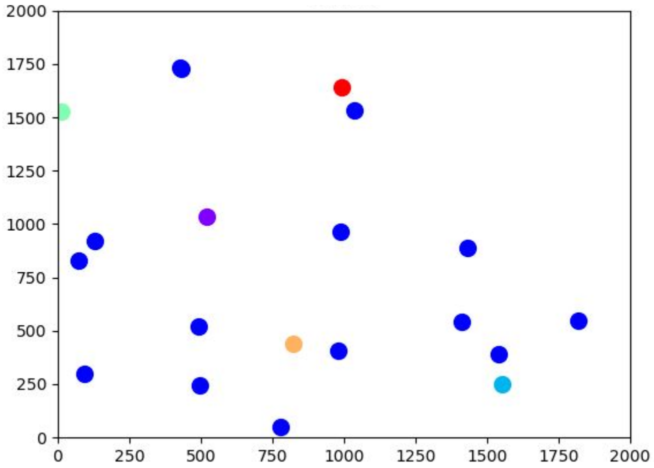
- Best of both worlds - consensus and auction
- Decentralized task selection (auction)
- Decentralized conflict resolutions (consensus)
- Consensus is on winning bids, not situational awareness

- CBAA for single-assignment problem
- CBBA for multi-assignment problem (generalized CBAA)

Why Bundle?

- Alternative to sequential auctions
- Bid on entire bundles of tasks
- Faster convergence, more logical grouping of tasks
- Enumerating all possible bundles is expensive

- CBBA - each agent builds single bundle, bids based on improvement task adds to bundle
- Faster convergence
- Guaranteed 50% optimality (worst case)



Consensus-Based Auction Algorithm (CBAA)

- N_t - tasks
- N_u - agents
- L_t - max number of tasks per agent (Lt = 1)
- c_ij - reward for agent i performing task j
- x_ij - 1 if agent i takes task j, 0 otherwise
- p_i - path of tasks assigned to agent i

$$\max \quad \sum_{i=1}^{N_u} \left(\sum_{j=1}^{N_t} c_{ij}(\mathbf{x}_i, \mathbf{p}_i) x_{ij} \right)$$

subject to

$$\sum_{j=1}^{N_t} x_{ij} \leq L_t \qquad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_u} x_{ij} \leq 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{i=1}^{N_u} \sum_{j=1}^{N_t} x_{ij} = N_{\min} \triangleq \min\{N_t, N_u L_t\}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in \mathcal{I} \times \mathcal{J}$$

CBAA - Phase 1: Auction

- Each agent places a bid on a task (asynchronously)
- Awards itself best valid task based on knowledge of current best bids

Algorithm 1 CBAA Phase 1 for agent i at iteration t

procedure SELECT TASK($\mathbf{c}_i, \mathbf{x}_i(t-1), \mathbf{y}_i(t-1)$)
    $\mathbf{x}_i(t) = \mathbf{x}_i(t-1)$
    $\mathbf{y}_i(t) = \mathbf{y}_i(t-1)$
    if $\sum_j x_{ij}(t) = 0$ **then**
        $h_{ij} = \mathbb{I}(c_{ij} > y_{ij}(t)), \ \forall j \in \mathcal{J}$
        if $\mathbf{h}_i \neq \mathbf{0}$ **then**
            $J_i = \mathrm{argmax}_j \, h_{ij} \cdot c_{ij}$
            $x_{i,J_i}(t) = 1$
            $y_{i,J_i}(t) = c_{i,J_i}$
        end if
    end if
end procedure

x_i	0	0	0	0	0
c_i	1.7	2.3	2.1	1.4	2.5
y_i	1.9	1.8	2.7	3.5	1.6

x_i	0	0	0	0	1

y_i	1.9	1.8	2.7	3.5	2.5

CBAA - Phase 2: Consensus

- Send/receive winning bids list to connected neighbors
- Agents lose task if outbid by a neighbor

Algorithm 2 CBAA Phase 2 for agent i at iteration t:

```
SEND y_i to k with g_ik(τ) = 1
RECEIVE y_k from k with g_ik(τ) = 1
procedure UPDATE TASK(g_i(τ), y_{k∈{k|g_ik(τ)=1}}(t) , J_i)
    y_ij(t) = max_k g_ik(τ) · y_kj(t), ∀j ∈ 𝒥
    z_{i,J_i} = argmax_k g_ik(τ) · y_{k,J_i}(t)
    if z_{i,J_i} ≠ i then
        x_{i,J_i}(t) = 0
    end if
end procedure
```

$\textsc{Send}\ \mathbf{y}_i$ to k with $g_{ik}(\tau) = 1$

$\textsc{Receive}\ \mathbf{y}_k$ from k with $g_{ik}(\tau) = 1$

procedure $\textsc{Update Task}(\mathbf{g}_i(\tau), \mathbf{y}_{k \in \{k | g_{ik}(\tau)=1\}}(t)\ ,\ J_i)$

$y_{ij}(t) = \max_k g_{ik}(\tau) \cdot y_{kj}(t),\ \forall j \in \mathcal{J}$

$z_{i,J_i} = \operatorname{argmax}_k g_{ik}(\tau) \cdot y_{k,J_i}(t)$

if $z_{i,J_i} \neq i$ **then**

$x_{i,J_i}(t) = 0$

end if

end procedure

x_i	0	0	0	0	1
y_i	1.9	1.8	2.7	3.5	2.5
y_k	1.9	2.3	1.2	3.5	3.1

y_i	1.9	2.3	2.7	3.5	3.1
x_i	0	0	0	0	0

CBAA Summary

- Rinse and repeat Phase 1 (Auction) and Phase 2 (Consensus) until convergence
- Converges to conflict-free single-task assignments

- Specific case of Consensus-Based Bundle Algorithm (CBAA) with L_t = 1

CBBA - Phase 1: Bundle Assignment

- y_i(t) - winning bids (according to agent i)
- z_i(t) - winning agents
- b_i(t) - agent i bundle of tasks (sequential)
- p_i(t) - agent i path of tasks (optimally ordered)

- c_ij - reward for agent i to do task j
- h_ij - can agent i bid higher than current highest bid for task j? (1 or 0)

$\mathbf{p}_i \oplus_n \{j\}$ - insert task j after task n in path p_i

$S_i^{\mathbf{p}_i}$ - total reward for agent i performing tasks in path p_i

$\max_{n \leq |\mathbf{p}_i|} S_i^{\mathbf{p}_i \oplus_n \{j\}}$ - maximum reward for adding task j at the best position n in path p_i

Algorithm 3 CBBA Phase 1 for agent i at iteration t:

procedure BUILD BUNDLE($\mathbf{z}_i(t-1)$, $\mathbf{y}_i(t-1)$, $\mathbf{b}_i(t-1)$)
  $\mathbf{y}_i(t) = \mathbf{y}_i(t-1)$
  $\mathbf{z}_i(t) = \mathbf{z}_i(t-1)$
  $\mathbf{b}_i(t) = \mathbf{b}_i(t-1)$
  $\mathbf{p}_i(t) = \mathbf{p}_i(t-1)$
  while $|\mathbf{b}_i| < L_t$ **do**
    $c_{ij} = \max_{n \leq |\mathbf{p}_i|} S_i^{\mathbf{p}_i \oplus_n \{j\}} - S_i^{\mathbf{p}_i}, \forall j \in \mathcal{J} \setminus \mathbf{b}_i$
    $h_{ij} = \mathbb{I}(c_{ij} > y_{ij}), \ \forall j \in \mathcal{J}$
    $J_i = \operatorname{argmax}_j c_{ij} \cdot h_{ij}$
    $n_{i,J_i} = \operatorname{argmax}_n S_i^{\mathbf{p}_i \oplus_n \{J_i\}}$
    $\mathbf{b}_i = \mathbf{b}_i \oplus_{\text{end}} \{J_i\}$
    $\mathbf{p}_i = \mathbf{p}_i \oplus_{n_{i,J_i}} \{J_i\}$
    $y_{i,J_i}(t) = c_{i,J_i}$
    $z_{i,J_i}(t) = i$
  end while
end procedure

CBBA - Phase 2: Conflict Resolution

- Agents send/receive to connected neighbors their
- (1) list of winning bids (y_i)
- (2) list of winning agents (z_i)
- (3) timestamps of last communication with other agents (s_i)

- Timestamps updated at each communication

$$s_{ik} = \begin{cases} \tau_r, & \text{if } g_{ik} = 1 \\ \max_{m:g_{im}=1} s_{mk}, & \text{otherwise} \end{cases}$$

- Follow communication rules for updating bids, winning agents list

TABLE 1
ACTION RULE FOR AGENT i BASED ON COMMUNICATION WITH AGENT k REGARDING TASK j

Agent k (sender) thinks z_{kj} is	**Agent i (receiver) thinks z_{ij} is**	**Receiver's Action (default: leave)**
k	i	if $y_{kj} > y_{ij} \rightarrow$ update
	k	update
	$m \notin \{i, k\}$	if $s_{km} > s_{im}$ or $y_{kj} > y_{ij} \rightarrow$ update
	none	update
i	i	leave
	k	reset
	$m \notin \{i, k\}$	if $s_{km} > s_{im} \rightarrow$ reset
	none	leave
$m \notin \{i, k\}$	i	if $s_{km} > s_{im}$ and $y_{kj} > y_{ij} \rightarrow$ update
	k	if $s_{km} > s_{im} \rightarrow$ update else $\rightarrow$ reset
	m	$s_{km} > s_{im} \rightarrow$ update
	$n \notin \{i, k, m\}$	if $s_{km} > s_{im}$ and $s_{kn} > s_{in} \rightarrow$ update if $s_{km} > s_{im}$ and $y_{kj} > y_{ij} \rightarrow$ update if $s_{kn} > s_{in}$ and $s_{im} > s_{km} \rightarrow$ reset
	none	if $s_{km} > s_{im} \rightarrow$ update
none	i	leave
	k	update
	$m \notin \{i, k\}$	if $s_{km} > s_{im} \rightarrow$ update
	none	leave

1) *update*: $y_{ij} = y_{kj}, z_{ij} = z_{kj}$;
2) *reset*: $y_{ij} = 0, z_{ij} = \emptyset$;
3) *leave*: $y_{ij} = y_{ij}, z_{ij} = z_{ij}$.

CBBA (and CBAA) Convergence Guarantees

- Diminishing Marginal Gain (DMG) - $c_{ij}[\mathbf{b}_i] \geq c_{ij}[\mathbf{b}_i \oplus_{\text{end}} \mathbf{b}]$
- Using DMG, CBBA gives same solution as the Sequential Greedy Algorithm (SGA)

- (SGA assumes full knowledge of all agents - no need for consensus)

Algorithm 4 Sequential greedy algorithm

$\mathcal{I}_1 = \mathcal{I},\ \mathcal{J}_1 = \mathcal{J}$
$\eta_i = 0,\ \forall i \in \mathcal{I}$
$c_{ij}^{(1)} = c_{ij}[\{\emptyset\}],\ \forall (i,j) \in \mathcal{I} \times \mathcal{J}$
for $n = 1$ to $N_{\min}$ **do**
  $(i_n^\star, j_n^\star) = \operatorname{argmax}_{(i,j) \in \mathcal{I} \times \mathcal{J}} c_{ij}^{(n)}$
  $\eta_{i_n^\star} = \eta_{i_n^\star} + 1$
  $\mathcal{J}_{n+1} = \mathcal{J}_n \setminus \{j_n^\star\}$
  $\mathbf{b}_{i_n^\star}^{(n)} = \mathbf{b}_{i_n^\star}^{(n-1)} \oplus_{\text{end}} \{j_n^\star\}$
  $\mathbf{b}_i^{(n)} = \mathbf{b}_i^{(n-1)},\ \forall i \neq i_n^\star$
  if $\eta_{i_n^\star} = L_t$ **then**
    $\mathcal{I}_{n+1} = \mathcal{I}_n \setminus \{i_n^\star\}$
    $c_{i_n^\star, j}^{(n+1)} = 0, \forall j \in \mathcal{J}$
  else
    $\mathcal{I}_{n+1} = \mathcal{I}_n$
  end if
  $c_{i, j_n^\star}^{(n+1)} = 0,\ \forall i \in \mathcal{I}_{n+1}$
  $c_{ij}^{(n+1)} = c_{ij}[\mathbf{b}_i^{(n)}],\ \forall (i,j) \in \mathcal{I}_{n+1} \times \mathcal{J}_{n+1}$
end for

CBBA (and CBAA) Convergence Guarantees

- Network diameter D (assuming connected)
- Convergence time T_C

$$D \triangleq \max_{(i,k)\in\mathcal{I}^2} d_{ik}.$$

$$T_C \triangleq \min t \in \mathcal{T}$$

where the set $\mathcal{T}$ is defined as

$$\mathcal{T} = \left\{ t \in \mathbb{Z}_+ \middle| \forall s \geq t : x_{ij}(s) = x_{ij}(t), \sum_{i=1}^{N_u} x_{ij}(s) = 1 \right.$$
$$\left. \sum_{j=1}^{N_t} x_{ij}(s) \leq L_t, \sum_{j=1}^{N_t} \sum_{i=1}^{N_u} x_{ij}(s) = N_{\min} \right\}$$

Theorem 1 (Convergence of CBBA): Provided that the scoring function is DMG, the CBBA process with a synchronized conflict resolution phase over a static communication network with diameter D satisfies the following.

1) CBBA produces the same solution as SGA with the corresponding winning bid values and winning agent information being shared across the fleet, i.e.,

$$z_{i,j_k^\star} = i_k^\star \qquad \forall k \leq N_{\min} \quad \forall i \in \mathcal{I}$$
$$y_{i,j_k^\star} = c_{i_k^\star,j_k^\star}^{(k)} \qquad \forall k \leq N_{\min} \quad \forall i \in \mathcal{I}. \qquad (31)$$

2) The convergence time T_C is bounded above by $N_{\min} D$.

$$N_{\min} \triangleq \min\{N_t, N_u L_t\}$$

CBAA Minimum Performance Guarantee (Optimality)

- Single assignment optimality (extends to multiple assignment)

$$SOPT \leq 2CBAA.$$

- Guaranteed at least 50% of optimal task assignment score

CBAA Minimum Performance Guarantee (Optimality)

- Prove for Sequential Greedy Algorithm (has same performance as CBBA/CBAA)

- Relabel tasks/agents s.t. agent i's task is task i, agent j's task is task j, etc.
- Greedy algorithm implies $c_{ii} \geq c_{jj}, \quad \text{if } i < j$ and $$c_{ii} \geq c_{ij} \quad \forall i \quad \forall j > i$$ $$c_{ii} \geq c_{ji} \quad \forall i \quad \forall j > i.$$
- Total objective value is $CBAA = \sum_{i=1}^{N_{\min}} c_{ii}.$

CBAA Minimum Performance Guarantee (Optimality)

- Worst-case scenario - swapping assignments would produce largest possible improvement in overall score
- Agents i and j switch tasks - original score $c_{ii} + c_{jj}$ becomes $c_{ij} + c_{ji}$
- Upper bound - $c_{ij} + c_{ji} \leq c_{ii} + c_{ii} = 2c_{ii}$ (achieved if $c_{ij} = c_{ji} = c_{ii}$.)

- Suppose above holds for all pairs of agents
- Swap tasks for agents 1 and n, 2 and n-2, etc.

$$c_{ij} = c_{ii} \qquad \forall i \qquad \forall j > i$$

$$c_{ji} = c_{ii} \qquad \forall i \qquad \forall j > i$$

$$J_i^* = \begin{cases} N_{\min} - i + 1, & \text{if } i \in \{1, \ldots, N_{\min}\} \\ \emptyset, & \text{otherwise} \end{cases}$$

CBAA Minimum Performance Guarantee (Optimality)

$$
\begin{aligned}
SOPT &= \sum_{i=1}^{\lceil N_{\min}/2 \rceil} c_{ii} + \sum_{i=\lceil N_{\min}/2 \rceil+1}^{N_{\min}} c_{(N_{\min}-i+1),(N_{\min}-i+1)} \\
&= 2 \times \sum_{i=1}^{\lfloor N_{\min}/2 \rfloor} c_{ii} + \sum_{i=\lfloor N_{\min}/2 \rfloor+1}^{\lceil N_{\min}/2 \rceil} c_{ii} \le 2 \times \sum_{i=1}^{N_{\min}} c_{ii} \\
&= 2CBAA.
\end{aligned}
$$

Greedy vs. Optimal

c1	1.5	0.1	0.1	0.1
c2	3.5	3.4	0.1	0.1
c3	3.4	0.1	0.1	0.1
c4	1.5	0.1	0.1	0.1

Greedy:

(1, 4) - 0.1

(2, 1) - 3.5

(3, 2) - 0.1

(4, 3) - 0.1

Total: 3.8

Optimal:

(1, 4) - 0.1

(2, 2) - 3.4

(3, 1) - 3.4

(4, 3) - 0.1

Total: 7.0

Greedy vs. Optimal - Worst Case

c1	3.5	0	0	3.5
c2	0	2.5	2.5	0
c3	0	2.5	0	0
c4	3.5	0	0	0

Greedy:

(1, 1) - 3.5
(2, 2) - 2.5
(3, 3) - 0
(4, 4) - 0

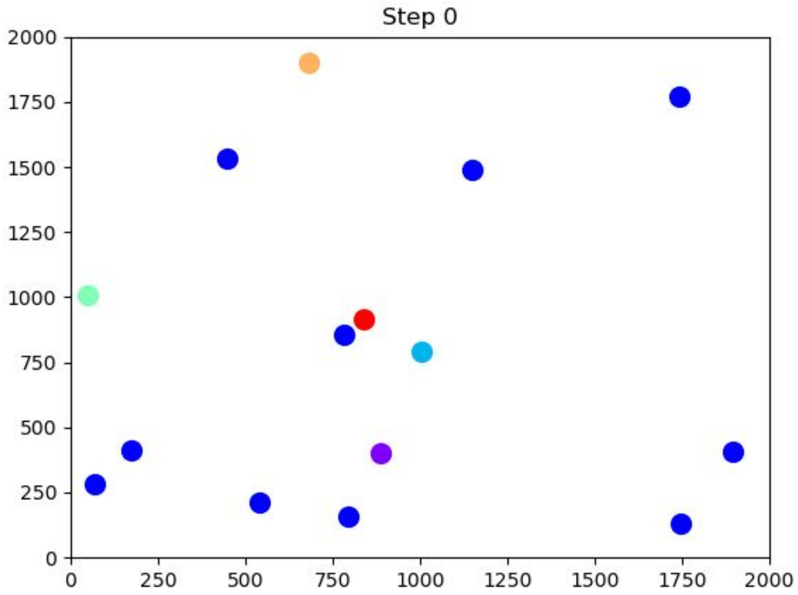
Total: 6.0

Optimal:

(1, 4) - 3.5
(2, 3) - 2.5
(3, 2) - 2.5
(4, 1) - 3.5

Total: 12.0

Simulation Results



N_t = 10 tasks

N_u = 5 agents

L_t = 5 tasks/agent

Simulation Results