Distributed tasking algorithms

Siarhei Dymkou

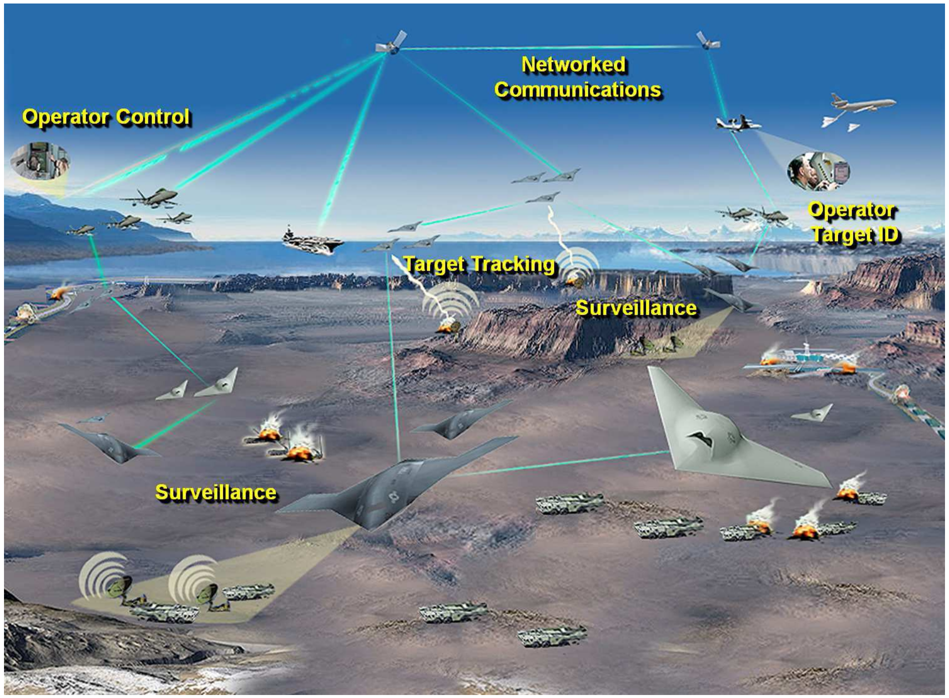
Temasek Laboratories

National University of Singapore

T-Lab Building 5A, Engineering Drive 1,05-02 Singapore 117411

Motivation

Modern missions involve multi-agent teams cooperating to perform tasks:



- search and track;
- classify targets, monitor status;
- rescue operations.

Key questions:

- How to coordinate team behavior to improve mission performance?
- How to hedge against uncertainty in dynamic environments?
- How to handle varying communication constraints?

Problem Statement

Objective: Automate task allocation to improve mission performance

Problem Statement:

- Maximize mission score
- Satisfy constraints
- **Decision variables:**
  - Team assignments, Service times

- Spatial and temporal coordination of team;
- Computational efficiency for real-time implementation;

Key Technical Challenges::

- Complex agent modeling (stochastic, nonlinear, time-varying)
- Constraints due to limited resources (fuel, payload, bandwidth, etc)
- Dynamic networks and communication requirements

Tasking approaches

Most involve centralized planning

- GCS plans and distributes tasks to all agents;
- Requires full situational awareness;
- High bandwidth, slow reaction to local changes

$\Rightarrow$

Motivates distributed planning

- Agents make plans individually and coordinate with each other;
- Faster reaction to local information;
- Increased agent autonomy

Key questions for distributed planning:

- What quantities should the agents agree upon? (Information / tasks and plans / objectives / constraints)
- How to ensure that planning is robust to inaccurate information and models?

Distributed Planning

Centralized Problem:

- Maximize mission score
- Satisfy constraints
- **Decision variables:**
  - Team assignments, Service times

$\Rightarrow$

Distributed Problem:

- Maximize mission score individually
- Satisfy constraints
- **Decision variables:**
  - Agent assignments, Service times

Main issues: Coupling and Communication:

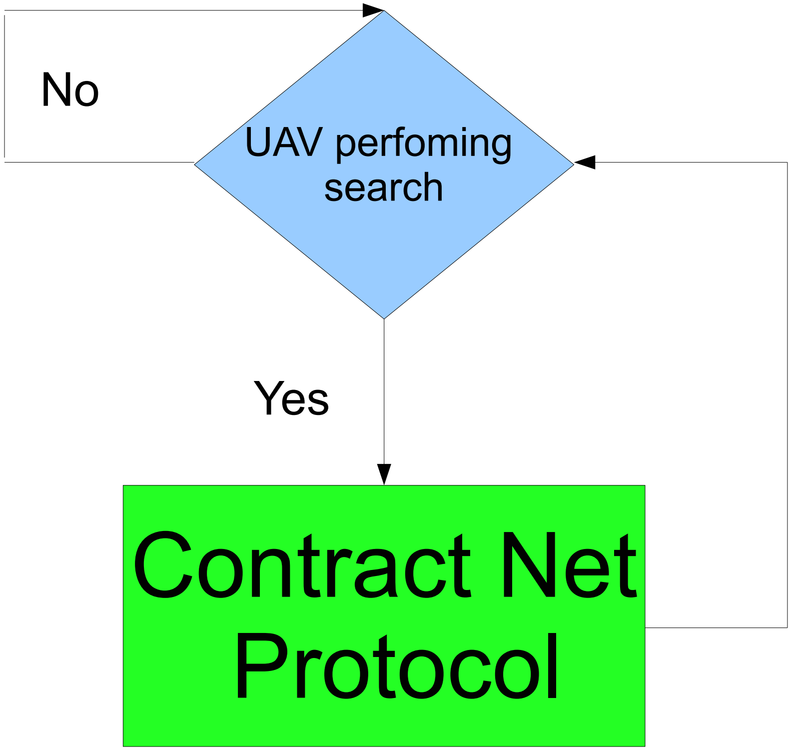
- Agent score functions depend on other agents decisions
- Joint constraints between multiple agents
- Agent optimization is based on local information

Key challenge: How to design appropriate protocols?

- Specify what information to communicate
- Create rules to process received information and modify plans
- Performance guarantees
- Will algorithm converge to a feasible assignment?

Contract Net Protocol (CNP)

CNP is starting when the UAV found a target or received corresponding message

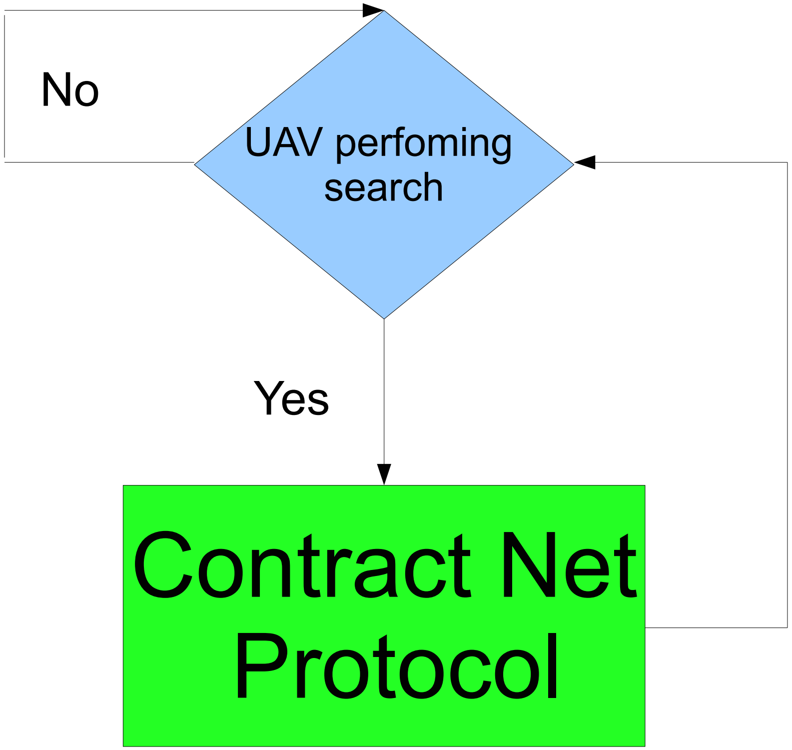


Application: An autonomous team of UAVs performing search and track missions

Team of UAVs autonomously searching an area for vehicles that could be stationary or moving. Once found, the UAVs will track the vehicles. This is performed autonomously. Algorithms are required to run onboard the UAVs to make them work collaboratively to complete the mission.

Contract Net Protocol (CNP)

CNP is starting when the UAV found a target or received corresponding message

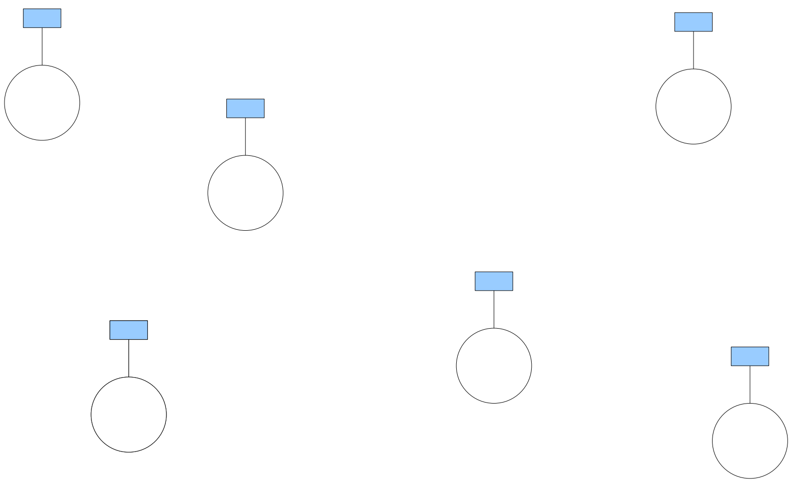


Application: Scenario parameters are as follows:

- 6 Mini-class UAVs(Speeds; Climb rates; Max bank angles; Turn radius; GPS navigation accuracy; Endurance; Communications range; Sensor footprint)
- 10 Targets and the area to search (2 km x 2 km)

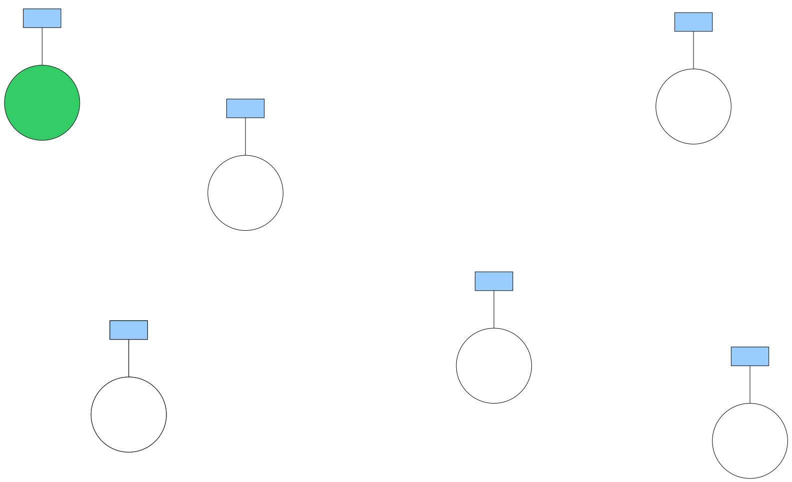
Team of UAVs autonomously searching an area for vehicles that could be stationary or moving. Once found, the UAVs will track the vehicles. This is performed autonomously. Algorithms are required to run onboard the UAVs to make them work collaboratively to complete the mission.

Contract Net Stages



- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

Contract Net Stages



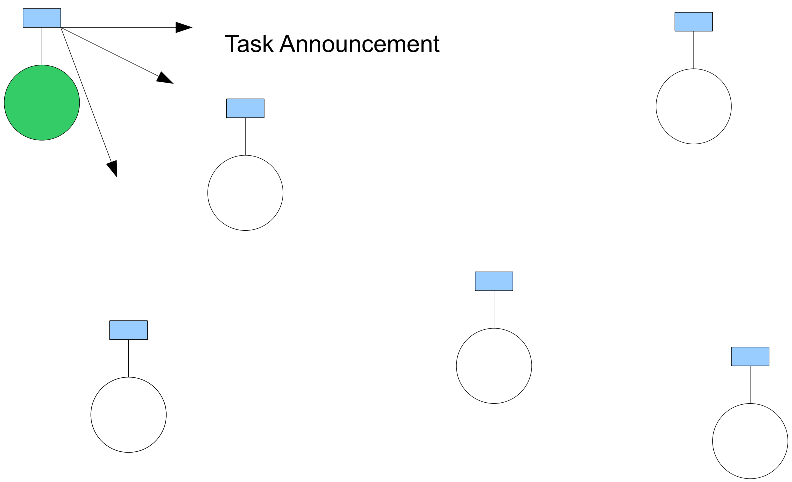
- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

In this stage, an agent recognises it has a problem it wants help with.
Agent has a goal, and either

- realises it cannot achieve the goal in isolation - does not have capability;
- realises it would prefer not to achieve the goal in isolation (typically because of solution quality, deadline, etc)

As a result, it needs to involve other agents.

Contract Net Stages



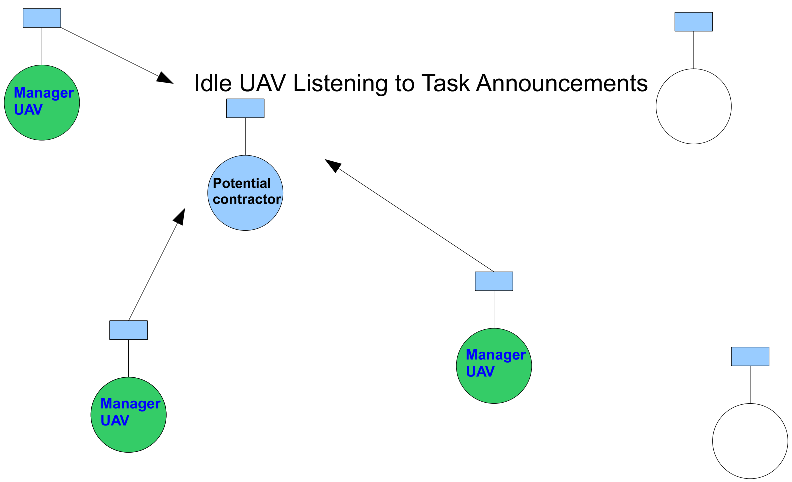
- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

In this stage, the agent with the task sends out an announcement of the task which includes a specification of the task to be achieved.
Specification must encode:

- description of task itself (maybe executable);
- any constraints (e.g., deadlines, quality constraints).
- meta-task information (e.g.,bids must be submitted by...)

The announcement is then broadcast.

Contract Net Stages



- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

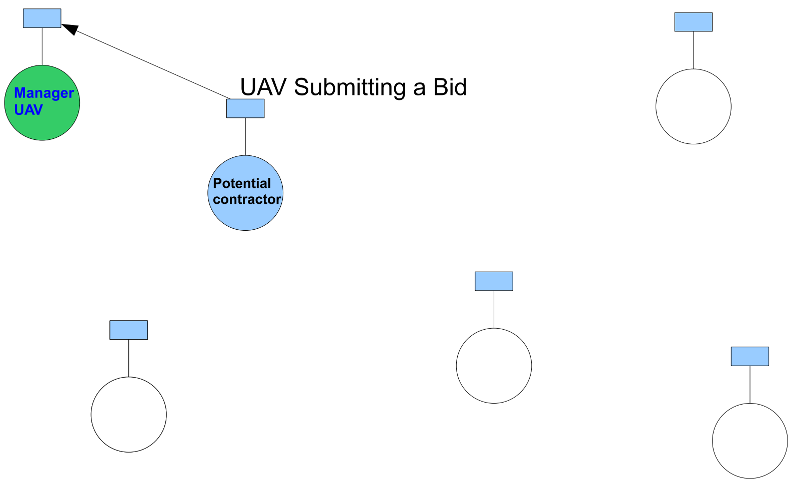
In this stage, the agent with the task sends out an announcement of the task which includes a specification of the task to be achieved.

Specification must encode:

- description of task itself (maybe executable);
- any constraints (e.g., deadlines, quality constraints).
- meta-task information (e.g.,bids must be submitted by...)

The announcement is then broadcast.

Contract Net Stages



- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

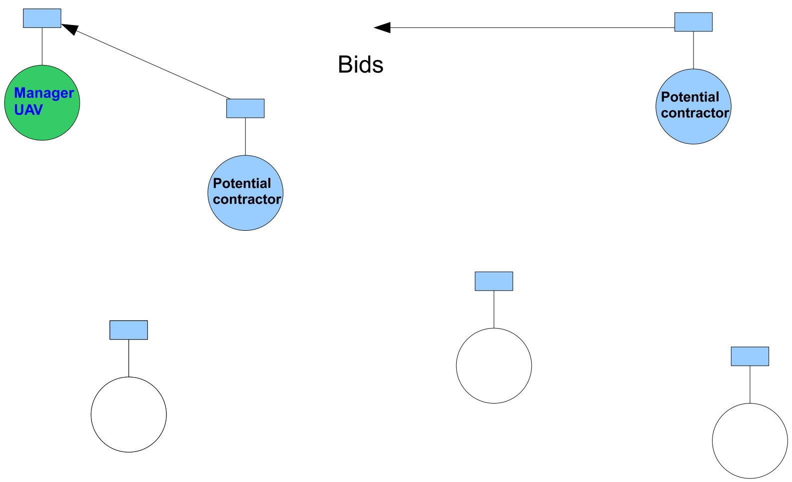
UAVs that receive the announcement decide for themselves whether they wish to bid for the task.

Factors:

- agent must decide whether it is capable of expediting task;
- agent must determine quality constraints and price information (if relevant).

If they do choose to bid, then they submit a tender.

Contract Net Stages



- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

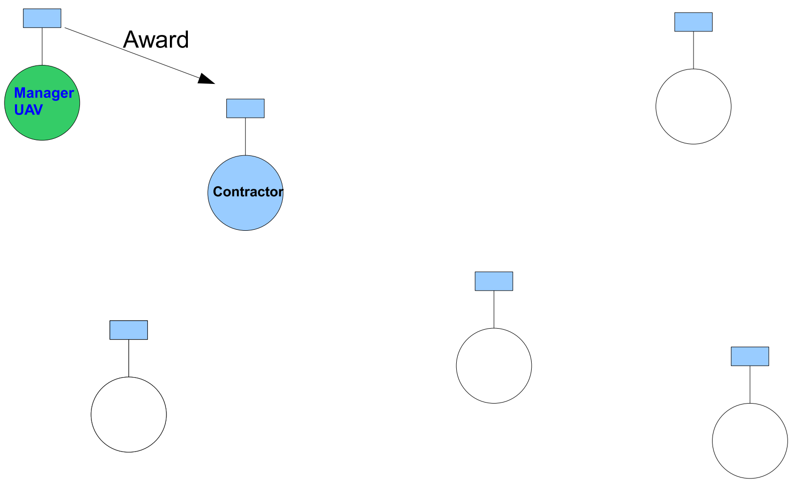
UAVs that receive the announcement decide for themselves whether they wish to bid for the task.

Factors:

- agent must decide whether it is capable of expediting task;
- agent must determine quality constraints and price information (if relevant).

If they do choose to bid, then they submit a tender.

Contract Net Stages

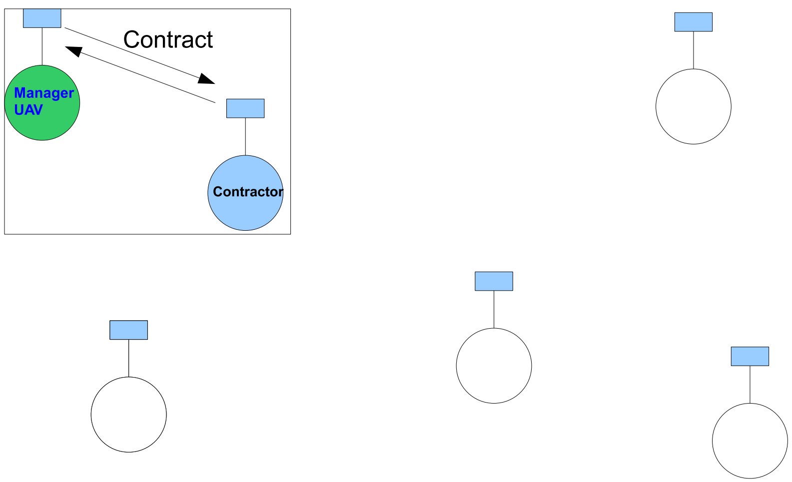


- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

Agent that sent task announcement must choose between bids and decide who to "award the contract" to.

The result of this process is communicated to agents that submitted a bid.

Contract Net Stages

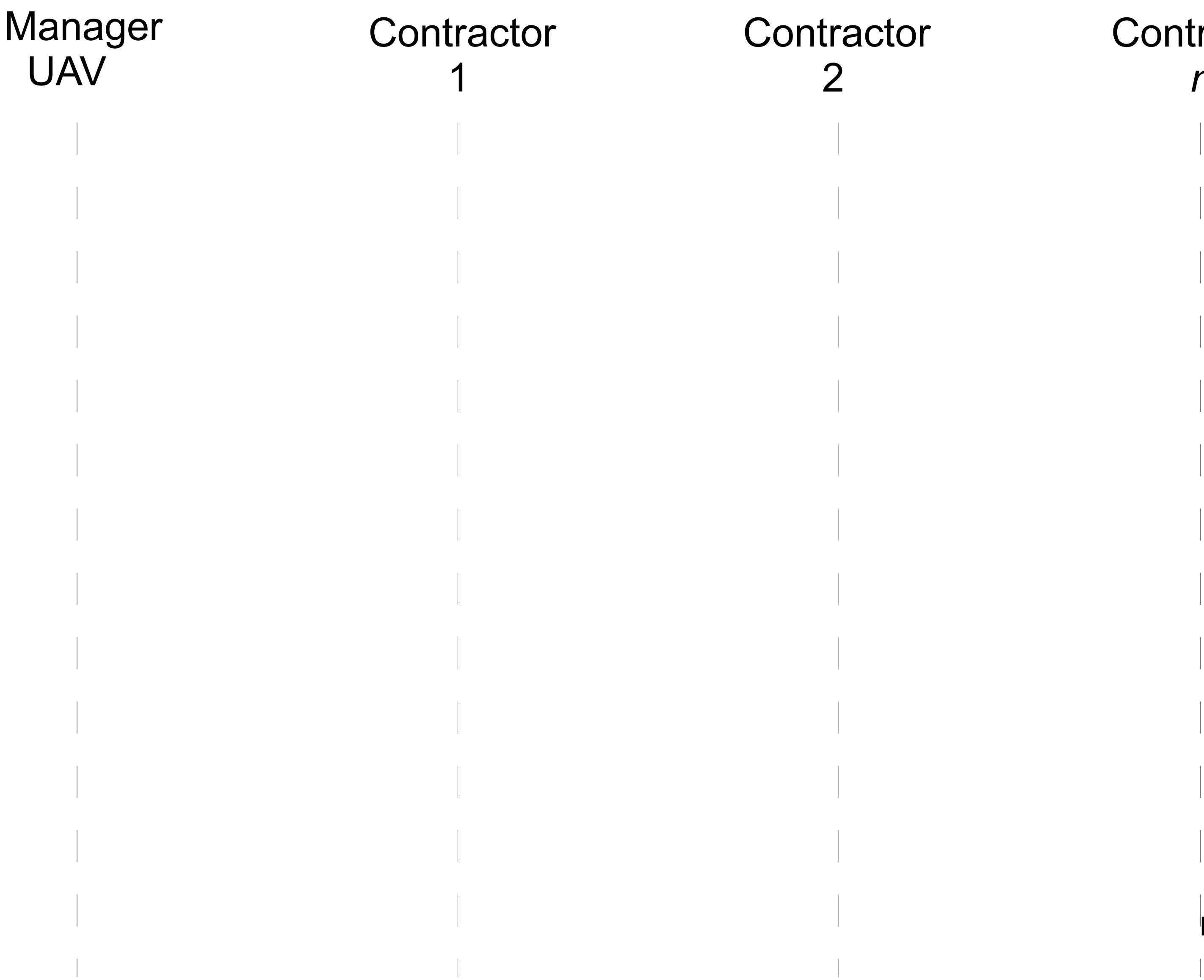


- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

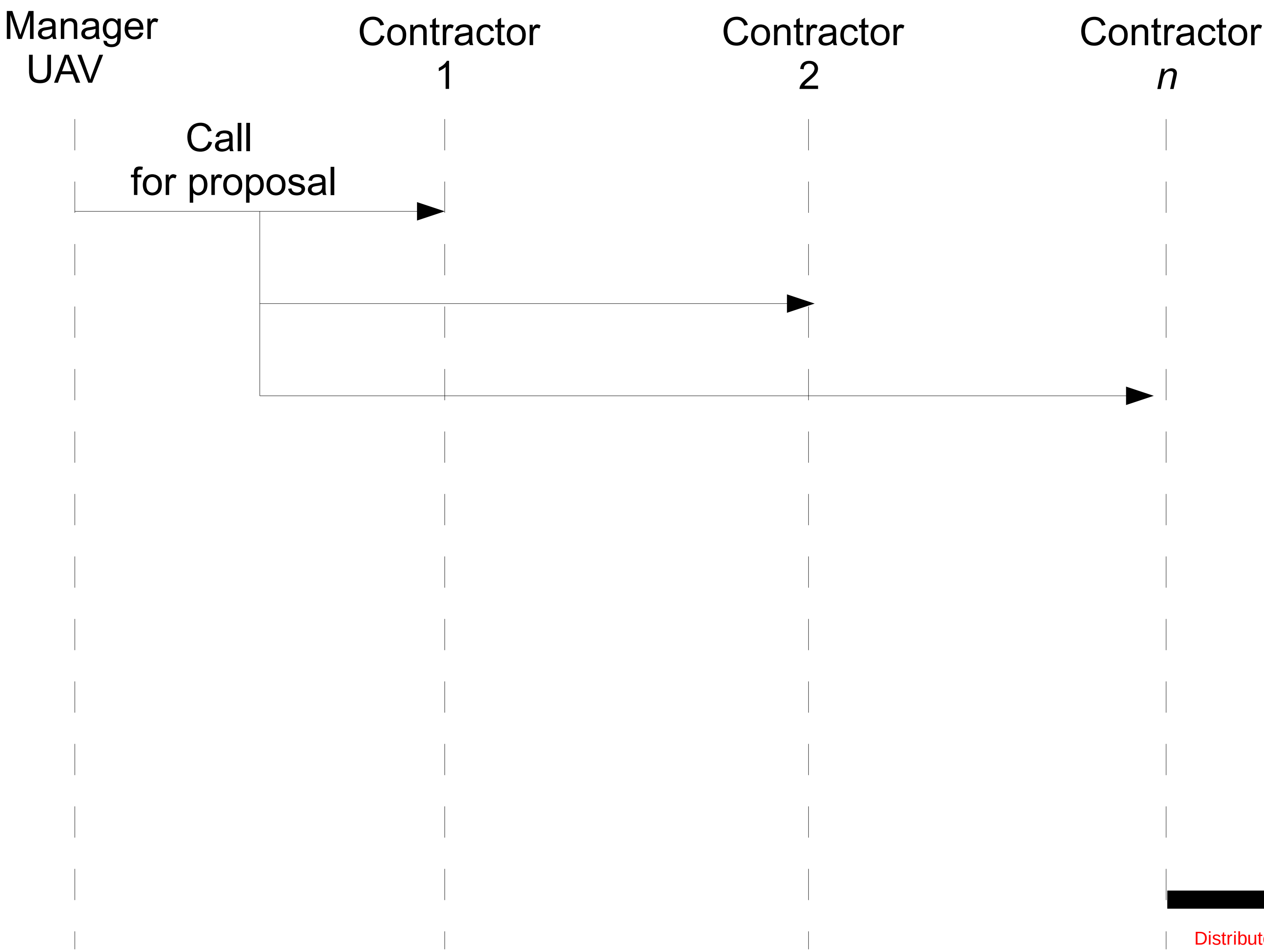
Agent that sent task announcement must choose between bids and decide who to "award the contract" to.
The result of this process is communicated to agents that submitted a bid.
The successful contractor then expedites the task.
May involve generating further manager-contractor relationships: sub-contracting.

- May involve another contract net.

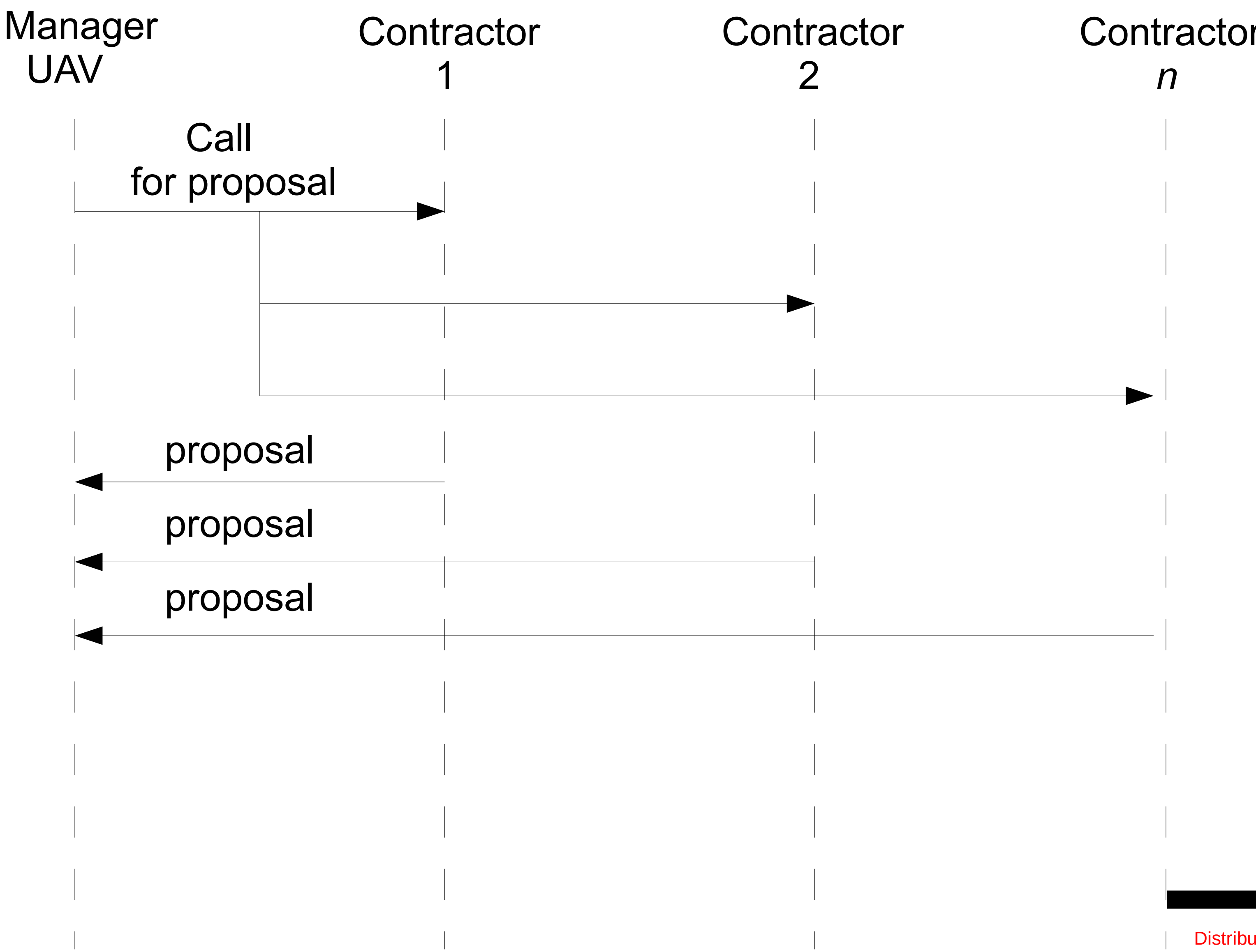
Procedure Diagram



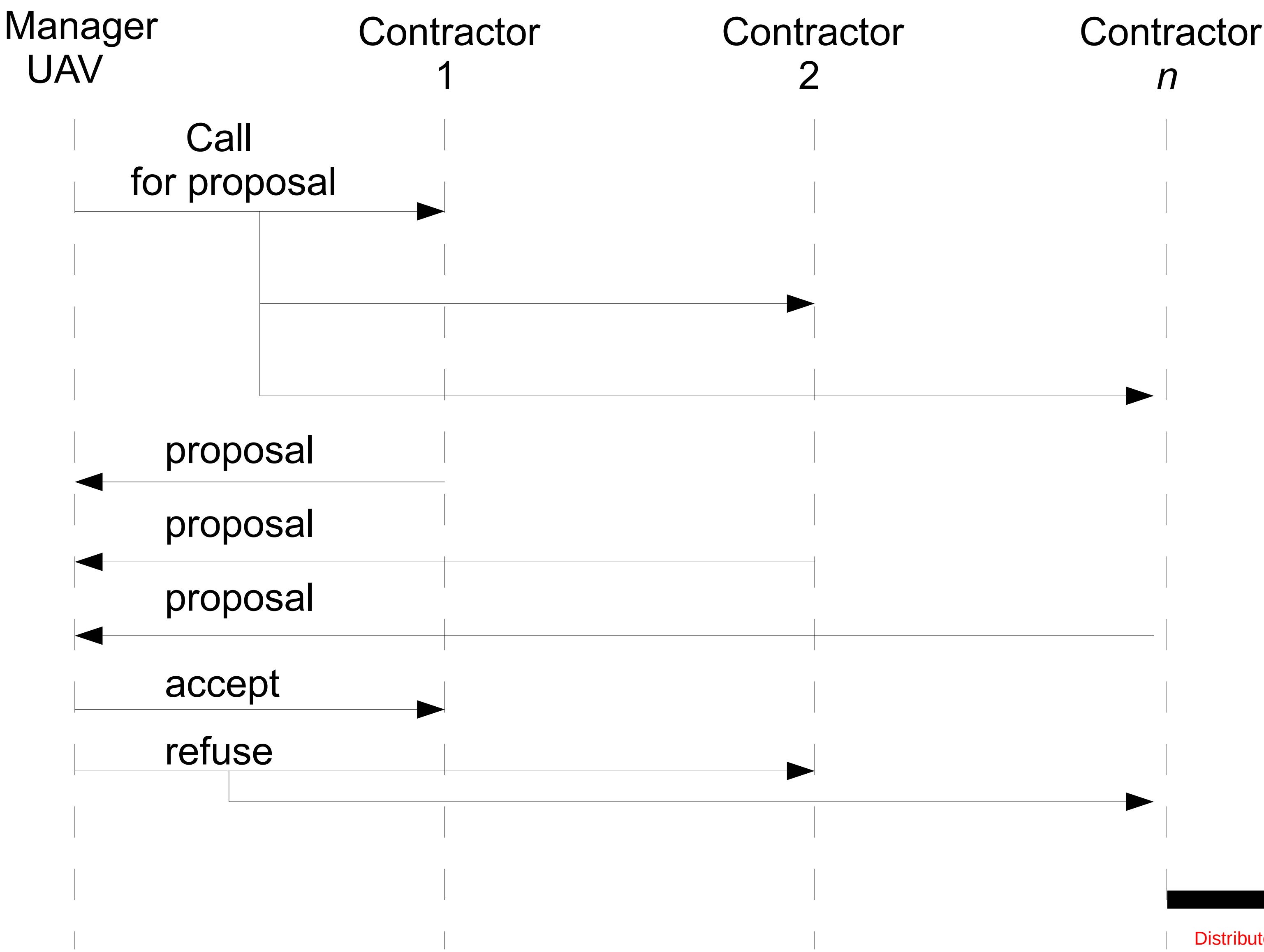
Procedure Diagram



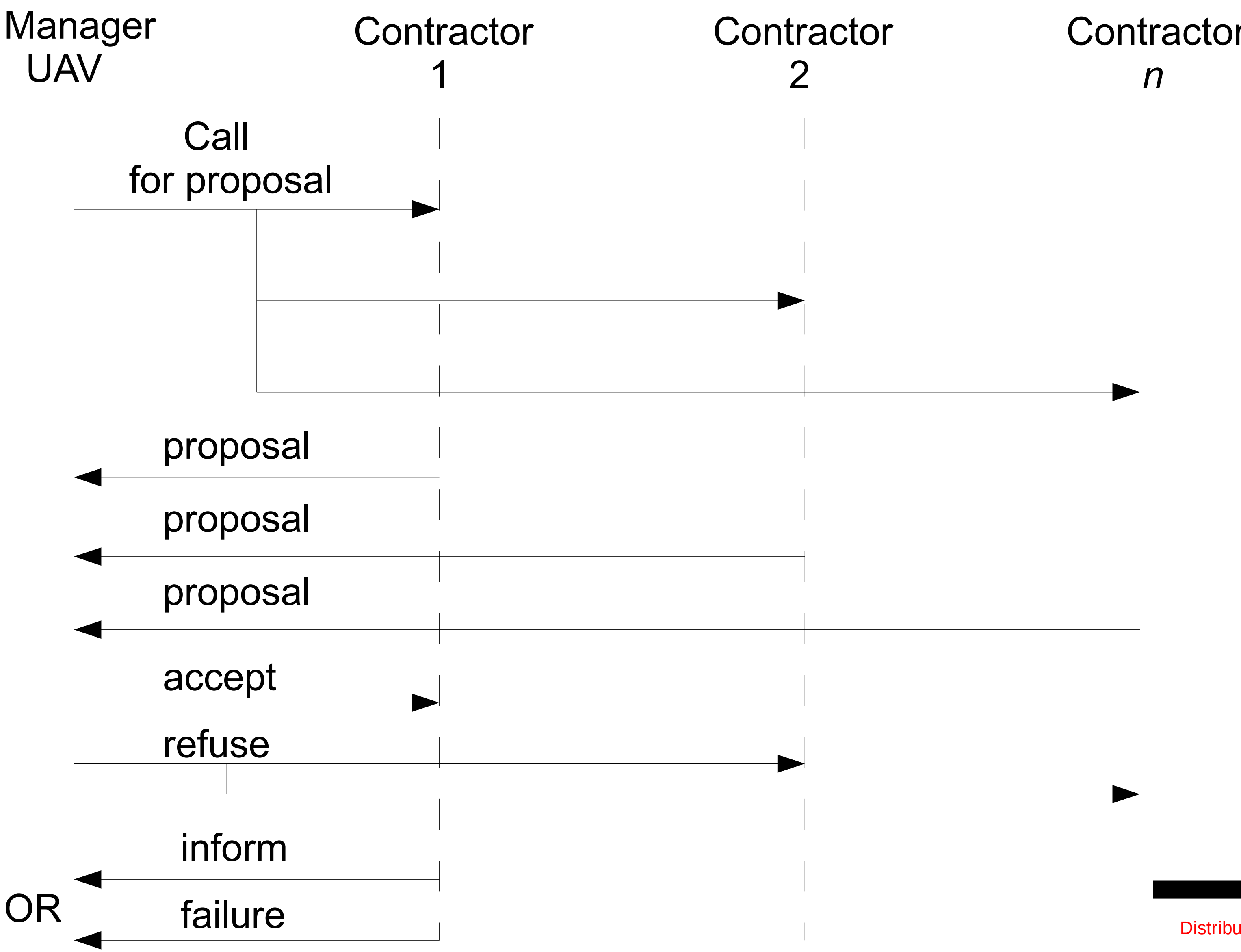
Procedure Diagram



Procedure Diagram



Procedure Diagram

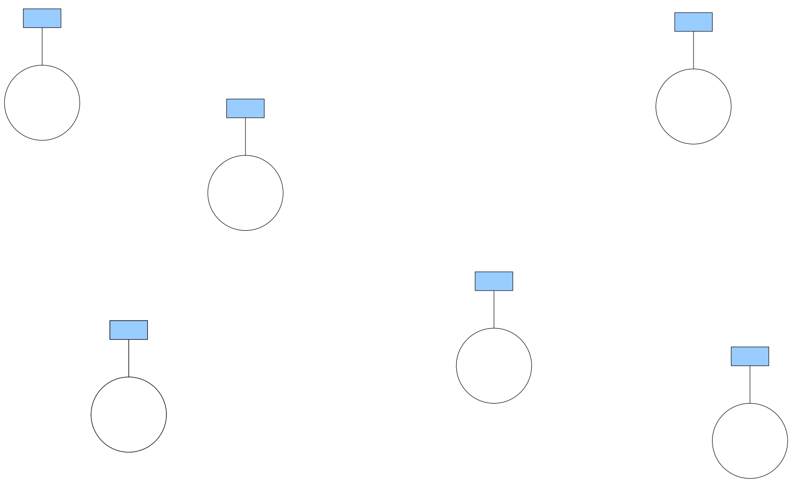


Issues for Implementing Contract Net

How to. . .

- ... specify tasks?
- ... specify quality of service?
- ... decide how to bid?
- ... select between competing offers?
- ... differentiate between offers based on multiple criteria?

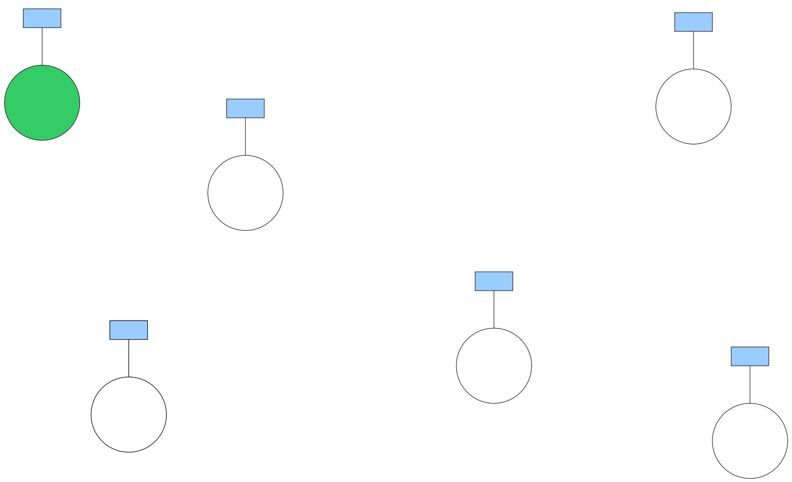
Agent Information



- A bundle of targets, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$					
$Path$					
$Time$					

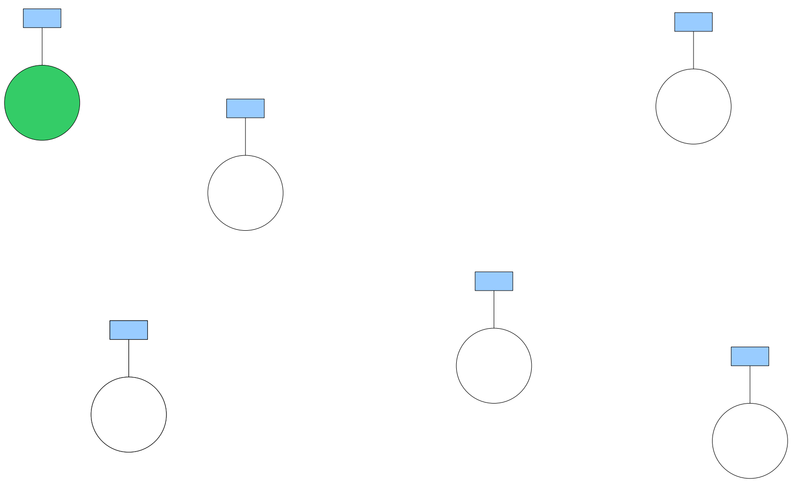
Agent Information



- A bundle of targets, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	✓				$b_i = [b_{i1}]$
$Path$					
$Time$					

Agent Information

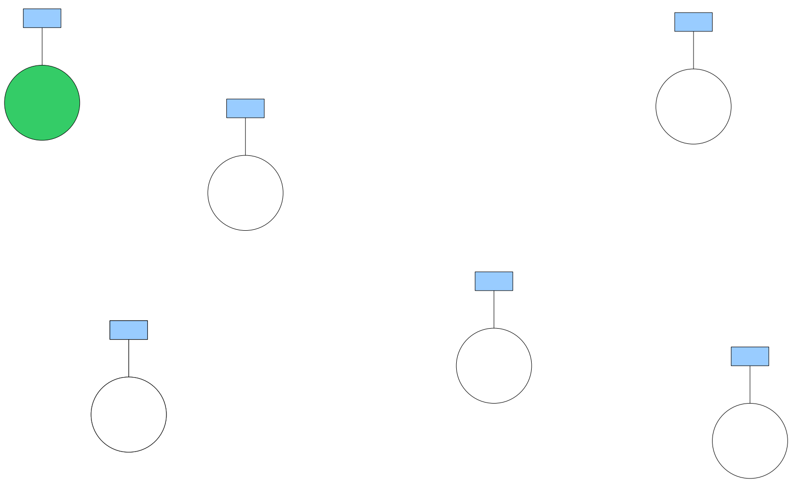


Recognition phase

Type of target (static or moving);

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}]$
$Path$					
$Time$					

Agent Information



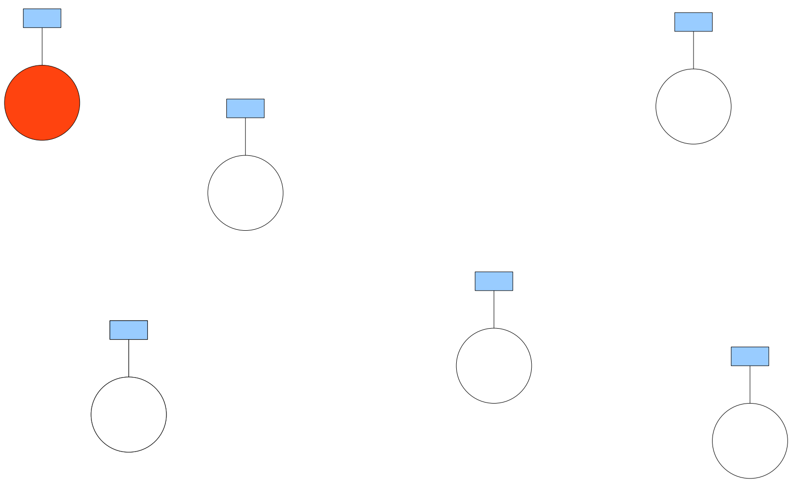
Recognition phase

Type of target (static or moving);

Switch to track and update own vectors of information, and disseminate to other UAVs.

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}]$
$Path$					
$Time$					

Agent Information



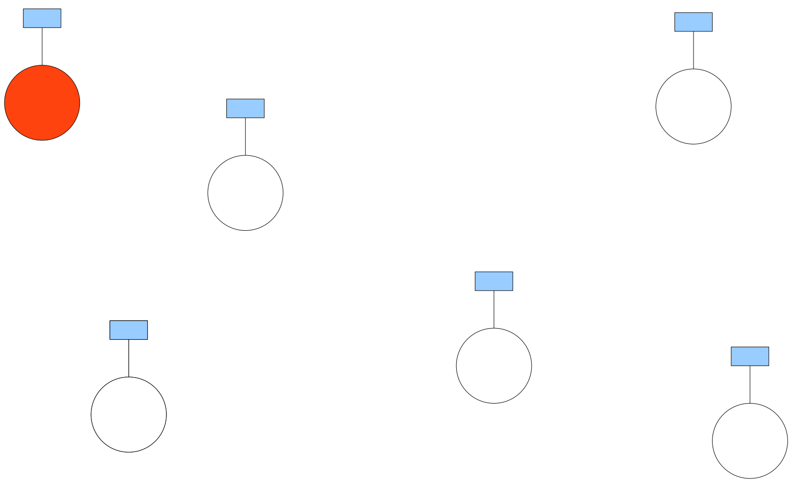
Recognition phase

Type of target (static or moving);

Switch to track and update own vectors of information, and disseminate to other UAVs.

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}, b_{i2}]$
$Path$	p_{i1}				$p_i = [p_{i1}]$
$Time$					

Agent Information



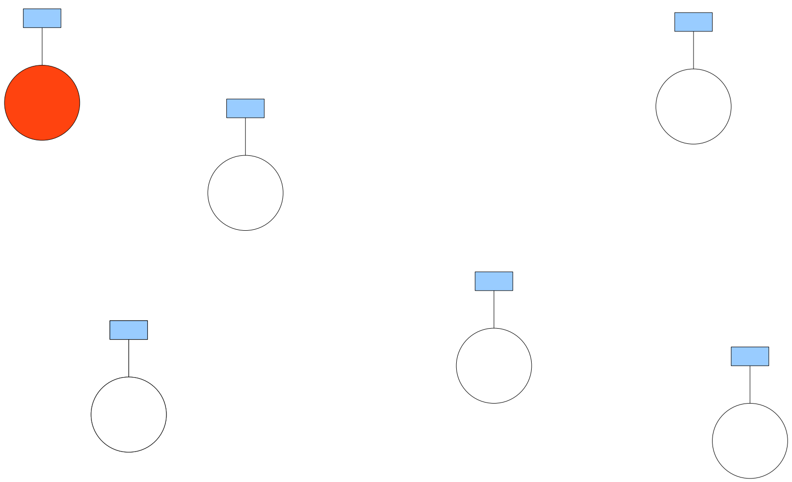
Recognition phase

Type of target (static or moving);

Switch to track and update own vectors of information, and disseminate to other UAVs.

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}]$
$Path$	p_{i1}				$p_i = [p_{i1}]$
$Time$	τ_{i1}				$\tau_i = [\tau_{i1}]$

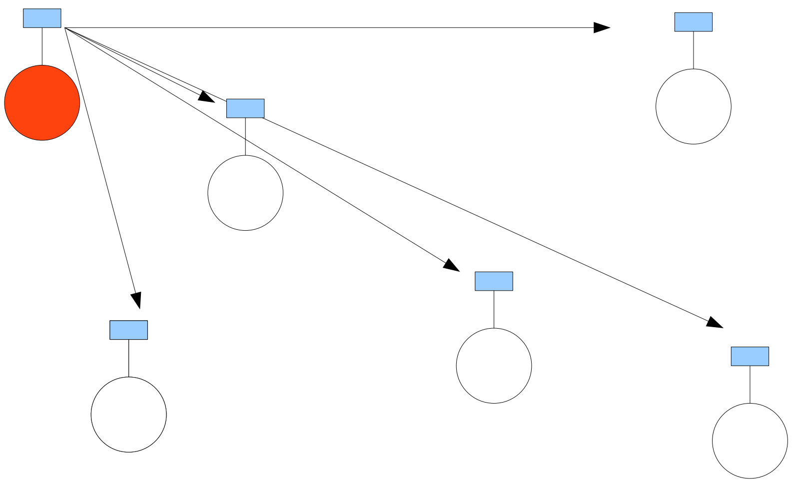
Agent Information



Recognition phase
Type of target (static or moving);
Switch to track and update own vectors of information, and disseminate to other UAVs.

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Winning\ \ Agent$	i				$z_i = [z_{i1}]$
$WinningBids$	y_{i1}				$y_i = [y_{i1}]$

Agent Information



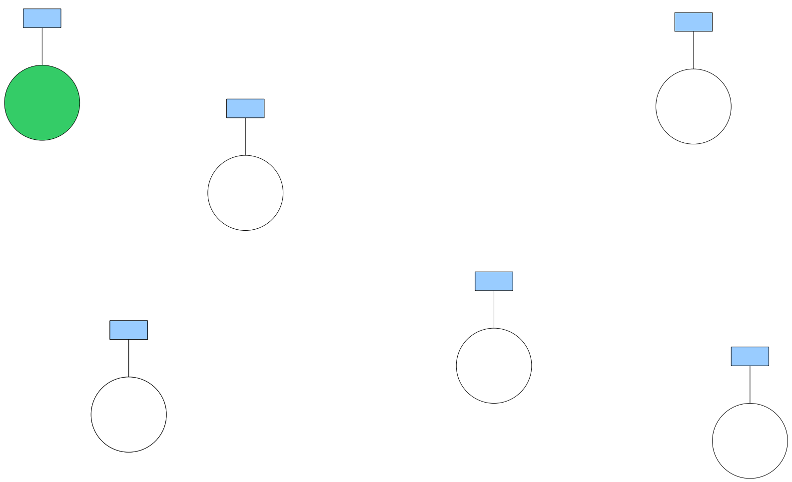
Recognition phase

Type of target (static or moving);

Switch to track and update own vectors of information, and disseminate to other UAVs.

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}]$
$Winning\ \ Agent$	i				$z_i = [z_{i1}]$
$WinningBids$	y_{i1}				$y_i = [y_{i1}]$

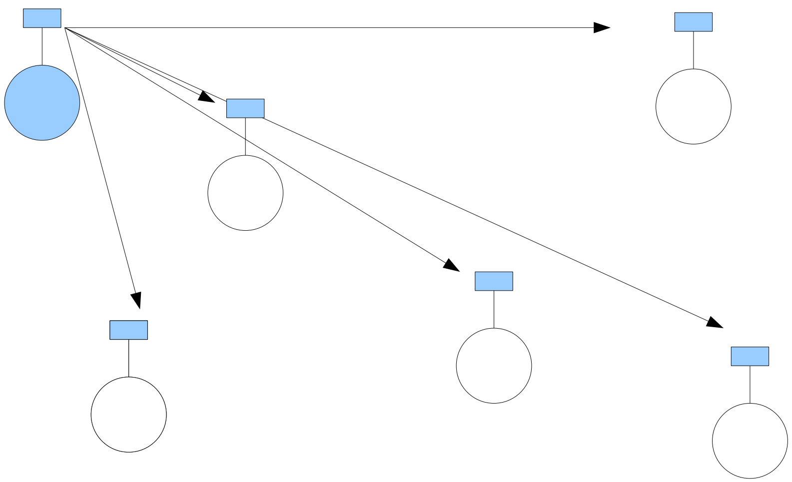
Agent Information



Recognition phase
Type of target (static or moving);
Add target to current pass and calculate the bid for new pass, then send message announcement to potential contractor

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}]$
$Path$					
$Time$					

Agent Information



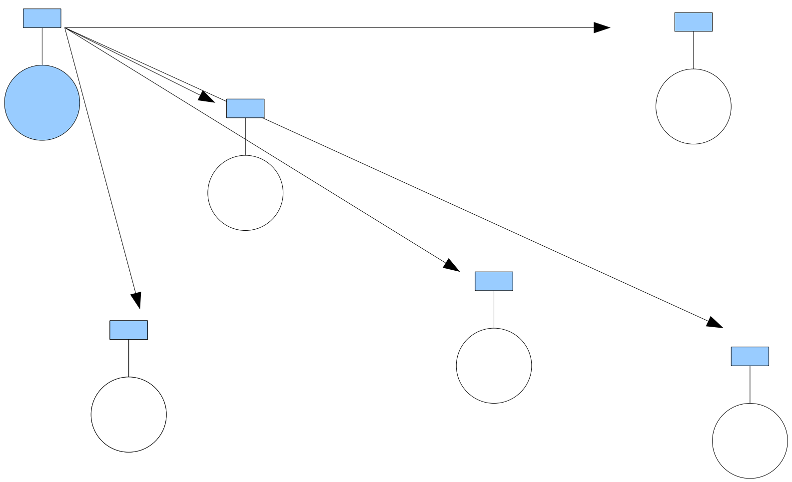
Recognition phase

Type of target (static or moving);

Add target to current pass and calculate the bid for new pass, then send message announcement to potential contractor .

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}]$
$Winning\ \ Agent$	i				$z_i = [z_{i1}]$
$WinningBids$	y_{i1}				$y_i = [y_{i1}]$

Agent Information



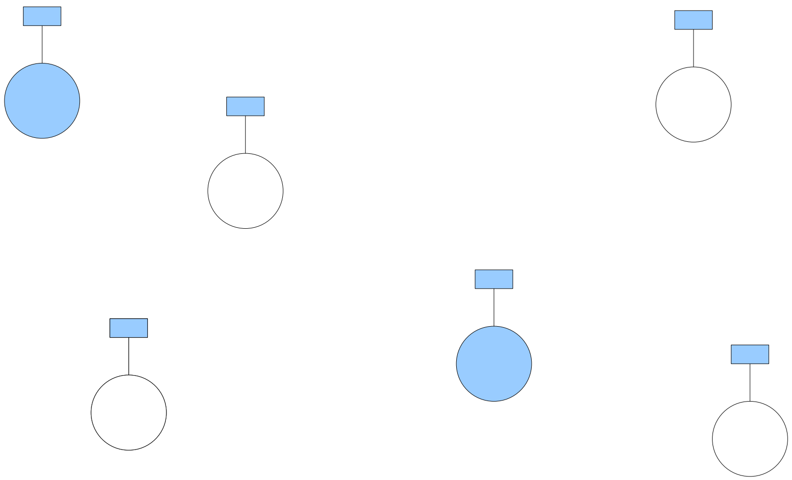
Recognition phase

Type of target (static or moving);

Do this until number of static target in the bundle ≤ 2.

i	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$				$b_i = [b_{i1}]$
$Winning\ \ Agent$	i				$z_i = [z_{i1}]$
$WinningBids$	y_{i1}				$y_i = [y_{i1}]$

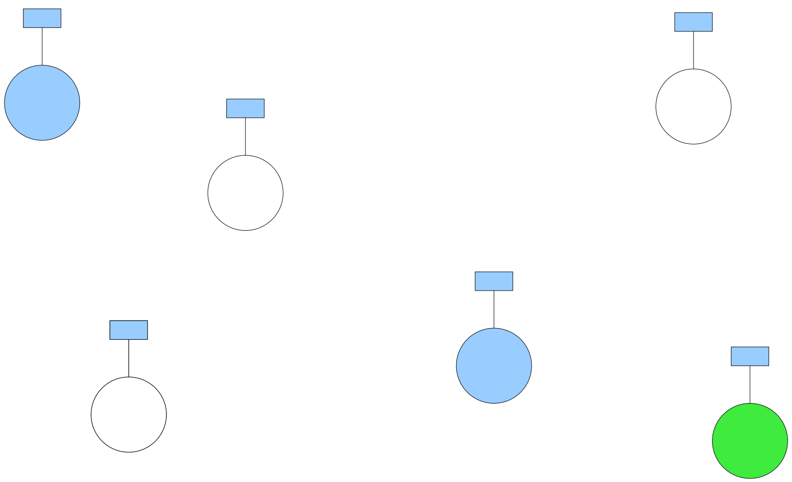
Agent Information



For example two static targets a found then each UAVs have the following information

k	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$	$\checkmark$			$b_k = [b_{k1}, b_{k2}]$
$Winning\ \ Agent$	i	j			$z_k = [z_{k1}, z_{k2}]$
$WinningBids$	y_{k1}	y_{k2}			$y_k = [y_{k1}, y_{k2}]$

Agent Information

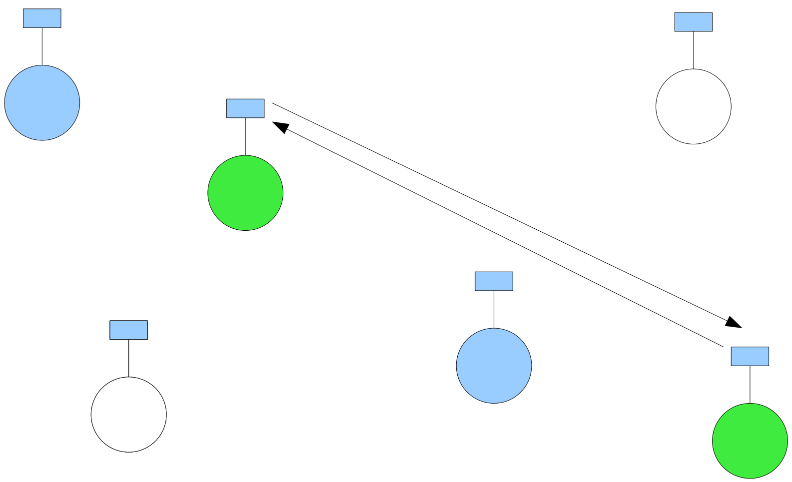


Recognition phase

Type of target (static or moving);

k	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	$\checkmark$	$\checkmark$	$\maltese$		$b_k = [b_{k1}, b_{k2}, b_{kk}]$
$Winning\ \ Agent$	i	j			$z_k = [z_{k1}, z_{k2}]$
$WinningBids$	y_{k1}	y_{k2}			$y_k = [y_{k1}, y_{k2}]$

Agent Information



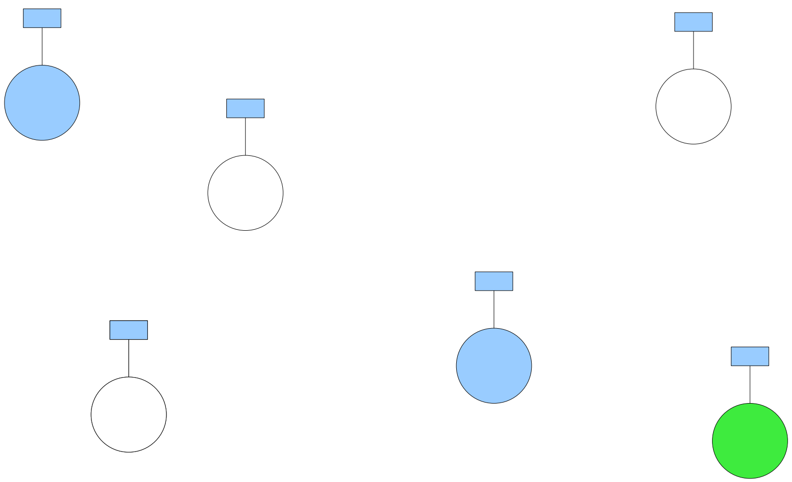
Recognition phase

Type of target (static or moving);

Check of existence of another new static target, if exist more then 1, select a manager UAV.

k	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	✓	✓	✠	✠	$[b_{k1}, b_{k2}, b_{kk}, b_{kN_t}]$
$Winning\ \ Agent$	i	j			$z_k = [z_{k1}, z_{k2}]$
$WinningBids$	y_{k1}	y_{k2}			$y_k = [y_{k1}, y_{k2}]$

Agent Information



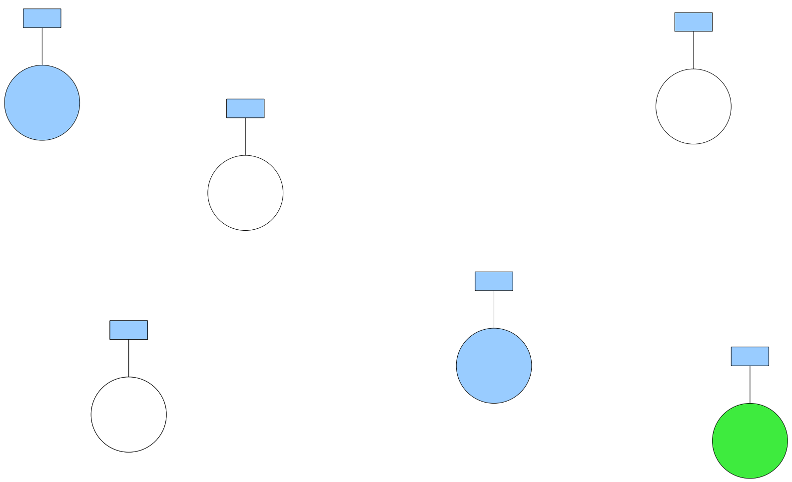
Recognition phase

Then do "Separation procedure": where input are: locations of static targets and

Number of subgroups =Total static target - number of new static targets)

k	$Target_1$	$Target_2$	$Target_k$	$Target_{N_t}$	$Values$
$Bandle$	✓	✓	✠	✠	$[b_{k1}, b_{k2}, b_{kk}, b_{kN_t}]$
$Winning\ \ Agent$	i	j			$z_k = [z_{k1}, z_{k2}]$
$WinningBids$	y_{k1}	y_{k2}			$y_k = [y_{k1}, y_{k2}]$

Agent Information



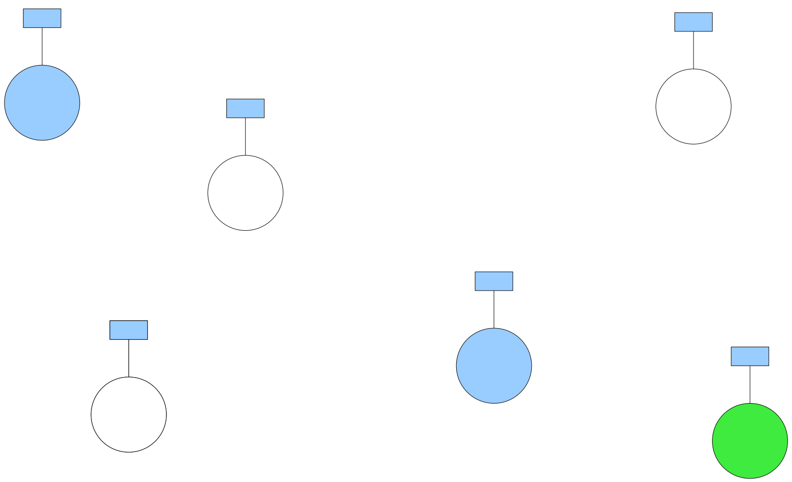
Recognition phase

Then do "Separation procedure": where input are: locations of static targets and

Number of subgroups = current length of bundle |b| - number of new static targets)

k	$Task_1$	$Task_2$			$Values$
$Bandle$	✓	✓	✠	✠	$[b_{k1}, b_{k2}, b_{kk}, b_{kN_t}]$
$Winning\ Agent$					$z_k = [z_{kt1}, z_{kt2}]$
$WinningBids$	y_{kt1}	y_{kt2}			$y_k = [y_{kt1}, y_{kt2}]$

Agent Information



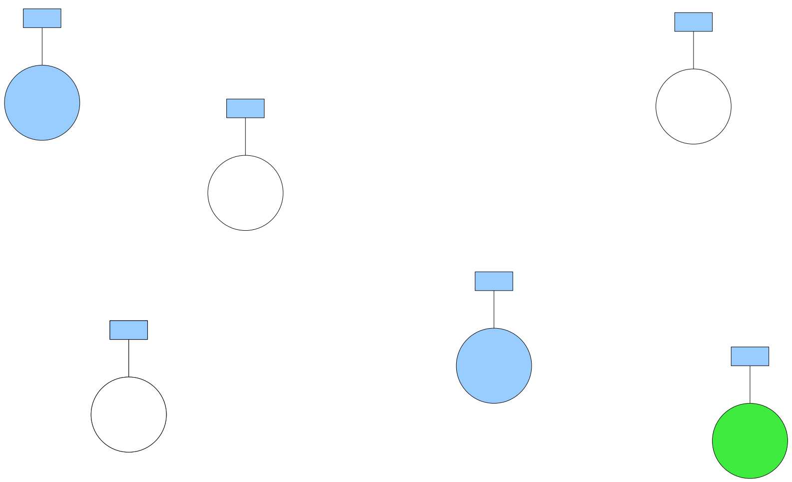
Recognition phase

Then do "Separation procedure": where input are: locations of static targets and

Number of subgroups = current length of bundle |b| - number of new static targets)

k	$Task_1$	$Task_2$			$Values$
$Bandle$	$\checkmark$	$\checkmark$			$[b_{kt1}, b_{kt2}]$
$Winning\ \ Agent$					$z_k = [z_{kt1}, z_{kt2}]$
$WinningBids$	y_{kt1}	y_{kt2}			$y_k = [y_{kt1}, y_{kt2}]$

Agent Information



- Recognition;
- Announcement ;
- Bidding;
- Awarding;
- Expediting.

k	$Task_1$	$Task_2$			$Values$
$Bandle$	$\checkmark$	$\checkmark$			$[b_{kt1}, b_{kt2}]$
$Winning\ \ Agent$					$z_k = [z_{kt1}, z_{kt2}]$
$WinningBids$	y_{kt1}	y_{kt2}			$y_k = [y_{kt1}, y_{kt2}]$

Manager UAV

Manager UAV_i					$Values$
$Bundle$					$b_i = []$
$Path$					$p_i = []$
$Time$					$\tau_i = []$

$Performing\ \ Search$

Manager UAV_i					$Values$
$Winning\ \ Agent$					$z_i = []$
$WinningBids$					$y_i = []$

Manager UAV

Manager UAV_i	$Target_1$				$Values$
$Bundle$	✠				$b_i = []$
$Path$					$p_i = []$
$Time$					$\tau_i = []$

$Found\ \ target$

Manager UAV_i					$Values$
$Winning\ \ Agent$					$z_i = []$
$WinningBids$					$y_i = []$

Manager UAV

Manager UAV_i	$Target_1$				$Values$
$Bundle$					$b_i = []$
$Path$					$p_i = []$
$Time$					$\tau_i = []$

$moving$

Manager UAV_i					$Values$
$Winning\ \ Agent$					$z_i = []$
$WinningBids$					$y_i = []$

Manager UAV

Manager UAV_i	$Target_1$				$Values$
$Bundle$	$\checkmark$				$b_i = []$
$Path$					$p_i = []$
$Time$					$\tau_i = []$

$$\Downarrow$$

$$Switch\ to\ track$$

Manager UAV

Manager UAV_i	$Target_1$				$Values$
$Bundle$					$b_i = []$
$Path$					$p_i = []$
$Time$					$\tau_i = []$

$static$

Manager UAV_i					$Values$
$Winning\ \ Agent$					$z_i = []$
$WinningBids$					$y_i = []$

Manager UAV

Manager UAV_i	$Target_1$				$Values$
$Bundle$	✠				$b_i = [b_{i1}]$
$Path$	p_{i1}				$p_i = [p_{i1}]$
$Time$	τ_{i1}				$\tau_i = [\tau_{i1}]$

$Calculate\ arrival\ time\ \tau_{i1}(p)\ and\ corresponding\ bid\ \ y_{i1}$

Manager UAV_i					$Values$
$Winning\ \ Agent$					$z_i = []$
$WinningBids$					$y_i = []$

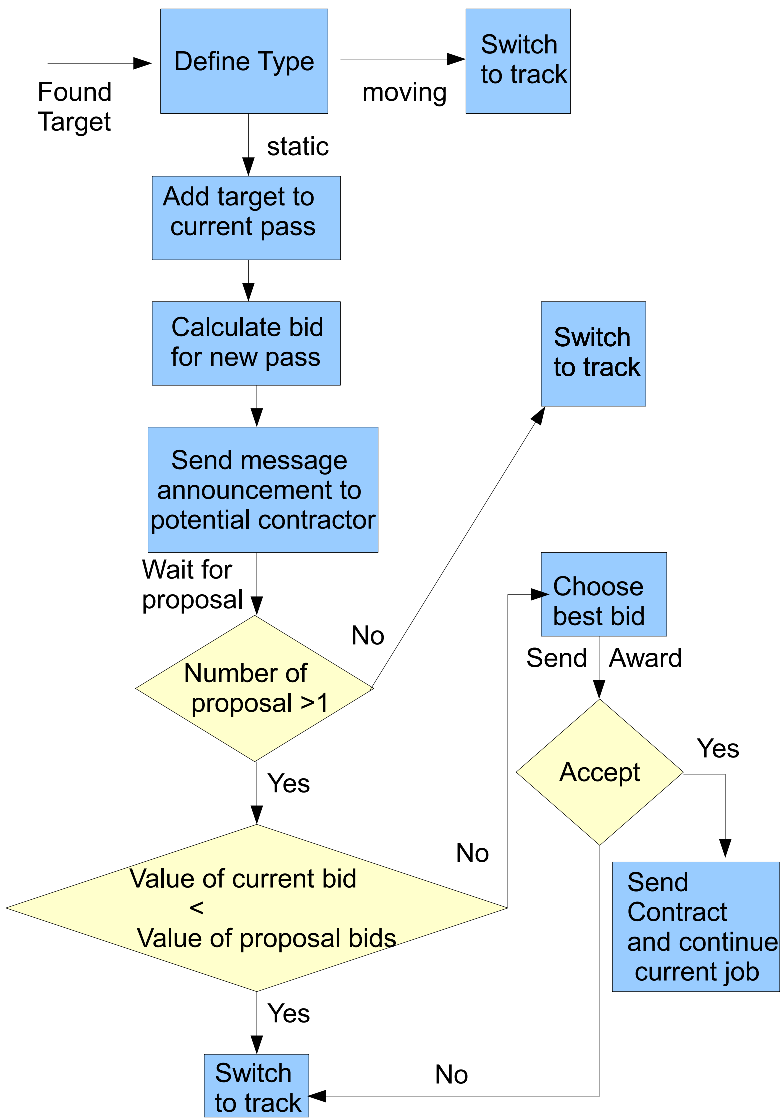
Manager UAV

Manager UAV_i	$Target_1$				$Values$
$Bundle$	✠				$b_i = [b_{i1}]$
$Path$	p_{i1}				$p_i = [p_{i1}]$
$Time$	τ_{i1}				$\tau_i = [\tau_{i1}]$

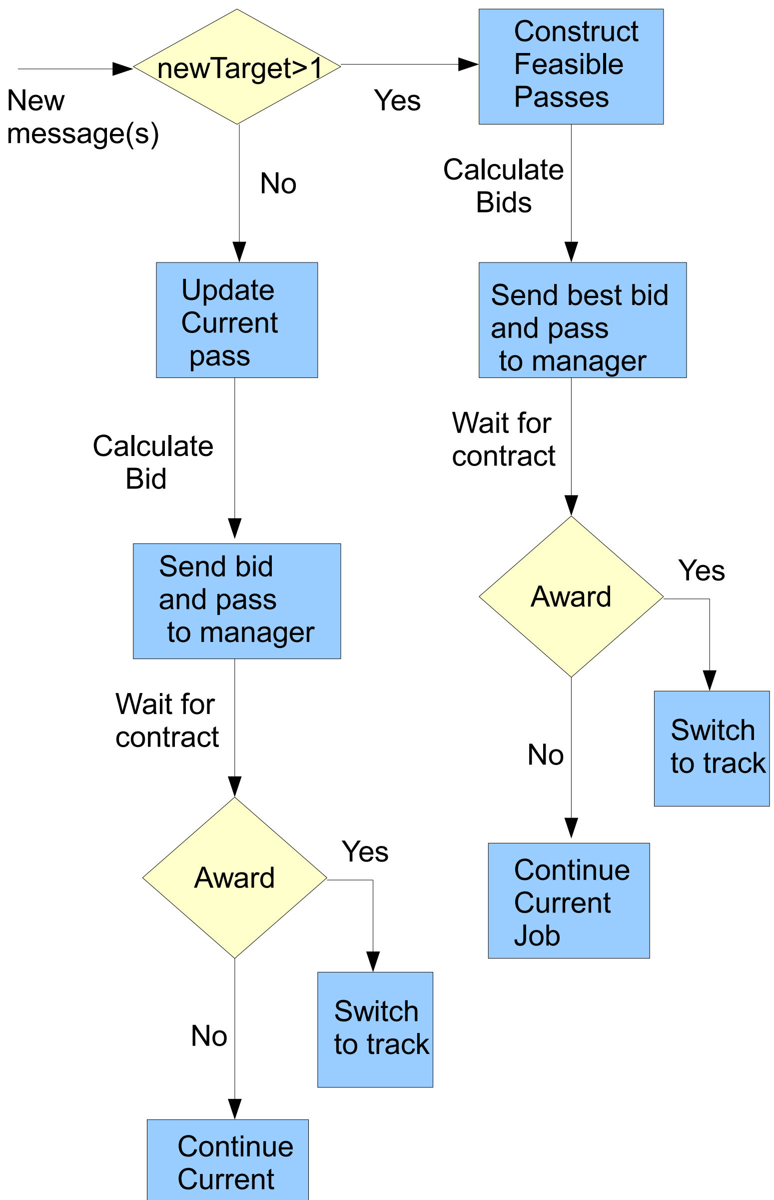
$Calculate\ arrival\ time\ \tau_{i1}(p)\ and\ corresponding\ bid\ \ y_{i1}$

Manager UAV_i	$Target_1$				$Values$
$Winning\ \ Agent$	i				$z_i = [z_{i1}]$
$WinningBids$	y_{i1}				$y_i = [y_{i1}]$

Manager statecharts



Potential contractors statecharts



Potential Contractors

Potential Contractor UAV_j					$Values$
$Bundle$					$b_j = []$
$Path$					$p_j = []$
$Time$					$\tau_j = []$

$Performing\ \ Search$

Potential Contractors

Potential Contractor UAV_j	$Target_1$				$Values$
$Bundle$	✠				$b_i = []$
$Path$					$p_i = []$
$Time$					$\tau_i = []$

$Recieve\ message$

Potential Contractors

Potential Contractor UAV_j	$Target_1$				$Values$
$Bundle$	✠				$b_j = [b_{j1}]$
$Path$	p_{j1}				$p_j = [p_{j1}]$
$Time$	τ_{j1}				$\tau_j = [\tau_{j1}]$

$Calculate\ arrival\ time\ \tau_{j1}(p)\ and\ corresponding\ bid\ y_{j1}$

Potential Contractor UAV_j	$Target_1$				$Values$
$Winning\ \ Agent$	j				$z_j = [z_{j1}]$
$WinningBids$	y_{j1}				$y_j = [y_{j1}]$

Potential Contractors

UAV_k		$Target_2$	$Target_k$		$Values$
$Bundle$		✓	✓		$b_k = [b_{k2}, b_{kk}]$
$Path$		p_{k2}	p_{kk}		$p_k = [p_{kk}, p_{k2}]$
$Time$		τ_{k2}	τ_{kk}		$\tau_k = [\tau_{kk}, \tau_{k2}]$

$Performing\ Tracking$

Potential Contractors

UAV_k	$Target_1$	$Target_2$	$Target_k$		$Values$
$Bundle$	$\maltese$	✓	✓		$b_k = [b_{k2}, b_{kk}]$
$Path$		p_{k2}	p_{kk}		$p_k = [p_{kk}, p_{k2}]$
$Time$		τ_{k2}	τ_{kk}		$\tau_k = [\tau_{kk}, \tau_{k2}]$

$Recieve\ \ message$

Potential Contractors

UAV_k	$Target_1$	$Target_2$	$Target_k$		$Values$
$Bundle$	$\checkmark$	$\checkmark$	$\checkmark$		$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{end} 1)$
$Path$		p_{k2}	p_{kk}		$p_k = [p_{kk}, p_{k2}]$
$Time$		τ_{k2}	τ_{kk}		$\tau_k = [\tau_{kk}, \tau_{k2}]$

$Update\ current\ bundle\ of\ targets$

Potential Contractors

UAV_k	$Target_1$	$Target_2$	$Target_k$		$Values$
$Bundle$	✓	✓	✓		$\mathbf{b}_k \leftarrow (\mathbf{b}_k \oplus_{end} 1)$
$Path$	p_{k1}	p_{k2}	p_{kk}		$\mathbf{p}_k \leftarrow (\mathbf{p}_k \oplus_{n_1} 1)$
$Time$		τ_{k2}	τ_{kk}		$\tau_k = [\tau_{kk}, \tau_{k2}]$

Update current pass

Potential Contractors

UAV_k	$Target_1$	$Target_2$	$Target_k$		$Values$
$Bundle$	✓	✓	✓		$\mathbf{b}_k \leftarrow (\mathbf{b}_k \oplus_{end} 1)$
$Path$	p_{k1}	p_{k2}	p_{kk}		$\mathbf{p}_k \leftarrow (\mathbf{p}_k \oplus_{n_1^*} 1)$
$Time$		τ_{k2}	τ_{kk}		$\tau_k = [\tau_{kk}, \tau_{k2}]$

Update current pass

$\Longrightarrow$ And optimal location n_1^* is then given by $n_1^* = \max_{n_1} c_1(\tau_{k1}^*(\mathbf{p}_k \oplus_{n_1} 1))$

Potential Contractors

UAV_k	$Target_1$	$Target_2$	$Target_k$		$Values$
$Bundle$	$\checkmark$	$\checkmark$	$\checkmark$		$\mathbf{b}_k \leftarrow (\mathbf{b}_k \oplus_{end} 1)$
$Path$	p_{k1}	p_{k2}	p_{kk}		$\mathbf{p}_k \leftarrow (\mathbf{p}_k \oplus_{n_1^*} 1)$
$Time$	τ_{k1}	τ_{k2}	τ_{kk}		$\tau_k \leftarrow (\tau_k \oplus_{n_{1*}} \tau_{k1}(\mathbf{p}_k \oplus_{n_{1*}}$

$Update\ current\ pass$

$\Longrightarrow$ And optimal location n_1^* is then given by $n_1^* = \max_{n_1} c_1(\tau_{k1}^*(\mathbf{p}_k \oplus_{n_1} 1))$

Potential Contractors

UAV_k	$Target_1$	$Target_2$	$Target_k$		$Values$
$Bundle$	$\checkmark$	$\checkmark$	$\checkmark$		$\mathbf{b}_k \leftarrow (\mathbf{b}_k \oplus_{end} 1)$
$Path$	p_{k1}	p_{k2}	p_{kk}		$\mathbf{p}_k \leftarrow (\mathbf{p}_k \oplus_{n_1^*} 1)$
$Time$	τ_{k1}	τ_{k2}	τ_{kk}		$\tau_k \leftarrow (\tau_k \oplus_{n_1*} \tau_{k1}(\mathbf{p}_k \oplus_{n_1*}$

$Update\ current\ pass$

$\Longrightarrow$ And optimal location n_1^* is then given by $n_1^* = \max_{n_1} c_1(\tau_{k1}^*(\mathbf{p}_k \oplus_{n_1} 1))$

$\Longrightarrow$ Then the final score for new task j(which is include $|b_k|$ targets) is $c_{kj}(\mathbf{p}_i) = c_j(\tau_{kj}^*(\mathbf{p}_k \oplus_{n_j^*} j))$

Compare bids

For case, when bundle of manager UAV3 was not empty $|b_3| \neq \emptyset$

	$Proposal1$	$Proposal2$	$Proposal3$
$UAV1$	c_{11}	-	-
$UAV2$	-	c_{22}	-
$UAV3$	-	-	c_{33}

Compare bids

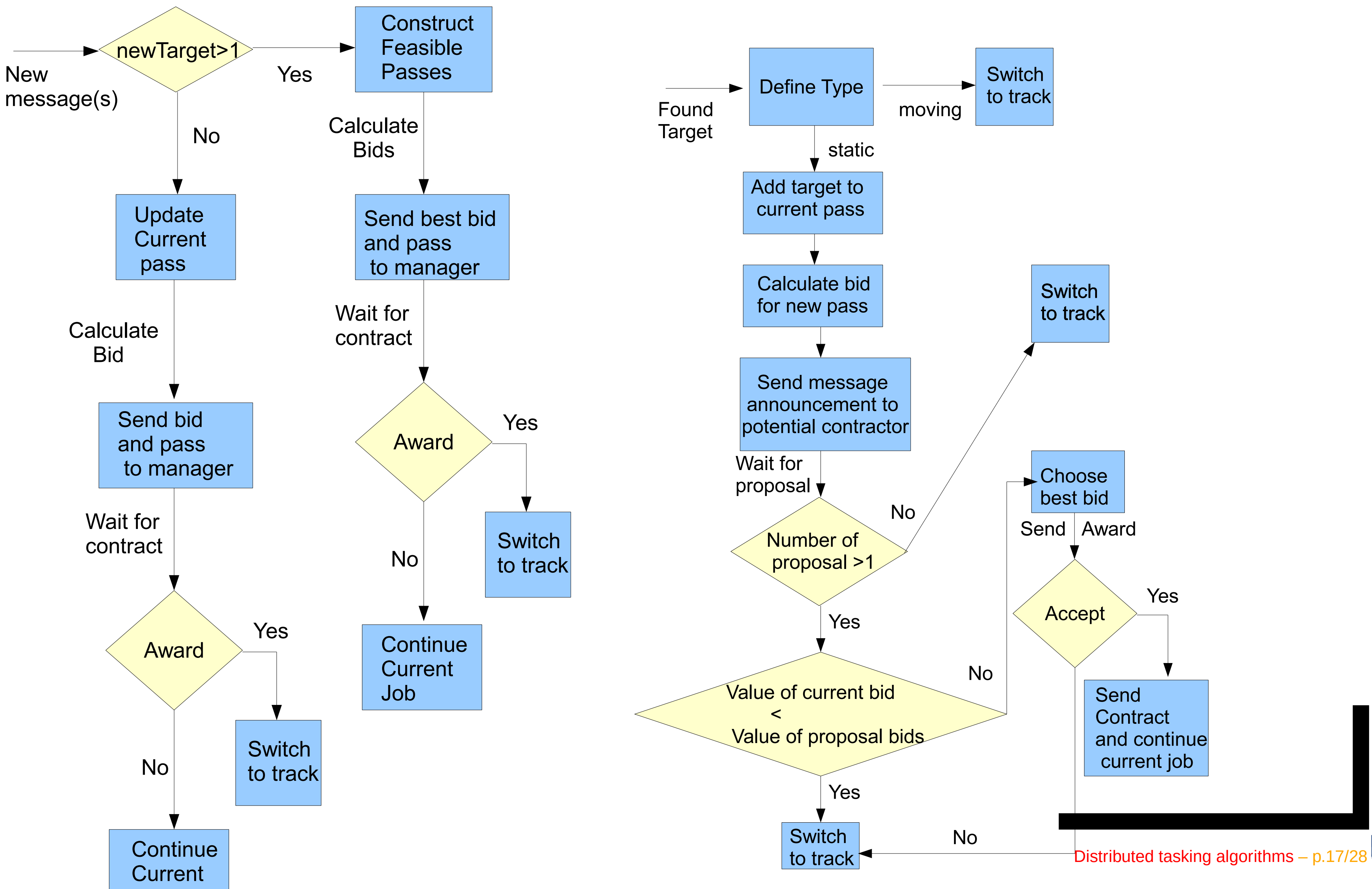
For case, when bundle of manager UAV3 was not empty $|b_3| \neq \emptyset$

	$Proposal1$	$Proposal2$	$Proposal3$
$UAV1$	c_{11}	-	-
$UAV2$	-	c_{22}	-
$UAV3$	-	-	c_{33}

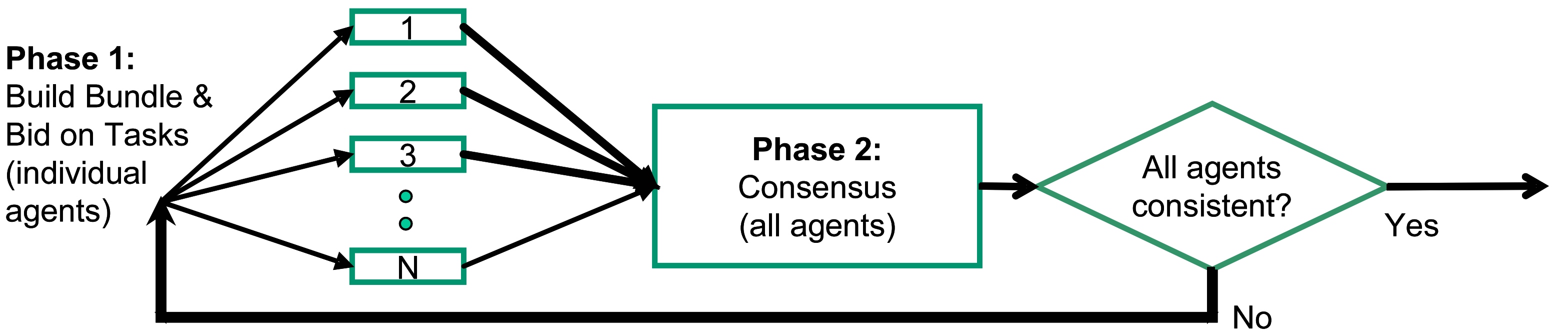
For case, when bundle of manager UAV3 was empty $|b_3| = \emptyset$

	$Proposal1$	$Proposal2$	$Proposal3$
$UAV1$	c_{11}	-	-
$UAV2$	-	c_{22}	-
$UAV3$	c_{31}	c_{32}	c_{33}

Potential contractors and Manager



Consensus-Based Bundle Algorithm



Core features of CBBA:

- Task selection - Polynomial-time, provably good approximate solutions
- Guaranteed real-time convergence even with inconsistent environment knowledge
- Time-varying score functions (e.g. time-windows of validity for tasks)

Application is the "Tethered UAVs Self-Assignment Problem": Find a logic that will enable the Tethered UAVs to self-deploy one UAV to each specified location.

Given:

- Locations (These are the locations pre-determined to be able to provide the necessary air-to-ground communications coverage for the ground users.)
- Homogenous UAVs

Problem statement

$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

$$subject\ \ to:$$

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

$$\sum_{i=1}^{N_a}\sum_{j=1}^{N_t} = N_{max}; \quad \sum_{i=1}^{N_a} x_{ij} \leq 1, \ \ \forall j \in \mathcal{J}$$

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

N_a Number of agents

N_t- Number of tasks

L_t- Maximum length of the bundle, i.e. each agent can be assigned a maximum L_t tasks

where $x_{ij} = 1$ if agent i is assigned to task j, and $\mathbf{x}_i \doteq \{x_{i1}, ..., x_{iN_t}\}$ is a vector of assignments for agent i, whose j-th element is x_{ij}.

The summation term in brackets in the objective function represents the local reward for agent i.

$\mathcal{I}$- Index set of agents where $\mathcal{I} \doteq \{1, ..., N_a\}$

$\mathcal{J}$- Index set of tasks where $\mathcal{J} \doteq \{1, ..., N_t\}$

$N_{max} \doteq \min\{N_t, N_a, L_t\}$

Problem statement

$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \rightarrow \max$$

$$subject\ \ to:$$

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

$$\sum_{i=1}^{N_a}\sum_{j=1}^{N_t} = N_{max}; \quad \sum_{i=1}^{N_a} x_{ij} \leq 1, \ \ \forall j \in \mathcal{J}$$

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

where $x_{ij} = 1$ if agent i is assigned to task j, and $\mathbf{x}_i \doteq \{x_{i1}, ..., x_{iN_t}\}$ is a vector of assignments for agent i, whose j-th element is x_{ij}.

The summation term in brackets in the objective function represents the local reward for agent i.

$\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$ - The variable length vector represent the path for agent i,an ordered sequence of tasks where the elements are the task indices, $p_{in} \in \mathcal{J}$ for $n = 1, ..., |\mathbf{p}_i|$, i.e. its n-th element is $j \in \mathcal{J}$ if agent i conducts task j at the n-th point along the path. The current length of the path is denoted by $|\mathbf{p}_i| \leq L_t$.

Problem statement

$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \rightarrow \max$$

$$subject\ \ to:$$

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

$$\sum_{i=1}^{N_a}\sum_{j=1}^{N_t} = N_{max}; \quad \sum_{i=1}^{N_a} x_{ij} \leq 1, \ \ \forall j \in \mathcal{J}$$

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

where $x_{ij} = 1$ if agent i is assigned to task j, and $\mathbf{x}_i \doteq \{x_{i1}, ..., x_{iN_t}\}$ is a vector of assignments for agent i, whose j-th element is x_{ij}.

The summation term in brackets in the objective function represents the local reward for agent i.

An assignment is said to be free of conflicts if each task is assigned to no more than one agent.

Key assumptions

$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

$$subject\ \ to:$$

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

$$\sum_{i=1}^{N_a}\sum_{j=1}^{N_t} = N_{max}; \quad \sum_{i=1}^{N_a} x_{ij} \leq 1, \ \ \forall j \in \mathcal{J}$$

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

- The score c_{ij} that agent i obtains by performing task j is defined as a function of the arrival time τ_{ij} at which the agent executes the task (or possibly the expected arrival time in a probabilistic setting).
- The arrival time τ_{ij} is uniquely defined as a function of the path $\mathbf{p}_i$ that agent i takes.
- The path $\mathbf{p}_i$ is uniquely defined by the assignment vector of agent i, $\mathbf{x}_i$.

Key assumptions

$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \rightarrow \max$$

$$subject\ \ to:$$

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

$$\sum_{i=1}^{N_a}\sum_{j=1}^{N_t} = N_{max}; \quad \sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J}$$

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

- The score c_{ij} that agent i obtains by performing task j is defined as a function of the arrival time τ_{ij} at which the agent executes the task (or possibly the expected arrival time in a probabilistic setting).
- The arrival time τ_{ij} is uniquely defined as a function of the path $\mathbf{p}_i$ that agent i takes.
- The path $\mathbf{p}_i$ is uniquely defined by the assignment vector of agent i, $\mathbf{x}_i$.

An example is the problem involving time-discounted values of targets, in which the sooner an agent arrives at the target, the higher the reward it obtains. Or for scenario involves re-visit tasks, where previously observed targets must be revisited at some scheduled time. In this case the score function would have its maximum at the desired re-visiting time and lower values at other re-visit times.

Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
  of variable length whose elements are defined by $b_{in} \in \mathcal{J}$ for $n = 1, ..., |\mathbf{b}_i|$. The current length of the bundle is denoted by b_i, which cannot exceed the maximum length L_t, and an empty bundle is represented by $b_i = \emptyset$ and $|\mathbf{b}_i| = 0$. The bundle represents the tasks that agent i has selected to do, and is ordered chronologically with respect to when the tasks were added (i.e. task b_{in} was added before task $b_{i(n+1)}$).
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
  whose elements are defined by $p_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$ for $n = 1, ..., |\mathbf{b}_i|$. The path contains the same tasks as the bundle, and is used to represent the order in which agent i will execute the tasks in its bundle. The path is therefore the same length as the bundle, and is not permitted to be longer than L_t; $|\mathbf{p}_i| = |\mathbf{b}_i| \leq L_t$.
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$

  whose elements are defined by τ_{in} for $n = 1, ..., |\tau_i|$. The times vector represents the corresponding times at which agent i will execute the tasks in its path, and is necessarily the same length as the path.
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t

  where each element $z_{ij} \in \{\mathcal{I} \cup \emptyset\}$ for $j = 1, ..., N_t$ indicates who agent i believes is the current winner for task j. Specifically, the value in element z_{ij} is the index of the agent who is currently winning task j according to agent i, and is $z_{ij} = \emptyset$; if agent i believes that there is no current winner.
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
  where the elements $y_{ij} \in [0, \infty)$ represent the corresponding winners bids and take the value of 0 if there is no winner for the task.
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

  where each element $s_{ik} \in [0, \infty)$ for $k = 1, ..., N_a$ represents the timestamp of the last information update agent i received about agent k, either directly or through a neighboring agent.

Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

$$\Downarrow$$

Each agent must carry these vectors of information in order to be able to perform decentralized algorithm which consists of iterations between two phases:
a bundle building phase where each vehicle greedily generates an ordered bundle of tasks, and a
consensus phase where conflicting assignments are identified and resolved through local communication between neighboring agents

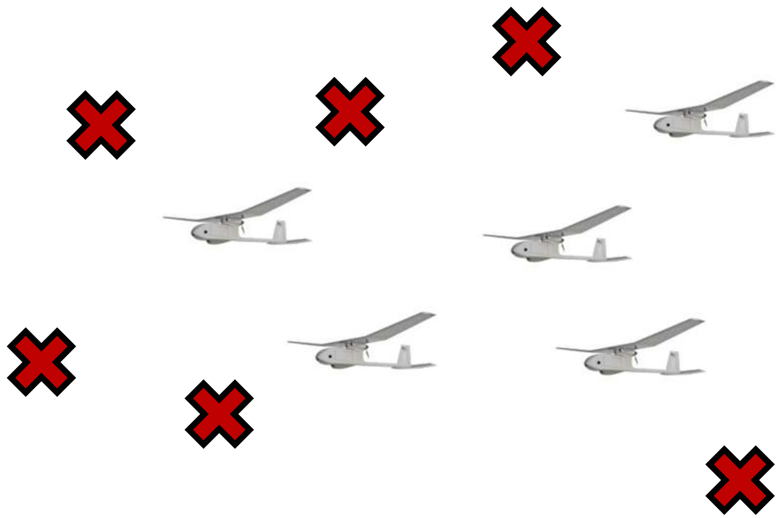
Six vectors of information for agent

- A bundle, $\mathbf{b}_i \doteq \{b_{i1}, ..., b_{i|\mathbf{b}_i|}\}$
- A corresponding path, $\mathbf{p}_i \doteq \{p_{i1}, ..., p_{i|\mathbf{p}_i|}\}$
- A vector of times $\tau_i \doteq \{\tau_{i1}, ..., \tau_{i|\tau_i|}\}$
- A winning agent list $\mathbf{z}_i \doteq \{z_{i1}, ..., z_{iN_t}\}$ of size N_t
- A winning bid list $\mathbf{y}_i \doteq \{y_{i1}, ..., y_{iN_t}\}$ of size N_t
- Vector of timestamps $\mathbf{s}_i \doteq \{s_{i1}, ..., s_{iN_a}\}$, of size N_a

$$\Downarrow$$

Algorithm will iterates between these two phases until no changes to the information vectors occur anymore.

Agent Information



$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

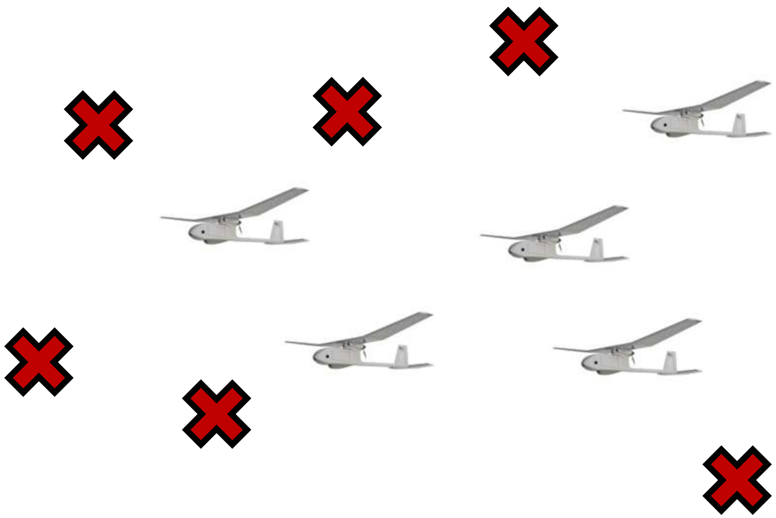
$$\sum_{j=1}^{N_t} x_{ij} \le L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \le 1, \quad \forall j \in \mathcal{J}$$

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

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	$\checkmark$		$\checkmark$		
$Path$					
$Time$					

Agent Information



$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

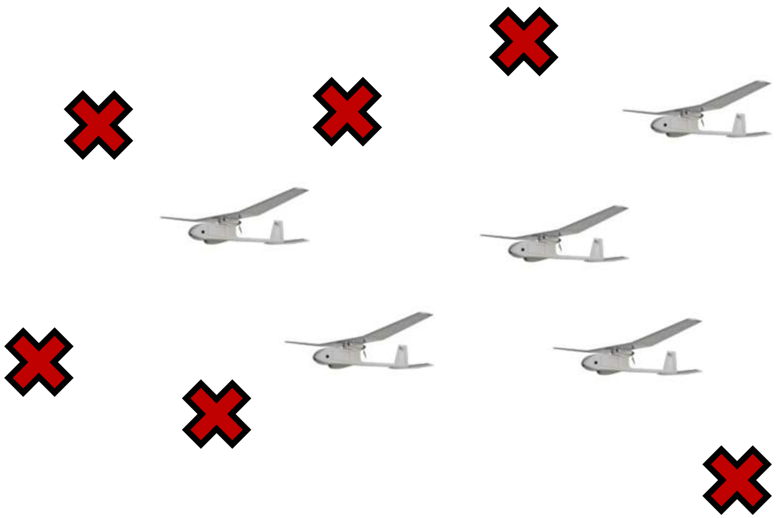
$$\sum_{j=1}^{N_t} x_{ij} \leq L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J}$$

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

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓		2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$					
$Time$					

Agent Information



$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

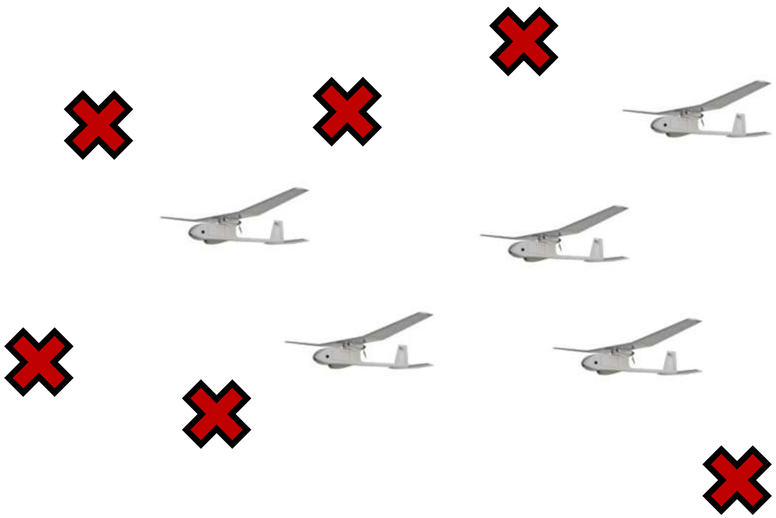
$$\sum_{j=1}^{N_t} x_{ij} \le L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \le 1, \quad \forall j \in \mathcal{J}$$

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

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓		2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$					

Agent Information



$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

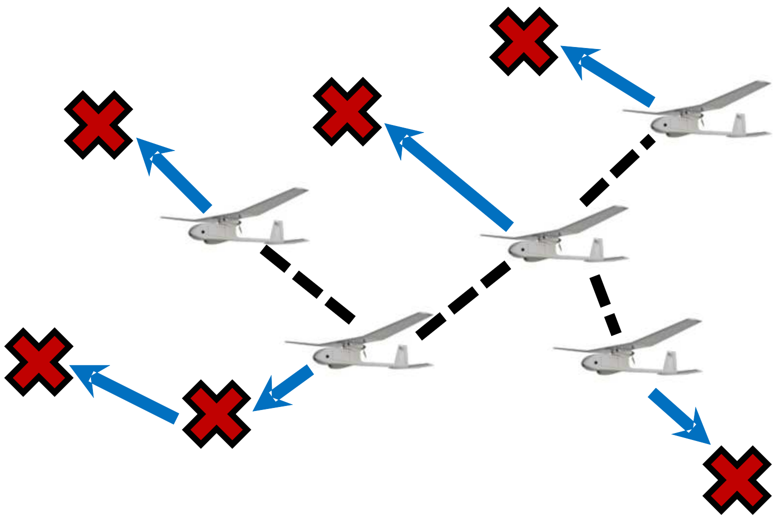
$$\sum_{j=1}^{N_t} x_{ij} \leq L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J}$$

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

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓		2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

Agent Information



$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \rightarrow \max$$

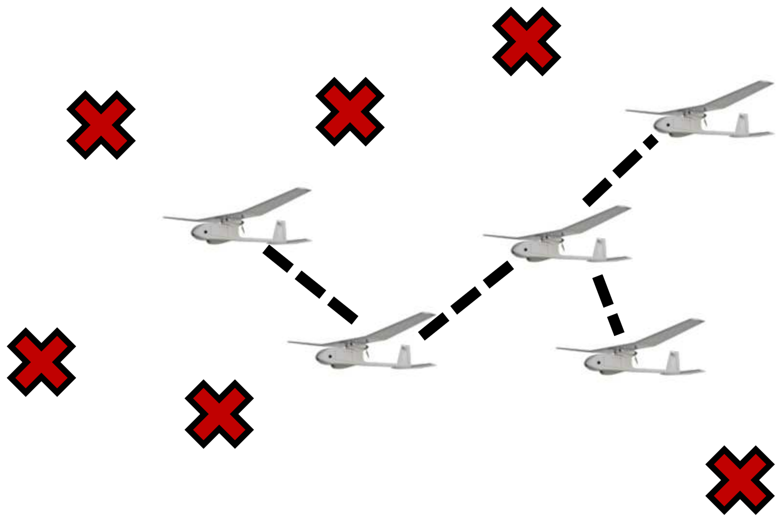
$$\sum_{j=1}^{N_t} x_{ij} \leq L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J}$$

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

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓		2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

Agent Information



$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

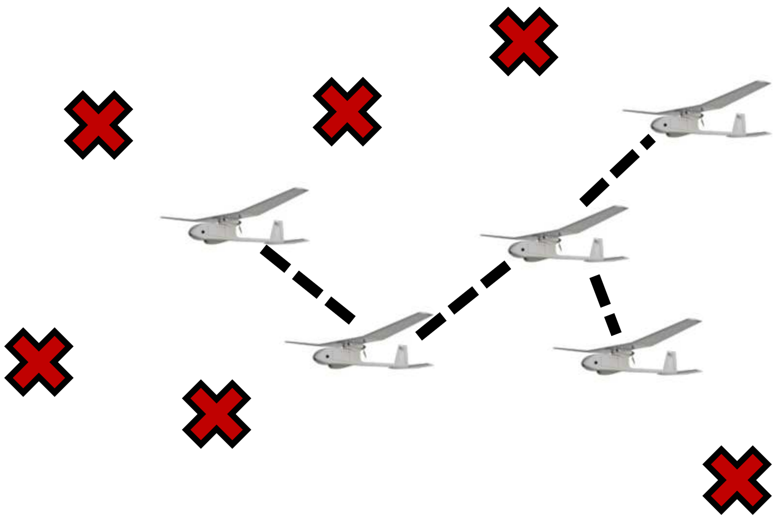
$$\sum_{j=1}^{N_t} x_{ij} \leq L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J}$$

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

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	2	4	i	k	$z_i = [z_{21}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$					

Agent Information



$$\sum_{i=1}^{N_a}\left(\sum_{j=1}^{N_t} c_{ij}(\tau_{ij}(\mathbf{p}_i(\mathbf{x_i})))x_{ij}\right) \to \max$$

$$\sum_{j=1}^{N_t} x_{ij} \le L_t, \quad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{N_a} x_{ij} \le 1, \quad \forall j \in \mathcal{J}$$

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

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 $\checkmark$		2 $\checkmark$		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$Calculate\ a\ score\ \ c_{ij} = c_{i2}\ \ and\ compare\ with\ current$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$Calculate\ a\ score\ \ c_{ij} = c_{i2}\ \ and\ compare\ with\ current$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$Calculate\ a\ score\ \ c_{ij} = c_{i2}\ \ and\ compare\ with\ current$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$To\ calculate\ best\ score\ for\ task\ j,\ we\ "insert"\ \ the\ task\ in\ some\ location\ n_j$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$To\ calculate\ best\ score\ for\ task\ j,\ we\ first\text{"}insert\text{"}\ the\ task\ in\ some\ location\ n_j$

And new path becomes $(\mathbf{p}_i \oplus_{n_j} j)$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$To\ calculate\ best\ score\ for\ task\ j,\ we\ first\text{"}insert\text{"}\ \ the\ task\ in\ some\ location\ n_j$

And new path becomes $(\mathbf{p}_i \oplus_{n_j} j)$ and second calculate the optimal execution time for this new path:

$$\tau^*_{ij}(\mathbf{p}_i \oplus_{n_j} j) = \max_{\tau_{ij} \in [0,\infty)} c_j(\tau_{ij})$$

$$subject\ \ to:$$

$$\tau^*_{ik}(\mathbf{p}_i \oplus_{n_j} j) = \tau^*_{ik}, \forall k \in \mathbf{p}_i$$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$\Longrightarrow$ optimal score for the task at location n_j is $c_j(\tau^*_{ij}(\mathbf{p}_i \oplus_{n_j} j))$.

$$\tau^*_{ij}(\mathbf{p}_i \oplus_{n_j} j) = \max_{\tau_{ij} \in [0,\infty)} c_j(\tau_{ij})$$

$$subject\ \ to:$$

$$\tau^*_{ik}(\mathbf{p}_i \oplus_{n_j} j) = \tau^*_{ik}, \forall k \in \mathbf{p}_i$$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$\Longrightarrow$ optimal score for the task at location n_j is $c_j(\tau^*_{ij}(\mathbf{p}_i \oplus_{n_j} j))$.

$\Longrightarrow$ And optimal location n^*_j is then given by $n^*_j = \max_{n_j} c_j(\tau^*_{ik}(\mathbf{p}_i \oplus_{n_j} j))$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$\Longrightarrow$ optimal score for the task at location n_j is $c_j(\tau_{ij}^*(\mathbf{p}_i \oplus_{n_j} j))$.

$\Longrightarrow$ And optimal location n_j^* is then given by $n_j^* = \max_{n_j} c_j(\tau_{ik}^*(\mathbf{p}_i \oplus_{n_j} j))$

$\Longrightarrow$ Final score for task j is $c_{ij}(\mathbf{p}_i) = c_j(\tau_{ij}^*(\mathbf{p}_i \oplus_{n_j^*} j))$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓	✠	$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

$\Longrightarrow$ optimal score for the task at location n_j is $c_j(\tau^*_{ij}(\mathbf{p}_i \oplus_{n_j} j))$.

$\Longrightarrow$ And optimal location n^*_j is then given by $n^*_j = \max_{n_j} c_j(\tau^*_{ik}(\mathbf{p}_i \oplus_{n_j} j))$

$\Longrightarrow$ Final score for task j is $c_{ij}(\mathbf{p}_i) = c_j(\tau^*_{ij}(\mathbf{p}_i \oplus_{n^*_j} j))$

$\Longrightarrow$ Final step is to select the highest scoring task to add to the bundle

$j^* = \max_{j \notin \mathbf{p}_i} c_{ij}(\mathbf{p}_i) h_{ij}$, where $h_{ij} = \mathbf{I}(c_{ij}(\mathbf{p}_i) > y_{ij})$ the indicator function

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✠	2 ✓		$b_i = [b_{i1}, b_{i2}]$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 $\checkmark$	$\checkmark$	2 $\checkmark$		$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{end} j^*)$
$Path$	2		1		$p_i = [p_{i1}, p_{i2}]$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✓	2 ✓		$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{end} j^*)$
$Path$	2		1		$\mathbf{p}_i \leftarrow (\mathbf{p}_i \oplus_{n_{j^*}} j^*)$
$Time$	20		10		$\tau_i = [\tau_{i1}, \tau_{i2}]$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✓	2 ✓		$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{end} j^*)$
$Path$	2		1		$\mathbf{p}_i \leftarrow (\mathbf{p}_i \oplus_{n_{j*}} j^*)$
$Time$	20		10		$\tau_i \leftarrow (\tau_i \oplus_{n_{j*}} \tau^*_{ij*}(\mathbf{p}_i \oplus_{n_{j*}} j^*$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	4	i	k	$z_i = [z_{i1}, z_{42}, z_{ik}, z_{kN_t}]$
$WinningBids$	9	5	8	7	$y_i = [y_{i1}, y_{42}, y_{ik}, y_{kN_t}]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
Bandle	1 ✓	✓	2 ✓		$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{end} j^*)$
Path	2		1		$\mathbf{p}_i \leftarrow (\mathbf{p}_i \oplus_{n_{j*}} j^*)$
Time	20		10		$\tau_i \leftarrow (\tau_i \oplus_{n_{j*}} \tau^*_{ij*}(\mathbf{p}_i \oplus_{n_{j*}} j^*$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
Winning Agent	i	i	i	k	$z_i = [z_{i1}, z_{i2}, ...]$
WinningBids	9	$c_{ij*}(\mathbf{p}_i)$	8	7	$y_i = [y_{i1}, y_{i2}, ...]$

Bundle construction(Task selection)

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Bandle$	1 ✓	✓	2 ✓		$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{end} j^*)$
$Path$	2		1		$\mathbf{p}_i \leftarrow (\mathbf{p}_i \oplus_{n_{j*}} j^*)$
$Time$	20		10		$\tau_i \leftarrow (\tau_i \oplus_{n_{j*}} \tau^*_{ij*}(\mathbf{p}_i \oplus_{n_{j*}} j^*$

$Bundle\ recursion\ continues\ until\ |\mathbf{b}_i| = L_t\ or\ h_{ij} = 0\ for\ all\ j \notin \mathbf{p}_i$

i	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$	i	i	i	k	$z_i = [z_{i1}, z_{i2}, \ldots]$
$WinningBids$	9	$c_{ij*}(\mathbf{p}_i)$	8	7	$y_i = [y_{i1}, y_{i2}, \ldots]$

Consensus

i, (receiver)	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$					$z_i = [z_{i1}, z_{i2}, ...]$
$WinningBids$					$y_i = [y_{i1}, y_{i2}, ...]$

$$Update: \quad z_{ij} = z_{kj}, \quad y_{ij} = y_{kj}$$

$$Reset: \quad z_{ij} = \emptyset, \quad y_{ij} = 0$$

$$Leave: \quad z_{ij} = z_{ij}, \quad y_{ij} = y_{ij}$$

k, (sender)	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$					$z_k = [z_{k1}, z_{k2}, ...]$
$WinningBids$					$y_k = [y_{k1}, y_{k2}, ...]$

Decision Rules

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
k	i	if $y_{kj} > y_{ij} \rightarrow update$
k	k	$update$
k	$m \notin \{i, k\}$	$if\ s_{km} > s_{im}$ or $y_{kj} > y_{ij} \rightarrow update$
k	$none$	$update$

$$s_{ik} = \begin{cases} \tau_r (i.e.\ message\ reception\ time), & if\ g_{ik} = 1; \\ \max\{s_{mk} \mid m \in \mathcal{I}, g_{im} = 1\}, & otherwise \end{cases}$$

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
i	i	$leave$
i	k	$reset$
i	$m \notin \{i, k\}$	$if\ s_{km} > s_{im} \rightarrow reset$
i	$none$	$leave$

Decision Rules

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
$m \notin \{i,k\}$	i	$if\ s_{km} > s_{im}\ and\ y_{kj} > y_{ij} \rightarrow update$
$m \notin \{i,k\}$	k	$if\ s_{km} > s_{im} \rightarrow update$ $else \rightarrow reset$
$m \notin \{i,k\}$	m	$s_{km} > s_{im} \rightarrow update$
$m \notin \{i,k\}$	$n \notin \{i,k,m\}$	$if\ s_{km} > s_{im}\ and\ s_{kn} > s_{in} \rightarrow update$ $if\ s_{km} > s_{im}\ and\ y_{kj} > y_{ij} \rightarrow update$ $if\ s_{kn} > s_{in}\ and\ s_{im} > s_{km} \rightarrow reset$
$m \notin \{i,k\}$	$none$	$if\ s_{km} > s_{im} \rightarrow update$

Agent k thinks z_{kj} is	Agent i thinks z_{ij} is	Receiver Action
$none$	i	$leave$
$none$	k	$update$
$none$	$m \notin \{i,k\}$	$if\ s_{km} > s_{im} \rightarrow update$
$none$	$none$	$leave$

Decision Rules

i, (receiver)	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$					$z_i = [z_{i1}, z_{i2}, ...]$
$WinningBids$					$y_i = [y_{i1}, y_{i2}, ...]$

$$Update: \quad z_{ij} = z_{kj}, \quad y_{ij} = y_{kj}$$

$$Reset: \quad z_{ij} = \emptyset, \quad y_{ij} = 0$$

$$Leave: \quad z_{ij} = z_{ij}, \quad y_{ij} = y_{ij}$$

k, (sender)	$Task_1$	$Task_2$	$Task_k$	$Task_{N_t}$	$Values$
$Winning\ \ Agent$					$z_k = [z_{k1}, z_{k2}, ...]$
$WinningBids$					$y_k = [y_{k1}, y_{k2}, ...]$

Algorithm summary

- Calculate marginal score for all tasks

$$c_{ij}(\mathbf{p}_i) = \begin{cases} 0, & if\ j \in \mathbf{p}_i; \\ \max\limits_{n \le l_b} S_{path}(\mathbf{p}_i \oplus_n j) - S_{path}(\mathbf{p}_i), & otherwise \end{cases}$$

- Determine which tasks are winnable
- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*
- If $c_{ij^*} \le 0$, then return. otherwise, continue
- Update agent information
- Update shared information vectors
- if $l_b = L_t$, then return, otherwise, go to 1.

Algorithm summary

- Calculate marginal score for all tasks
- Determine which tasks are winnable

  $$h_{ij} = \mathbf{I}(c_{ij}(\mathbf{p}_i) > y_{ij}), \forall j \in \mathcal{J}$$

- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*
- If $c_{ij^*} \leq 0$, then return. otherwise, continue
- Update agent information
- Update shared information vectors
- if $l_b = L_t$, then return, otherwise, go to 1.

Algorithm summary

- Calculate marginal score for all tasks
- Determine which tasks are winnable
- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*

$$j^* = \max_{j \in \mathcal{J}} c_{ij} h_{ij}$$

$$n_j^* = \max_{n \in \{0,\ldots,l_b\}} S_{path}(\mathbf{p}_i \oplus_n j^*)$$

- If $c_{ij^*} \leq 0$, then return. otherwise, continue
- Update agent information
- Update shared information vectors
- if $l_b = L_t$, then return, otherwise, go to 1.

Algorithm summary

- Calculate marginal score for all tasks
- Determine which tasks are winnable
- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*
- If $c_{ij^*} \leq 0$, then return. otherwise, continue

- Update agent information
- Update shared information vectors
- if $l_b = L_t$, then return, otherwise, go to 1.

Algorithm summary

- Calculate marginal score for all tasks
- Determine which tasks are winnable
- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*
- If $c_{ij^*} \leq 0$, then return. otherwise, continue
- Update agent information

$$\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{l_b} j^*)$$

$$\mathbf{p}_i \leftarrow (\mathbf{p}_i \oplus_{n_j^*} j^*)$$

- Update shared information vectors
- if $l_b = L_t$, then return, otherwise, go to 1.

Algorithm summary

- Calculate marginal score for all tasks
- Determine which tasks are winnable
- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*
- If $c_{ij^*} \leq 0$, then return. otherwise, continue
- Update agent information
- Update shared information vectors

$$y_{i(j^*)} = c_{i(j^*)}$$

$$z_{i(j^*)} = i$$

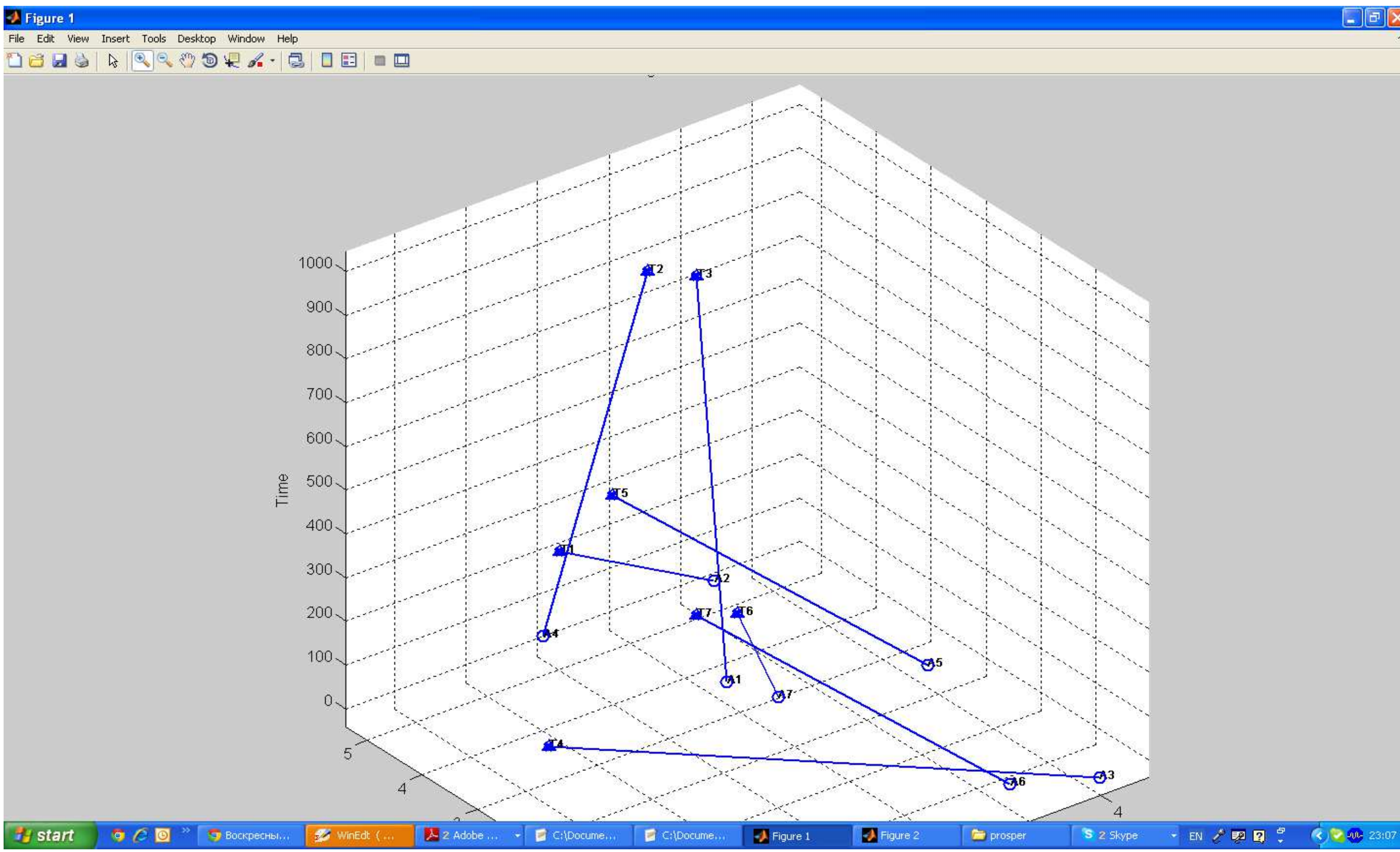
- if $l_b = L_t$, then return, otherwise, go to 1.

Algorithm summary

- Calculate marginal score for all tasks
- Determine which tasks are winnable
- Select the index of the best eligible task, j^*, and select best location in the plan to insert the task, n_j^*
- If $c_{ij^*} \leq 0$, then return. otherwise, continue
- Update agent information
- Update shared information vectors
- if $l_b = L_t$, then return, otherwise, go to 1.

Simulation

The solution of Tethered UAVs Self-Assignment problem presented in a figure below, namely, we find the logic that enabled the tethered UAVs (7 UAVs) to self-deploy one UAV to each specified location (7 locations).



The end

Thank you!