Task Allocation with Executable Coalitions in Multirobot Tasks

Yu ("Tony") Zhang   Lynne E. Parker

Distributed Intelligence Laboratory
Electrical Engineering and Computer Science Department
University of Tennessee, Knoxville TN, USA

IEEE International Conference on
Robotics and Automation, 2012

Task allocation with multirobot tasks

Multirobot tasks:

Individual robots may not have all the required capabilities

$\Longrightarrow$



Task allocation:

- A set of robots, $R = \{r_1, r_2, ...\}$
- A set of tasks to be assigned, $T = \{t_1, t_2, ...\}$

Find assignments in $C \rightarrow T$, $C = 2^R$

Task allocation with multirobot tasks

Given:

- Each robot r_i is associated with a vector $\boldsymbol{B}_i$ capabilities
- Each task t_l requires a vector $\boldsymbol{P}_l$ capabilities
- A vector $\boldsymbol{W}$ of costs for capabilities
- A vector $\boldsymbol{V}$ of rewards for tasks
- Communication and coordination costs, $C \times T \rightarrow \Re^0$
- A utility function U for $m_{jl} = c_j \rightarrow t_l$, defined as $U(m_{jl}) =$

$$\begin{cases} \boldsymbol{V}[l] - \sum_h \boldsymbol{P}_l[h]\boldsymbol{W}[h] - Cost(c_j, t_l) & \text{if } \forall h : \sum_{r_i \in c_j} \boldsymbol{B}_i[h] \geq \boldsymbol{P}_l[h] \\ 0 & \text{otherwise} \end{cases}$$

$\max \sum U(m)$ A NP-hard problem

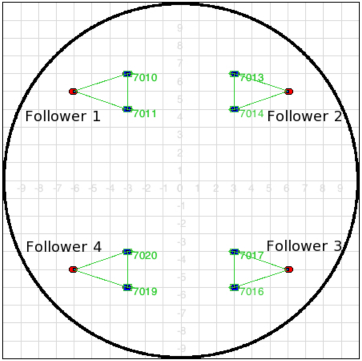
NP-hardness of the task allocation problem

The NP-hardness is partly due to the size of $C = 2^R$

To reduce the number of coalitions $|C|$, previous approaches:

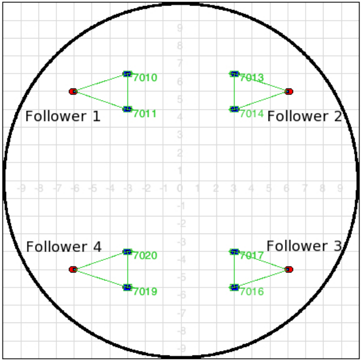
1. Consider coalitions that satisfy $\sum_{r_i \in c_j} \boldsymbol{B}_i[h] \geq \boldsymbol{P}_l[h]$
2. Restrict coalition size to k, $C = O(R^k)$
3. Group homogeneous robots, thus reducing $|R| = \sum_k N_k$ to k

Previous approaches to reduce $|C|$ – Approach 1



Considering coalitions that satisfy task capability requirement (i.e., $\sum_{r_i \in c_j} \boldsymbol{B}_i[h] \geq \boldsymbol{P}_l[h]$)

Previous approaches to reduce $|C|$ – Approach 1

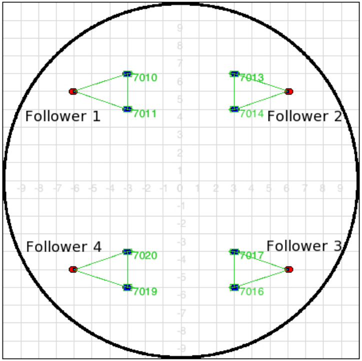


Considering coalitions that satisfy task capability requirement (i.e., $\sum_{r_i \in c_j} \boldsymbol{B}_i[h] \geq \boldsymbol{P}_l[h]$)

Issue:

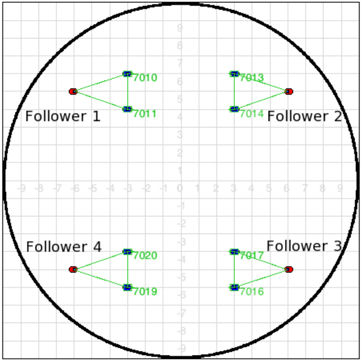
$|C|$ can still be exponential in R

Previous approaches to reduce $|C|$ – Approach 2



Restricting coalition size to k,
$C = O(R^k)$

Previous approaches to reduce $|C|$ – Approach 2

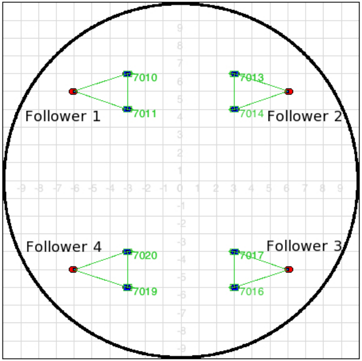


Restricting coalition size to k,
$C = O(R^k)$

Issues:

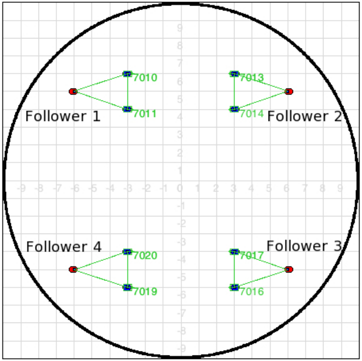
1. $|C|$ is a high order polynomial of $|R|$
2. Task execution can be super-additive

Previous approaches to reduce $|C|$ – Approach 3



Grouping homogeneous robots, thus reducing $|R|$ to k

Previous approaches to reduce $|C|$ – Approach 3



Grouping homogeneous robots, thus reducing $|R|$ to k

Issue:

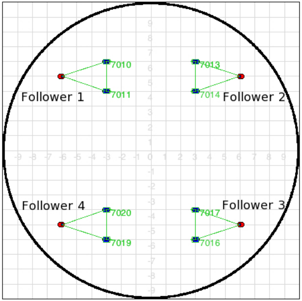
Homogeneous robots are often not equivalent

Contributions

1. To address the previous issues for reducing $|C|$:
   - Introduce task allocation with *executable coalitions*
2. To perform task allocation:
   - Apply a layering technique to perform task allocation with executable coalitions
3. For tasks with no executable coalitions:
   - Introduce a process that decomposes unsatisfied task preconditions to create task plans

Executable coalitions: coalitions that are feasible for task execution in the current situation

What determines an executable coalition



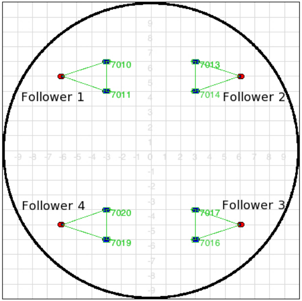
Approach 1, 2 (task capability, restrict size)

Type	Capabilities	Robot ID
1	Fiducial, Laser	(1 - 4)
2	Fiducial, Laser, Localization	(5 - 12)

Approach 3 (grouping)

Type	Capabilities	Count
1	Fiducial, Laser	4
2	Fiducial, Laser, Localization	8

What determines an executable coalition



Approach 1, 2 (task capability, restrict size)

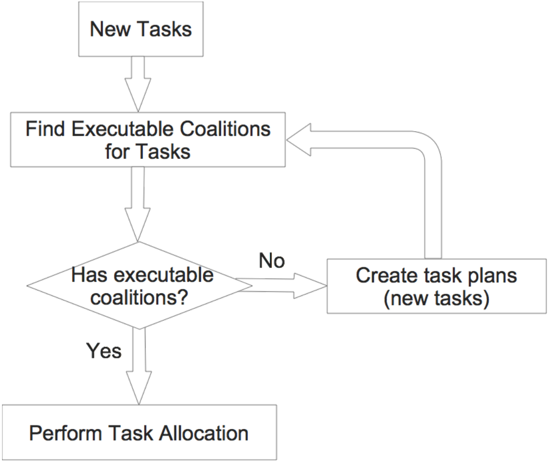
Type	Capabilities	Robot ID
1	Fiducial, Laser	(1 - 4)
2	Fiducial, Laser, Localization	(5 - 12)

Approach 3 (grouping)

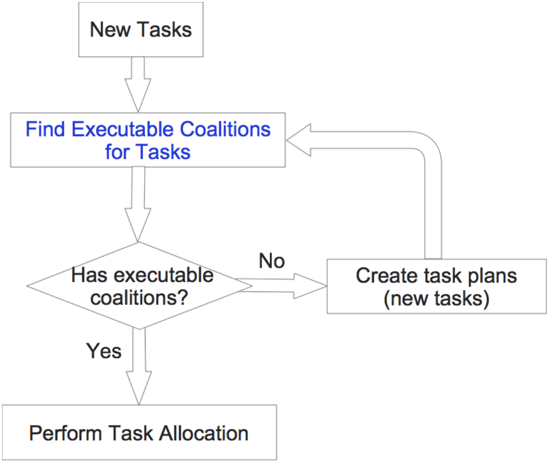
Type	Capabilities	Count
1	Fiducial, Laser	4
2	Fiducial, Laser, Localization	8

The desired configurations (i.e., *preconditions*) for task execution are not considered, which determine whether a coalition is executable.

Process flow



Process flow

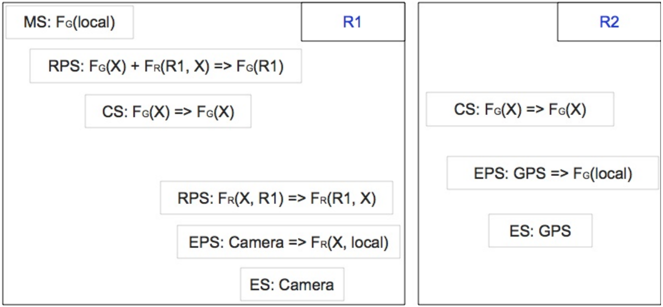


Forming executable coalitions with IQ-ASyMTRe

IQ-ASyMTRe [Zhang and Parker, 2010b] defines robot capabilities as:

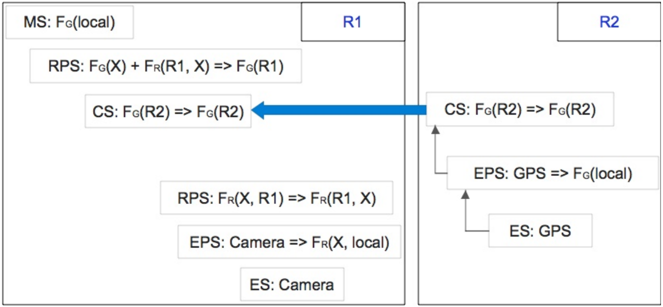
- Motor Schema (MS)
- Environmental Sensor (ES)
- Perceptual Schema (PS)
- Communication Schema (CS)

Forming executable coalitions with IQ-ASyMTRe



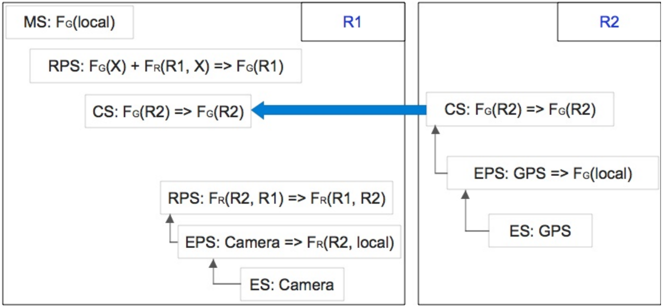
An example

Forming executable coalitions with IQ-ASyMTRe



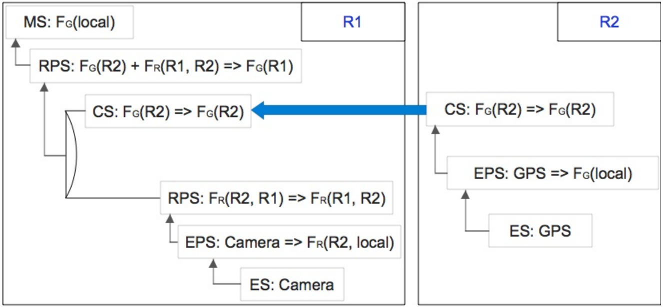
An example

Forming executable coalitions with IQ-ASyMTRe



An example

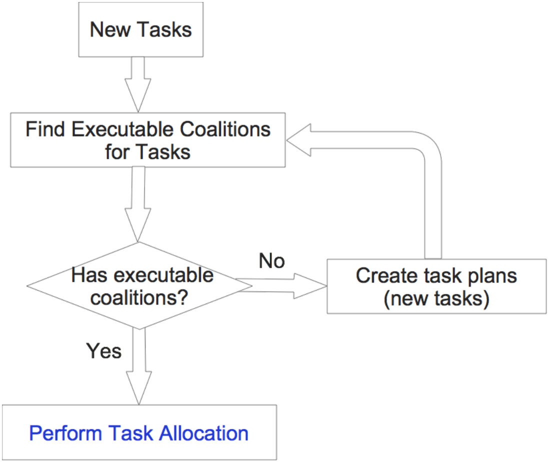
Forming executable coalitions with IQ-ASyMTRe



An example

Required information flow → proper configurations → satisfied preconditions

Process flow



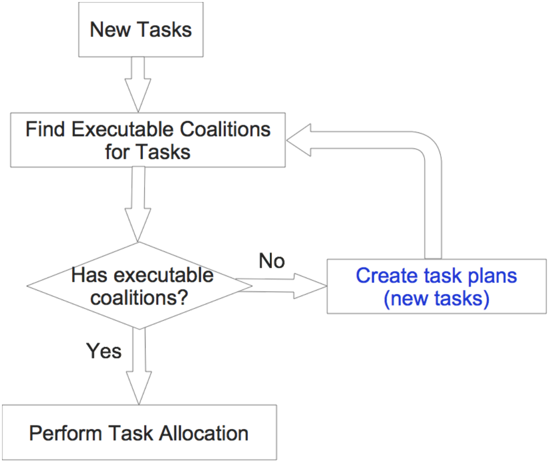
Layering task allocation

Layer with any task allocation algorithm:

Forming executable coalitions with IQ-ASyMTRe → Task allocation

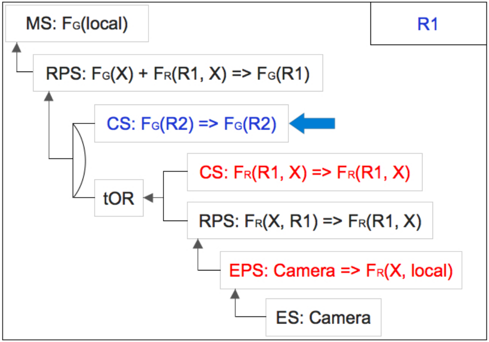
- Schemas → Robot capabilities
- Desired motor schemas → Required task capabilities
- Schema cost → Capability cost
- Task reward
- CS cost → Communication and coordination costs

Process flow



Tasks with no executable coalitions

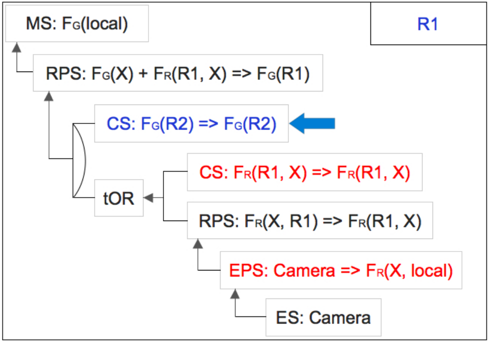
Extending MS to be capable of outputting information:



An illustrative example for using IMS

Tasks with no executable coalitions

Extending MS to be capable of outputting information:



An illustrative example for using IMS

Divide unsatisfied preconditions into satisfiable components

Tasks with no executable coalitions

Auctioneer Process

Create empty *new_task* and *announced_task* lists.
while true **do**
    Receive new tasks and put them on the *new_task* list.
    for all tasks in *announced_list* that are initiating tasks for the new IMS tasks received **do**
        Update the task's preconditions.
        Move the task from *announced_list* to *new_list*.
    end for
    IMS Auction: announce tasks in *announced_list*.
    Easy Auction: announce tasks in *new_task* list for which preconditions are satisfied.
    Move the announced tasks to *announced_list*.
    Wait a while for bids.
    Collect bids from robots.
    Invoke task allocation algorithms to determine the task assignments.
    Remove tasks that are assigned from *new_task* list.
    Move tasks for which no bids are submitted or no bids are beneficial to *announced_list*.
end while

Tasks with no executable coalitions

Auctioneer Process

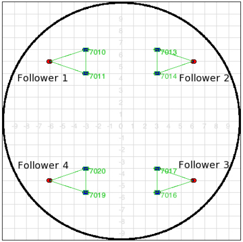
```
Create empty new_task and announced_task lists.
while true do
    Receive new tasks and put them on the new_task list.
    for all tasks in announced_list that are initiating tasks
    for the new IMS tasks received do
        Update the task's preconditions.
        Move the task from announced_list to new_list.
    end for
    IMS Auction: announce tasks in announced_list.
    Easy Auction: announce tasks in new_task list for which
    preconditions are satisfied.
    Move the announced tasks to announced_list.
    Wait a while for bids.
    Collect bids from robots.
    Invoke task allocation algorithms to determine the task
    assignments.
    Remove tasks that are assigned from new_task list.
    Move tasks for which no bids are submitted or no bids
    are beneficial to announced_list.
end while
```

Robot Process

```
while true do
    if the robot has a winning bid then
        Set up the coalition and execute the task.
    end if
    Receive new task announcements.
    for all received tasks do
        if task announced for Easy Auction then
            Invoke IQ-ASyMTRe to search for
            executable coalitions and submit bids.
        else if task announced for IMS Auction then
            Invoke IQ-ASyMTRe to submit
            information task requests.
        end if
    end for
end while
```

Task allocation with executable coalitions

Example configurations

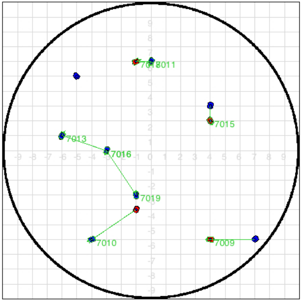


Previous approaches to reduce $|C|$ vs. Executable coalitions

Followers can navigate / # followers	# executable coalitions	# coalitions for approach 1	# executable coalitions with $k = 3$	# coalitions combining approach 1, 2 with $k = 3$
4/4	12	3824	12	192

Task allocation with executable coalitions

What about in random configurations?



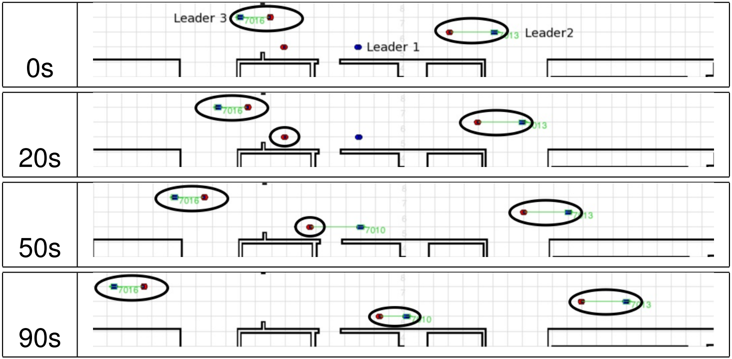
Task allocation with executable coalitions

Previous approaches to reduce $|C|$ vs. Executable coalitions

Followers can navigate / # followers	# executable coalitions	# coalitions for approach 1	# executable coalitions with $k = 3$	# coalitions combining approach 1, 2 with $k = 3$
4/4	33	3824	17	192
3/4	13	3824	9	192
2/4	3	3824	3	192
2/4	5	3824	3	192
2/4	6	3824	5	192
1/4	3	3824	3	192
2/4	15	3824	9	192
4/4	4	3824	4	192
4/4	11	3824	9	192
4/4	12	3824	11	192

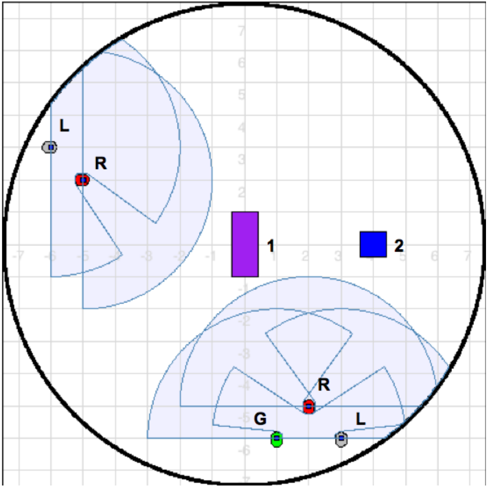
Task allocation with executable coalitions can significantly reduce the number of coalitions

Tasks with no executable coalitions



Robots can autonomously create task plans

A general scenario



Contributions

- Introduce task allocation with executable coalitions
- Apply a layering technique to perform task allocation with IQ-ASyMTRe
- Introduce a process that decomposes unsatisfied task preconditions to create task plans

References

Parker, L. and Tang, F. (2006).
Building multirobot coalitions through automated task solution synthesis.
Proc. of the IEEE, 94(7):1289–1305.

Zhang, Y. and Parker, L. (2010a).
A general information quality based approach for satisfying sensor constraints in multirobot tasks.
In *IEEE International Conference on Robotics and Automation*.

Zhang, Y. and Parker, L. (2010b).
IQ-ASyMTRe: Synthesizing coalition formation and execution for tightly-coupled multirobot tasks.
In *IEEE/RSJ Int'l Conference on Intelligent Robots and Systems*.