# Solution Space Reasoning to Improve IQ-ASyMTRe in Tightly-Coupled Multirobot Tasks

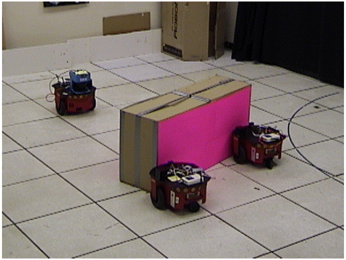
Yu ("Tony") Zhang Lynne E. Parker

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

IEEE International Conference on
Robotics and Automation, 2011

# Tightly-coupled multirobot tasks

- Tight coordinations through explicit or implicit capability sharing.



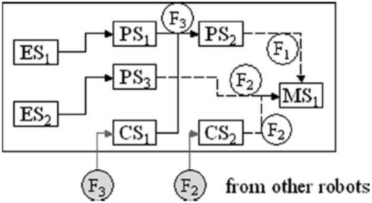
(a) [Gerkey and Mataric, 2001]

(b) [Parker and Tang, 2006]

# ASyMTRe

ASyMTRe [Parker and Tang, 2006] divides robot capabilities into:

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



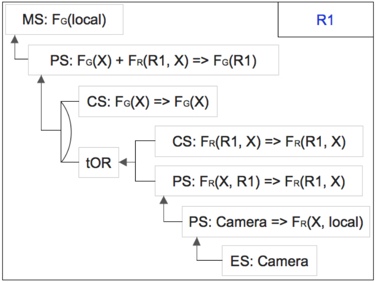

(a) [Parker and Tang, 2006]

Capability sharing is achieved through communication

# IQ-ASyMTRe

IQ-ASyMTRe [Zhang and Parker, 2010] addresses several limitations of ASyMTRe by:

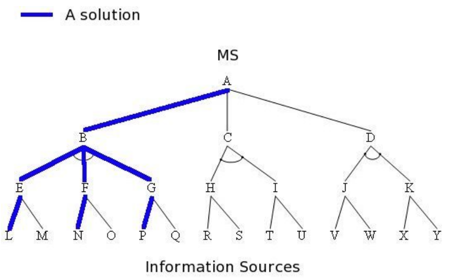
- Introducing a complete reference of information.
- Introducing information conversions.
- Incorporating information quality in coalition formation.



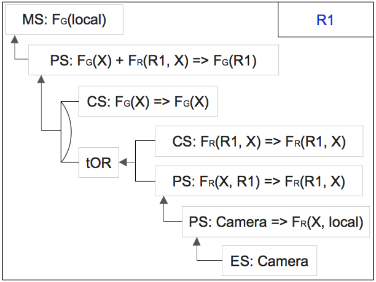

(b) A solution space

# Issues of IQ-ASyMTRe

The number of potential solutions can grow exponentially:



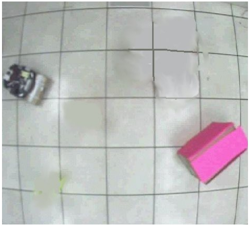

(a) Exponential solution space




(b) A solution space

# Issues of IQ-ASyMTRe

A single behavior can be implemented by multiple MSs:

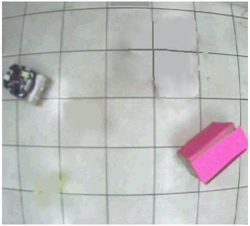


(a) Navigating with an overhead camera

(b) Navigating with a localization capability

# Issues of IQ-ASyMTRe

A single behavior can be implemented by multiple MSs:



(a) Navigating with an overhead camera

(b) Navigating with a localization capability

How to utilize these MSs for more flexible execution?

# Contributions

- *Introducing the independence of information instances*

  Reduces from an exponential to a linear search space for practical applications

- *Identifying and removing unnecessary potential solutions in the solution space*

  Improves the search performance further

- *Relating behaviors with information requirements and providing a method to utilize different MSs*

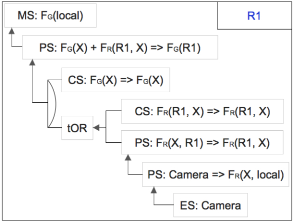
  Achieves more flexibility during execution

# Issues of IQ-ASyMTRe

- The number of potential solutions can grow exponentially
  - How to improve the search performance?
- A single behavior can be implemented by multiple MSs
  - How to utilize different MSs for more flexible execution?

# Performance analysis of IQ-ASyMTRe

For one information instance




- $N_c$: the max # of RPSs -> the same information type
- $N_t$: the # of information types related to the information instance
- $N_r$: the max # of referents associated with information instances

In the worst case, the # of potential solutions is $O(N_t 2^{N_r} N_c^{N_t 2^{N_r}})$

# For multiple information instances

In a robot searching task:

$$F_R(target, X) \text{ and } F_G(X)$$

| $F_R(target, X)$ | $F_G(X)$ |
|---|---|
| 1. $F_R(target, r_1) + F_R(r_1, X)$ | 1. $F_R(X, r_2) + F_G(r_2)$ |
| 2. $F_R(target, camera) + F_R(camera, X)$ | 2. $F_R(X, r_3) + F_G(r_3)$ |

# For multiple information instances

In a robot searching task:

$$F_R(target, X) \text{ and } F_G(X)$$

| $F_R(target, X)$ | $F_G(X)$ |
|---|---|
| 1. $F_R(target, r_1) + F_R(r_1, X)$ | 1. $F_R(X, r_2) + F_G(r_2)$ |
| 2. $F_R(target, camera) + F_R(camera, X)$ | 2. $F_R(X, r_3) + F_G(r_3)$ |

Exponential growth!

# Independence of information instances

**Definition**

*An information instance is independent of another if there are no uninstantiated referents labeled the same in both information instances (such referents are required to be instantiated to the same entity).*

$F_R(target, X)$ and $F_G(X)$, $X$ must be instantiated the same

Vs.

$F_R(target, r_1)$ and $F_G(r_1)$, referents are all instantiated

# Performance improvement

Let

- $N_i$: the number of information instances to be reasoned about

Group information instances into mutually independent sets:

- $H$: the maximum cardinality of all independent sets

Complexity to search the solution space:

$$O(exp(N_i)) \Rightarrow O(N_i exp(H))$$

# Unnecessary potential solutions

Unnecessary potential solutions:

- RPS (i.e., $F_R(Y, X) \rightarrow F_R(X, Y)$): $\{F_R(r_1, r_2)\}$ and $\{F_R(r_2, r_1)\}$
- Uninstantiated referents: $\{F_G(X), F_R(X, r_1)\}$ and $\{F_G(Y), F_R(Y, r_1)\}$

**Lemma**

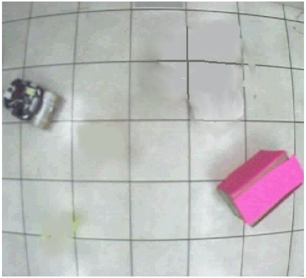
*For any two potential solutions, if they have the same number of distinct labels (including instantiated referents) and for all distinct labels in one, we can sequentially find a matching label not previously matched in the other with the same set of information types having the same instantiated referents and the same referent instantiation constraints, then only one of them is necessary.*

# Issues of IQ-ASyMTRe

- The number of potential solutions can grow exponentially
  - How to improve the search performance?
- A single behavior can be implemented by multiple MSs
  - How to utilize different MSs for more flexible execution?

# Input information requirement

Different MSs have different input information requirements:



(a) $F_R(X, robot), F_R(X, goal)$

(b) $F_G(robot), F_G(goal)$

# Expressive ability of information instances

**Lemma**

*The semantic meaning related to any information requirement can be expressed using information instances exactly. Furthermore, the finite set of information instances required for the exact expression is always the same.*

Given certain assumptions, we have

**Theorem**

*For any behavior, given that the exact information requirement has a finite representation, the MinIIS with a finite representation exists and is unique.*

# Utilizing different MSs

Although we cannot determine the MinIIS, it can be approximated.

*Algorithm for approximation*

**for all** $IIS_i \in$ known options **do**
  Compute $S_i = P(IIS_i)$.
**end for**
**return** $S = \cap_i(S_i)$.

# Utilizing different MSs

Although we cannot determine the MinIIS, it can be approximated.

*Algorithm for approximation*

**for all** $IIS_i \in$ known options **do**
  Compute $S_i = P(IIS_i)$.
**end for**
**return** $S = \cap_i(S_i)$.

For the *go-to-goal* behavior,

- MS 1. $\{F_G(robot), F_G(goal)\}$
- MS 2. $\{F_R(X, robot), F_R(X, goal)\}$

Output: $F_R(robot, goal)$

# Unnecessary potential solutions

Unnecessary potential solutions can be removed

Table: SOLUTION SPACE COMPARISON

| |
|---|
| 1. Laser:$F_G(local)$ |
| 2. Fiducial:$F_R(X, local)$, CS:$F_G(X)$ |
| 3. CS:$F_G(X)$, CS:$F_R(local, X)$ |
| 4. CS:$F_G(X)$, CS:$F_R(X, local)$ |
| 5. CS:$F_G(X)$, CS:$F_R(local, X)$ |
| 6. CS:$F_G(X)$, CS:$F_R(local, Y)$, CS:$F_R(X, Y)$ |
| 7. CS:$F_G(X)$, CS:$F_R(X, Y)$, CS:$F_R(local, Y)$ |

# Independence of information instances

MinIIS is associated with more potential solutions

Table: MinIIS AND INDEPENDENCE OF INFORMATION INSTANCE

| Op. | **Go-to-goal** | # PoSs | After rem. | Use Ind. |
|---|---|---|---|---|
| 1 | $F_G(local), F_G(goal)$ | 18 | 10 | 7 |
| 2 | $F_R(goal, local)$ | 31 | 15 | 15 |
| Op. | **Push-box** | | | |
| 1 | $F_G(local), F_G(box), F_G(goal)$ | 36 | 20 | 9 |
| 2 | $F_R(box, local), F_R(goal, local)$ | 961 | 185 | 30 |
| | Time for a full search (s) | 15.1 | 2.9 | 0.02 |
| | Time for removal from 961 (s) | | 14.8 | 0.02 |

# Independence of information instances

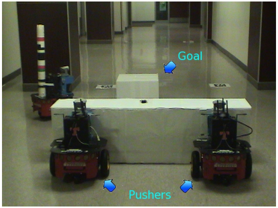
Performance is significantly improved

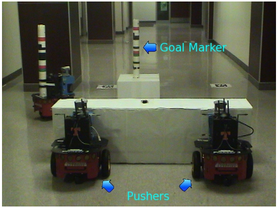
Table: MinIIS AND INDEPENDENCE OF INFORMATION INSTANCE

| Op. | **Go-to-goal** | # PoSs | After rem. | Use Ind. |
|---|---|---|---|---|
| 1 | $F_G(local), F_G(goal)$ | 18 | 10 | 7 |
| 2 | $F_R(goal, local)$ | 31 | 15 | 15 |
| Op. | **Push-box** | | | |
| 1 | $F_G(local), F_G(box), F_G(goal)$ | 36 | 20 | 9 |
| 2 | $F_R(box, local), F_R(goal, local)$ | 961 | 185 | 30 |
| | Time for a full search (s) | 15.1 | 2.9 | 0.02 |
| | Time for removal from 961 (s) | | 14.8 | 0.02 |

# Towards more flexibility

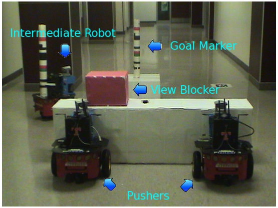
1. Pushers can localize and know the global goal




2. Pushers can localize and see the goal marker

3. Pushers cannot localize




4. Goal blocked and int. robot can see the goal and localize

5. Int. robot cannot localize

# Towards more flexibility

More flexibility is achieved using the MinIIS

Table: Select one Vs. Select approx. MinIIS

| Configurations | Select one | Select approx. MinIIS |
|---|---|---|
| 1. Localize, Global Goal | All retrievable | All retrievable |
| 2. Localize | All retrievable | All retrievable |
| 3. No Localize | No $F_G(goal\|local)$ | All retrievable |
| 4. Blocked, Int. Localize | All retrievable | All retrievable |
| 5. Blocked, Int. No Localize | No $F_G(goal\|local)$ | All retrievable |

# Contributions

- *Introducing the independence of information instances*

  Reduces from an exponential to a linear search space for practical applications

- *Identifying and removing unnecessary potential solutions in the solution space*

  Improves the search performance further

- *Relating behaviors with information requirements and providing a method to utilize different MSs*

  Achieves more flexibility during execution

# References

- Gerkey, B. and Mataric, M. (2001).
  Sold!: Auction methods for multi-robot coordination.
  *IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems*.
- 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. (2010).
  IQ-ASyMTRe: Synthesizing coalition formation and execution for tightly-coupled multirobot tasks.
  In *IEEE/RSJ Int'l Conference on Intelligent Robots and Systems*.