A Decentralized Approach to Cooperative Situation Assessment in Multi-Robot Systems

Giuseppe P. Settembre and Paul Scerri and Alessandro Farinelli and Katia Sycara and Daniele Nardi

Presented by: Sotirios Ch. Diamantas
05 April, 2013

CMANTIC Lab
Department of Computer Science
University of Nebraska, Omaha
Problem Studied

- To act effectively under uncertainty, multi-robot teams need to accurately estimate the state of the environment.
- Although individual robots, with uncertain sensors, may not be able to accurately determine the current situation, the team as a whole should have the capability to perform *situation assessment*.
- However, sharing all information with all other team mates is not scalable nor is centralization of all information possible.
- This paper presents a decentralized approach to cooperative situation assessment that balances use of communication bandwidth with the need for good situation assessment.
Problem Studied

• When a robot believes locally that a particular plan should be executed, it sends a proposal for that plan, to one of its team mates.
• The robot receiving the plan proposal, can either agree with the plan and forward it on, or it can provide sensor information to suggest that an alternative plan might have higher expected utility.
• Once sufficient robots agree with the proposal, the plan is initiated.
Problem Studied

- To coordinate their activities in complex unstructured scenarios, robots need the ability to aggregate information into estimates of *features* relevant to their mission and their interactions.
- The process of acquiring this knowledge is known as *situation assessment*.
- Situations are organized into a hierarchy with the most general situations at the top of the hierarchy and the most specific situations at the leaves.
- It is assumed that more specific plans are more effective, but it is better to execute a more general plan than a more specific one for the wrong situation.
Problem Studied

- For example, in the rescue domain the presence of a victim the most general situation, with an unconscious, badly injured victim and a conscious, uninjured victim being more specific situations.
- Based on belief in the current situation and a model of the reward for different plans in different situations an individual can compute the expected utility (EU) of each plan.
Notation

- $R = \{r_1, \ldots, r_m\}$ is the set of mobile robots
- $E = \{e^t_1, \ldots, e^t_n\}$ is a set of events which occur in the environment and can be imperfectly observed by robots
- Observations are not direct sensor readings (e.g., laser scans) but the result of a feature extraction process indicating the presence of an interesting feature in a given location of the environment
- By integrating observations over time, robots can have a belief over events $Bel_r(e^t_i)$ which represents the probability that the event $e^t_i$ is true
- A *situation* represents the fact that a certain combination of events are true or false at the same time in a certain location
Example

• For example, the situation of an unconscious victim would be defined by events \textit{victim present, localized,} $35 < \text{heat source} < 42$, \textit{O}_2 \textit{present and no movement}

• If events are independent, then the belief in a situation is simply the product of the belief of the constituent events

\[ S = \{ s^t_1, \ldots, s^t_m \} \quad S_{CL} = \{ S_{\perp}, S_0, \ldots, S_k \} \]
Constraints

- The first constraint requires that higher reward will be received when more specific plans are executed in appropriate situations.
- The second constraint says that it is better to execute more general plans than inappropriate but more specific plans.
- The final constraint simply requires that there is no reward or cost for not acting when the situation does not require action.

\[ U : \mathcal{P} \times S_{CL} \rightarrow \mathcal{R} \]

- \( s \in S_i \) and \( s \in S_j \) and \( S_i \subseteq S_j \) \( \rightarrow U(P_i, S_i) \geq U(P_j, S_i) \)
- \( s \notin S_i \) and \( s \in S_j \) and \( S_i \subseteq S_j \) \( \rightarrow U(P_i, S_j) < U(P_j, S_j) \)
- \( U(P_\bot, S_\bot) = 0 \)
Maximization of Reward

To perform the maximization specified in eqn. 1 (below), robots should execute at each time step $t$, for each situation instance $s$, the plan $P_k$ such that

$$
\max_k \sum_t U(P_k^t, S_i^t)
$$

$$
k^* = \arg \max_k \left( \sum_{S_x \in S_{CL}} Bel(S_x) \ast U(P_k, S_x) \right)
$$
Example

- $e_1$, representing presence of a human shape, and $e_2$, representing human like movement. $S_1 = \text{victim is defined by } e_1 \lor (e_2 \land \neg e_2)$ and $S_2 = \text{unconscious\_victim is defined by } e_1 \land \neg e_2$
- $P_1$ for $S_1$ where the plan involves sending in a robot to try to lead the victim to safety, sending human rescuers only if this fails. A plan $P_2$ involves immediately sending in human rescuers and is best suited for $S_2$ where the victim is unconscious.
- The rewards are: $U(P_1, S_1) = 2$, $U(P_1, S_2) = -1$, $U(P_2, S_1) = -2$ and $U(P_2, S_2) = 4$
- It is better to execute the more specific plan in the more specific situation (i.e. $U(P_2, S_2) > U(P_1, S_1)$) and it is better to act more general than wrong (i.e. $U(P_2, S_1) > U(P_1, S_1)$)
Algorithm 1: Algorithm executed by each robot

ONMSGRECEIVED(msg)
(1) INTEGRATEBELIFS(msg.obs)
(2) planAgree ← EVALARGUMENTS(msg.plan)
(3) if msg.status == PROPOSAL
(4) if planAgree
(5) msg.\$agree ← msg.\$agree + 1
(6) if \$agree < TTL
(7) SEND(msg,nextAgent())
(8) else
(9) INSTANTIATEPLAN(msg.plan)
(10) else
(11) msg.status ← CHALLENGE
(12) msg.obs ← RETRIEVERFUTURINGOBS(msg.plan)
(13) SEND(msg,msg.sender)
(14) else
(15) /* msg.status == CHALLENGE */
(16) if planAgree
(17) msg.status ← PROPOSAL
(18) msg.obs ← RETRIEVESUPPORTINGOBS(msg.plan)
(19) SEND(msg,origMsg.nextAgent())
(20) else
(21) if origMsg.prevAgent() ≠ null
(22) msg.obs ← RETRIEVERFUTURINGOBS(msg.plan)
(23) SEND(msg,origMsg.prevAgent())
(24) else
(25) DESTROY(msg)
Algorithm
Results

Figure 4: Performance comparison varying world size

Figure 5: Performance comparison varying world dynamics
Results

Figure 6: Performance comparison varying quality of perceptions

Figure 7: Performance comparison varying hierarchy depth
Results

Figure 8: Performance comparison for different TTL

Figure 9: Communication comparison for different TTL (number of messages)
Results

Figure 10: Number of messages required for each execution of the protocol
Conclusions and Future Work

• The approach presented in this paper performed almost as well as the centralized approach while using an order of magnitude less communication. It far out-performed the individual approach.

• An immediate point of interest is whether TTL can be dynamically adjusted to account for the amount of agreement or disagreement between agents.

• Another area of interest is whether plan deconfliction algorithms can be combined with this algorithm, potentially simplifying overall coordination and improving efficiency in one step.
Questions

Thank You!