On Deconflicting Local Coordination Among Agents (Extended Abstract)

Manh Tung Pham and Kiam Tian Seow
School of Computer Engineering, Nanyang Technological University, Singapore 639798
pham0028,asktseow@ntu.edu.sg

ABSTRACT

For conflict resolution between local coordination modules of distributed agents, synthesized based on inter-agent constraints, two original ideas are proposed. The first is a designer-comprehensible Distributed Constraint Specification Network (DCSN) for describing the constraint relationships among agents. The second is an algorithm using cut-set theory for generating, from a given DCSN, an AND/OR graph compactly representing all conflict resolution plans. A case study is presented.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents, Multiagent Systems

General Terms

Algorithms, Design, Theory

Keywords

Multiagent Coordination, Conflict Resolution

1. DCSN

We shall use small letters such as n, m, k, to denote integers, and for an integer $n \geq 1$, I_n denotes the set $\{1, 2, ..., n\}$.

DEFINITION 1. *Let $n \geq 2$, $m \geq 1$. A distributed constraint specification network (DCSN) $\mathcal{N}$ is a tuple $(\mathcal{A}, \mathcal{C})$, where $\mathcal{A} = \{A_i \mid i \in I_n\}$ is an agent set of size n and $\mathcal{C} = \{C^k_{J_k} \mid k \in I_m, J_k \subseteq I_n\}$ is an inter-agent constraint set of size m, where $C^k_{J_k} \in \mathcal{C}$ is a constraint specified for the group of agents $\mathcal{A}_{J_k} = \{A_i \mid i \in J_k\}$. In other words, for all $k \in I_m$, the agents in $\mathcal{A}_{J_k}$ must coordinate among themselves to respect $C^k_{J_k}$.*

Each $C^k_{J_k} \in \mathcal{C}$ in a DCSN $\mathcal{N}$ is said to be a relevant constraint for agents in the group $\mathcal{A}_{J_k} = \{A_i \mid i \in J_k\}$. Without loss of generality, assume henceforth that $\bigcup_{k \in I_m} J_k = I_n$, i.e., every agent in $\mathcal{A}$ is in $\mathcal{A}_{J_k}$ for some k, and so every agent needs to coordinate. Then a DCSN can be redefined as $\mathcal{N} = \{(J_k, C^k_{J_k}) \mid k \in I_m, J_k \subseteq I_n\}$. An element $\mathcal{N}^k_1 = (J_k, C^k_{J_k})$ of $\mathcal{N}$ is then called a basic subnet of $\mathcal{N}$; and a non-empty $\mathcal{N}^{\mathcal{S}_r}_r \subseteq \mathcal{N}$ consisting of $r = |\mathcal{S}_r| \geq 1$ basic subnets is called a r-constraint subnet of $\mathcal{N}$ with constraint subset $\{C^k_{J_k} \mid k \in \mathcal{S}_r\}$. Where the constraint subset is arbitrary, a r-constraint subnet is simply denoted by $\mathcal{N}_r$.

A DCSN can be graphically represented by an undirected graph with agents represented by rectangular nodes, and each constraint relevant for an agent group by an oval hyperedge with arcs connecting it to all the agents in the group. A r-constraint subnet $\mathcal{N}_r$ of $\mathcal{N}$ is said to be constraint-connected if the graph representing $\mathcal{N}_r$ is a connected graph.

Given a DCSN $\mathcal{N} = (\mathcal{A}, \mathcal{C})$, the problem of interest is to synthesize local plans and coordination strategies, called coordination modules (CM's) henceforth, for every agent in $\mathcal{A}$ to respect every constraint in $\mathcal{C}$. For this problem, we propose a compositional synthesis approach:

- **Step 1 Basic Subnet Synthesis:** Synthesize for every agent a set of local CM's, one for each of the agent's relevant constraints.
- **Step 2 Subnet Composition**
  -Step 2.1 Conflict Resolution Plan Generation: Generate a conflict resolution plan for the given DCSN.
  -Step 2.2 Conflict Resolution Plan Execution: Compose subnets with conflict resolution by following a precedence order of subnet composition operations in the plan. This is to completely deconflict the local CM's synthesized in Step 1. Each subnet composition operation entails designing deconflicting CM's for the agents concerned to ensure nonconflictingness in the composed subnet.

Henceforth, we present the theory for representing conflict resolution plans using AND/OR graphs [1]. We assume that a subnet composition is an operation on two subnets.

2. PLAN REPRESENTATION

A conflict resolution plan for a DCSN is a finite number of subnet composition operations, with ordering constraints between them. Such a plan may encompass several complete planning sequences, each of which is an ordered sequence of the subnet composition operations that satisfies all the ordering constraints. Executing a given plan means following one of its complete planning sequences to successively compose (the solutions of) different pairs of subnets to form (solutions of) larger subnets, starting with all basic subnets "disconnected" from each other, and ending with all of them correctly composed to form the DCSN.

Observe that a conflict resolution planning sequence for a DCSN $\mathcal{N}$ is a reversal of a successive decomposition, starting with $\mathcal{N}$, of constraint-connected component subnets until only basic subnets remain. This suggests that the forward search problem of generating conflict resolution plans $\mathcal{N}$ can be addressed as a backward search problem of successively decomposing $\mathcal{N}$ into pairs of constraint-connected component subnets until only basic subnets are left.

DEFINITION 2. *The AND/OR graph of conflict resolution plans*

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

for a DCSN $\mathcal{N}$ is a hyper-graph $T_{\mathcal{N}} = (S_{\mathcal{N}}, H_{\mathcal{N}})$, where 1) $S_{\mathcal{N}}$ is the set of nodes of $T_{\mathcal{N}}$ and defined as $S_{\mathcal{N}} = \{\mathcal{N}_r \subseteq \mathcal{N} \mid \mathcal{N}_r$ is constraint-connected $\}$, and 2) $H_{\mathcal{N}}$ is the set of hyper-edges of $T_{\mathcal{N}}$ and defined as $H_{\mathcal{N}} = \{(\mathcal{N}_{r_1}, (\mathcal{N}_{r_2}, \mathcal{N}_{r_3})) \in S_{\mathcal{N}} \times (S_{\mathcal{N}} \times S_{\mathcal{N}}) \mid \mathcal{N}_{r_2} \cap \mathcal{N}_{r_3} \neq \emptyset$ and $\mathcal{N}_{r_1} = \mathcal{N}_{r_2} \cup \mathcal{N}_{r_3}\}$.

The nodes in the AND/OR graph $T_{\mathcal{N}}$ represent constraint-connected subnets of $\mathcal{N}$, and each of the hyper-edges is a pair $(\mathcal{N}_{r_1}, (\mathcal{N}_{r_2}, \mathcal{N}_{r_3}))$ denoting the decomposition of subnet $\mathcal{N}_{r_1}$ into two component subnets $\mathcal{N}_{r_2}$ and $\mathcal{N}_{r_3}$, or equivalently, the composition of $\mathcal{N}_{r_2}$ and $\mathcal{N}_{r_3}$ into $\mathcal{N}_{r_1}$. A hyper-edge points from a node representing a subnet to two nodes representing the component subnets. The node that represents the complete DCSN $\mathcal{N}$ is referred to as the root node and denoted by n_{root}, and the nodes representing basic subnets of $\mathcal{N}$ are referred to as the leaf nodes. The set of all leaf nodes of $T_{\mathcal{N}}$ is denoted by Θ_{leaf}. In what follows, a conflict resolution plan for $\mathcal{N}$ is represented by a tree in $T_{\mathcal{N}}$ that starts at n_{root} and terminates at Θ_{leaf}.

3. AND/OR GRAPH PLAN GENERATION

To generate an AND/OR graph representation of conflict resolution plans, the idea is to enumerate all possible decompositions of a DCSN $\mathcal{N}$ into two constraint-connected component subnets. Each such decomposition corresponds to an edge of the AND/OR graph $T_{\mathcal{N}}$ connecting the root node representing $\mathcal{N}$ to two nodes, with each representing a component subnet. The same decomposition process is then repeated for each of the component subnets until only basic subnets are left.

DEFINITION 3. *The constraint relational network (CRN) $\mathcal{CRN}_r$ of a r-constraint subnet $\mathcal{N}_r^{\mathcal{S}_r} = \{(J_k, C_{J_k}^k) \mid k \in \mathcal{S}_r\}$ is a tuple $(\mathcal{C}_r, \mathcal{R}_r)$, where $\mathcal{C}_r = \{C_{J_k}^k \mid k \in \mathcal{S}_r\}$ is the constraint set of size r in $\mathcal{N}_r$ and $\mathcal{R}_r \subseteq \mathcal{C}_r \times \mathcal{C}_r$ is a relation over $\mathcal{C}_r$, such that $(\forall C_{J_k}^k, C_{J_h}^h \in \mathcal{C}_r)[(C_{J_k}^k, C_{J_h}^h) \in \mathcal{R}_r \Leftrightarrow (J_k \cap J_h \neq \emptyset)]$.*

Observe that enumerating all possible decompositions of a subnet $\mathcal{N}_r$ into two constraint-connected subnets can be done by enumerating all possible cut-sets of its CRN $\mathcal{CRN}_r$. Specifically, consider a cut-set $(\mathcal{C}_x, \mathcal{C}_y)$ that decomposes $\mathcal{CRN}_r$ into two parts, where $\mathcal{C}_x$ and $\mathcal{C}_y$ are the two disjoint sets of vertices of $\mathcal{CRN}_r$ belonging to these two parts. Write $\mathcal{N}_x \sim \mathcal{C}_x$ and $\mathcal{N}_y \sim \mathcal{C}_y$ to denote respectively that $\mathcal{N}_x$ and $\mathcal{N}_y$ are the component subnets induced by $\mathcal{C}_x$ and $\mathcal{C}_y$. Then $\mathcal{N}_x$ and $\mathcal{N}_y$ are two constraint-connected component subnets decomposed from $\mathcal{N}_r$. Conversely, any decomposition of $\mathcal{N}_r$ into two constraint-connected component subnets $\mathcal{N}_x$ and $\mathcal{N}_y$ corresponds to a cut-set $(\mathcal{C}_x, \mathcal{C}_y)$ of $\mathcal{CRN}_r$, with $\mathcal{N}_x \sim \mathcal{C}_x$ and $\mathcal{N}_y \sim \mathcal{C}_y$.

Procedure *GenerateANDORGraph*($\mathcal{N}$)

Output: An AND/OR graph $T_{\mathcal{N}} = (S_{\mathcal{N}}, H_{\mathcal{N}})$ of conflict resolution plans for $\mathcal{N}$, initialized with $S_{\mathcal{N}} = \emptyset$ and $H_{\mathcal{N}} = \emptyset$

begin

Step 1: If $\mathcal{N}$ contains only one basic subnet then return; otherwise, convert $\mathcal{N}$ into a $\mathcal{CRN} = (\mathcal{C}, \mathcal{R})$;

Step 2: Compute $CutSets$ as the set of all cut-sets of $\mathcal{CRN}$;

Step 3 while $CutSets \neq \emptyset$ **do**

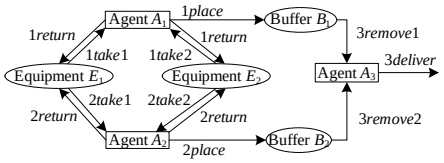
Step 3a Remove a cut-set $(\mathcal{C}_x, \mathcal{C}_y)$ from $CutSets$. Let $\mathcal{N}_x \sim \mathcal{C}_x$ and $\mathcal{N}_y \sim \mathcal{C}_y$;

Step 3b Add nodes and an edge to T: $S_{\mathcal{N}} = S_{\mathcal{N}} \cup \{\mathcal{N}_x, \mathcal{N}_y, \mathcal{N}_x \cup \mathcal{N}_y\}$, $H_{\mathcal{N}} \cup \{(\mathcal{N}_x \cup \mathcal{N}_y, \mathcal{N}_x, \mathcal{N}_y)\}$;

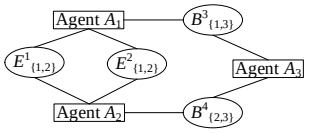
Step 3c For $r \in \{x, y\}$, $GenerateANDORGraph(\mathcal{N}_r)$;

4. A CASE STUDY

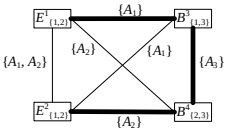
The system under study [Fig. 1(a)] consists of three agents A_1, A_2 and A_3, and four constraints $E_{\{1,2\}}^1$, $E_{\{1,2\}}^2$, $B_{\{1,3\}}^3$ and $B_{\{2,3\}}^4$. A_1 and A_2 are producer agents that continually follow a production plan: Acquire manufacturing equipment E_1 and E_2 in either order, produce a workpiece, return the equipment to their initial location, move to the buffers' location, place the finished workpiece into the respective one-slot buffer B_1 and B_2, and finally return to the initial state for a new production cycle. A_3 is a delivery agent that continually takes a work piece from either buffer B_1 or B_2, processes, and delivers it to customers. The four constraints $E_{\{1,2\}}^1$, $E_{\{1,2\}}^2$, $B_{\{1,3\}}^3$ and $B_{\{2,3\}}^4$ are formulated to respectively ensure mutual exclusion of equipment use, and no overflow or underflow of buffers.



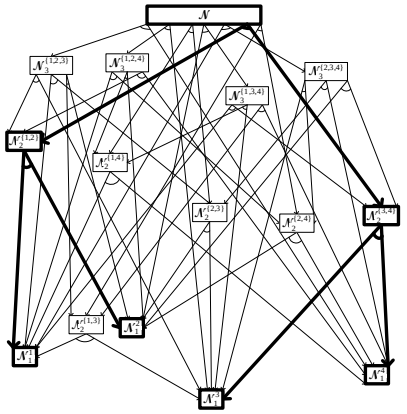
(a) Overall system



(b) DCSN



(c) CRN



(d) AND/OR graph plans

Figure 1: A manufacturing system.

Fig. 1 shows the DCSN [Fig. 1(b)], CRN [Fig. 1(c)] and AND/OR graph representation of conflict resolution plans [Fig. 1(d)] for the manufacturing system. This AND/OR graph can be automatically generated by applying the procedure *GenerateANDORGraph* to decompose the CRN in Fig. 1(c) recursively.

With appropriate weights assigned to hyper-edges of an AND/OR graph, an A^* search [2] can return a solution tree with minimum depth, namely, a plan that allows maximal simultaneity in executing deconflicting operations, such as the tree highlighted in Fig. 1(d). Presented elsewhere [3], a complete CM solution for this case study can be efficiently computed by executing this highlighted plan tree.

5. REFERENCES

[1] N.Deo. *Graph Theory with Applications to Engineering and Computer Science*. New York: Prentice-Hall, 1974.

[2] S.Russell and P.Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[3] M. T. Pham. *Discrete-event Multiagent Coordination: Framework and Algorithms*. PhD Thesis, NTU, 2011.