Partial Adjustable Autonomy in Multi-Agent Environment and Its Application to Military Logistics

Hoong Chuin LAU
School of Information Systems
Singapore Management University
hclau@smu.edu.sg

Lucas AGUSSURJA
School of Computing
National University of Singapore
lucasagu@comp.nus.edu.sg

Ramesh THANGARAJOO
The Logistics Institute Asia Pacific
National University of Singapore
tlitr@nus.edu.sg

Abstract

In a rapidly changing environment, the behavior and decision-making power of agents may have to be adaptive with respect to a fluctuating autonomy. In this paper, a centralized fuzzy approach is proposed to sense changes in environmental conditions and translate them to changes in agent autonomy. A distributed coalition formation scheme is then applied to allow agents in the new autonomy to renegotiate to establish schedule consistency. The proposed framework is applied to a real-time logistics control of a military hazardous material storage facility under peace-to-war transition.

1. Introduction

The dynamism in complex systems may require behavioral changes in agents in terms of their local goals and the manner they collaborate and negotiate with other agents. It may also require agents to transfer their decision-making power either completely or partially to other agents. The extent to which changes or transfers occur is often referred to as the *adjustable autonomy* of the agent. Often, adjustments occur in stages as situations on the ground continually change. The challenge is in devising mechanisms that continuously translate changes on the ground to the agents' adjustable autonomy.

In this paper, we propose a centralized approach to dynamically adjust the local objectives and autonomy levels of the agents. Here, a fuzzy framework is designed which translates the environmental changes to changes in the agents' local objectives and autonomy. Given the existing objectives and autonomy levels, a distributed coalition formation based algorithm is then applied where agents negotiate to obtain a consistent solution with respect to a set of global constraints and objective.

2. Literature Review

Various reasons entail changes of agent autonomy over time. In some instances, the agent might lack the capability to make decisions [5]. Others tend to make provisions for human intervention to cater for unexpected events where the agent may act irrationally [4]. Reduced confidence in a fully automated system is a common reason where some crucial decisions require user interference. [7] studies the possibility of transferring autonomy for higher quality decisions with minimizing coordination problems. While agent-to-agent transfer of autonomy is considered, there is little work on the needs for such a transfer and the practical advantages and implications of such a strategy. Most of the mentioned work have focused on a single agent and a single user. While [7] has expanded beyond the one-to-one relationships, the author limits the transfer to a complete transfer of autonomy and does not consider situations where some autonomy may be retained. [1] proposes three discrete levels of autonomy: ”Command-Driven” where the agent does not make decisions and must obey orders, ”Consensus” where the agent shares decision making equally with other agents and ”Locally Autonomous” where the agent makes decisions alone. In this paper, we consider a real-life military situation where there a need to vary the autonomy gradually and continuously to suit changing circumstances.

Constraint Satisfaction Problem (CSP), Dynamic CSP, Distributed CSP and other variants are well-studied problems. For Dynamic CSP, constraints can change over time. [3] developed methods for adapting consistency computations to such changes, and [2] solved this problem by seeking minimal change on the solution. [6] extended this problem to Distributed Constraint Optimization Problem as well and proposed the ADOPT algorithm which guaranteed a user-defined optimality. [8] also provide an algorithm (AptOPT) to solve distributed Constraint Optimization Problem, which experimentally performs better than ADOPT.

3. Problem Definition

Essentially, the problem we study can be divided into two parts, i.e. sense and response. The sense problem is how to correctly translate all sensing data gathered by the agents to new local objectives and autonomy levels for each agent. The response problem is, given the new objectives and the autonomy levels of the agents, how the agents should interact with each other to respond to the environment.

Let N be the index set of n agents, and each agent $i \in N$ holds a set of mutually exclusive variables $A_i = \{a_{i1}, a_{i2}, ..., a_{il_i}\}$. The domains for the variables held by agent i are $D_i = \{D_{i1}, D_{i2}, ..., D_{il_i}\}$. The values that can be taken by the variables $a_{11}, a_{12}, ..., a_{21}, a_{22}, ..., a_{nl_n}$ are restricted by a set of global constraints $\mathbf{C} = \{C_1, C_2, ..., C_q\}$. Each agent i has a set of *objective functions* $F_i = \{f_i^1, f_i^2, ..., f_i^{p_i}\}$. Each function is defined as $f_i^k : \prod_{1 \leq j \leq l_i} D_{ij} \rightarrow \mathbf{R}$.

These agents exist in a dynamic environment that are captured by the dynamic environment variables. Let $\mathbf{E}$ be the set of m *environment variables* $\mathbf{E} = \{e_1, e_2, .., e_m\}$ that can be sensed by the agents. The domain for variable e_i is the set D_{e_i} of real numbers. Each agent senses some of the environment variables. Let the function $sensed : N \rightarrow 2^{\mathbf{E}}$ returns the set of variables that an agent can sense, equivalently $sensed(i) \subseteq \mathbf{E}$. This is to capture the idea that each agent has its own sensor, and able to sense its surrounding.

The sense problem is defined as, given the existing values of the environment variables in $\mathbf{E}$, find

1. the objective that the agent must focus on, i.e. a mapping for each agent i, $\mathbf{M}_i : \prod_{1 \leq j \leq m} D_{e_j} \rightarrow F_i$;
2. the relative autonomy levels of the agents, i.e. a mapping for each agent i, $\mathbf{L}_i : \prod_{1 \leq j \leq m} D_{e_j} \rightarrow [0, 1]$.

The response problem is defined as, given both the objective f_i and autonomy level al_i set for each agent i, find an assignment for the agent variables $(a'_{11}, a'_{12}, ..., a'_{21}, a'_{22}, ..., a'_{nl_n})$ which is consistent with $\mathbf{C}$ and minimizes the agent local objectives and the global objective which is defined in this work as the sum of the relative "errors" of all agents, i.e. minimizing

$$\sum_{i \in N} \left(al_i * \frac{f_i^* - f_i(a'_{i1}, a'_{i2}, ..., a'_{il_i})}{f_i^*} \right)$$

where ${f_i}^*$ is the value of the objective given the optimal assignment to the variables of agent i.

4. Solution Approach

The overall solution approach is given as follows. For the sense problem, we propose a fuzzy framework to capture the relationships between environment variables and the agent objectives and autonomy levels. The role of the fuzzy framework is to continually capture the sensor data from all the agents and in turn translate these data into new objectives and autonomy level of the agents. Then, based on the objectives and the autonomy levels set, we apply a distributed branch-and-bound algorithm to obtain the solution for the response problem.

4.1. Fuzzy Framework

The system behavior of a sense-and-response system is often difficult to design. Due to the adaptive nature required of the system, the interpretation of the inputs to the system such that a proper reaction can be designed is a complex task. In this paper, we propose a fuzzy selection process to determine the autonomy and objective of agents. Fuzzy system has been studied extensively in the literature, and has been applied to a wide range of real life applications. The advantages of using a fuzzy system are that it generally captures relationships between non-discrete data and handles incomplete data effectively.

The five-stage fuzzy process listed below has been extensively reported [4][8]. In the following, we define the function $\mathbf{L}_i(e'_1, e'_2, ..., e'_m)$ as consisting of:

- **Normalization**, performs a scale transformation of the input metric into a normalized variable. For each input variable $e_j \in \mathbf{E}$, its domain $D_{e_j} = [0, r_j]$ is normalized into the interval $[0, a]$ for some value a, and the normalization function is:

$$\mathbf{n}_j(e'_j) = \left(\frac{a}{r_j} \right) .e'_j$$

- **Fuzzification**, determines the degree to which the normalized variables belong to each of the appropriate fuzzy sets via membership functions. The domain $[0, a]$ of each of the normalized variable e_j can be divided into several states, called fuzzy sets: $I_j^1, I_j^2, ..., I_j^{s_j}$, let $\mathbf{I}_j = \{I_j^1, I_j^2, ..., I_j^{s_j}\}$. This applies to the output variable as well. The partial functions $\mu_{I_j^k}$ for $1 \leq k \leq s_j$ measure the degree of membership of the normalized value to the set I_j^k.
- **Fuzzy Inferencing**, a rule base, also known as the inference engine, links the fuzzy inputs to the fuzzy out-

puts via linked statements called rules. A rule is an if-then statement of the form:

if e_j is in I_j^k and e_p is in I_p^q and ...
then al_i is in O_i^t.

for some p, q, t where al_i is the autonomy level of agent i, $O_i^t \in \mathbf{O}_i$, and $\mathbf{O}_i$ is the set of fuzzy sets for the output variable al_i. In other words, the if-part of a rule contains a set of fuzzy sets of the input variable, and the then-part of the rule contains a fuzzy set of the output variable. A rule will fire if $\mu_{I_j^k}(\mathbf{n}_j(e_j'))$ is defined for all I_j^k contained in the if-part of the rule. The result of firing a rule is the state (fuzzy set) that the output variable in and the degree of the membership which is $min\left\{\mu_{I_j^k}(\mathbf{n}_j(e_j'))\right\}$ for all I_j^k in the if-part of the rule.

- **Defuzzification**, the fuzzy sets for the output from different rules that fire are merged and the center of area of the combined fuzzy sets is found to get a defuzzified value called the centroid c_i.
- **Denormalization**, is the reverse of the normalisation process where the scaling biases are removed. The coefficients which are obtained for each agent, can be viewed as weights that have to be applied to determine the autonomy of the agent, i.e. $al_i = \left(\frac{1}{a}\right).c_i$

The value of a, the membership functions and the rules are defined based on empirical methods. $\mathbf{M}_i$ is defined in a similar way.

4.2. Coalition Formation Based Algorithm

Here, we propose a coalition formation based algorithm for the agents to find the assignment to all the variables $(a_{11}', a_{12}', ..., a_{21}', a_{22}', ..., a_{nl_n}')$ that is consistent with $\mathbf{C}$ while optimizing the measure of objective described above, or to output no solution if such solution is not found.

This algorithm is based on distributed branch-and-bound. The idea is to have the agents form coalitions, and to find local solutions. If these local solutions do not conflict each other with respect to $\mathbf{C}$, then the solution is found. If conflicts between these local solutions occur, then the agents will try to form a bigger coalitions, and then find the consistent solutions within this bigger coalitions. In the worst case, all the agents will form one big coalition, and if consistent solution still can not be found, the algorithm will output no solution.

The algorithm is divided into 3 parts, namely initialization, coalition formation, and negotiation.

During intialization, each agent first constructs an optimal assignment to its variables using a branch and bound search that maximizes its given objective function f_i. This optimal value is then stored in the variable f_i^*. G_i is the set that stores all the agents that are currently in coalition with agent i. The set is initialized to i itself. The variable L_i indicates the agent that has the role as the leader in the current coalition of agent i, this agent is the one with the highest autonomy level in the coalition. The set K_i stores all the agents having conflict with agent i, initially set to empty. Variable $state_i$ indicates the state of the agent i. After initialization, the agents will enter into coalition formation stage.

In coalition formation stage, each agent first checks for any external conflicts , which is done through exchange of variables. After detecting conflicting neighbors, each agent will send invitations to all its conflicting neighbors to form a coalition . Along with the invitation, the agent also sends the variable al_{L_i} which is the autonomy level of the leader of the current coalition, or the highest individual autonomy level in the set G_i. An agent will tend to join a coalition which has a member with the higher autonomy level. This is important for the convergence of the algorithm. The algorithm uses concurrent invite and accept to ensure each agent is in at most one coalition at any point of time. When the agents have formed the coalitions, they will enter the negotiating stage of the algorithm given as follows:

NEGOTIATE()
1 **if** (all agents are in $no\_conflict$ state)
2 **then** return $(a_{i1}, a_{i2}, ..., a_{il_i})$
3
4 $L_i \leftarrow j$ where $al_j \geq_{k \in G_i}^{max} \{al_k\}$
5 **if**($i \neq L_i$)
6 **then** SEND(($D_{i1}, D_{i2}, ..., D_{il_i}$),$f_i$,$f_i^*$) to L_i
7 **else**
8 $n \leftarrow \#G_i$
9 **when received**($(D_{j1}, D_{j2}, ..., D_{jl_j}) f_j, f_j^*$) **do**
10 $n \leftarrow n - 1$
11 **if**($n = 0$)
12 **then** do branch and bound search to find an assignment d_{pq} to all the variables $a_{pq} \in A_p$ for all $p \in G_i$ that minimizes $\sum_{p \in G_i}\left(al_p * \frac{f_p^* - f_p}{f_p^*}\right)$
13 **if** (no consistent solution found)
14 **then** return "NO SOLUTION"
15 SEND($d_{p1}, d_{p2}, ..., d_{pl_p}$) to all $p \in G_i$
16
17 **when received**($d_{i1}, d_{i2}, ..., d_{il_i}$) **do**
18 assign $(d_{i1}, d_{i2}, ..., d_{il_i})$ to $a_{i1}, a_{i2}, ..., a_{il_i}$
19
20 FORM-COALITION()

If all the agents are in $no\_conflict$ state, then a consistent solution has been reached, and each agent will return their current assignment to the variables as the solution (lines 1-2). If the agents are in negotiating state, then the agent with the highest autonomy level in a coalition will gather the data from all other agents in the coalition. A branch and bound search will then be performed to find a consistent solution that minimizing the sum of the relative error of each agent in the coalition. If a consistent solution cannot be reach, the algorithm returns no solution.

When a solution is reached, it is broadcasted to every member of the coalition, and the agents return to coalition formation stage.

5. Application in Military Storage Facility

In this section, we apply our solution framework to solve a real-time control problem within a military storage facility with a high space restriction.

In this problem a wide range of hazardous products are to be stored in containers. A job comprises a combination of products of given quantities retrieved from a given set of containers. One or more vehicles will be used to transport the requested products out of the facility. Demands are dynamic since their arrival timings are not known apriori. The paths leading to the entrance are defined by a road network and there is a restriction on the number of vehicles that can travel on a particular road segment at any time point. Road segments are separated by doors which should be kept closed for safety of the facility, and opened only to allow vehicle moments. The number of door opening and closing should also be minimized because they generate heat within the facility. The door temperature increases each time the door is opened or closed, and decreases gradually over time in between. There is also a safety consideration that discourages doors from opening concurrently (to contain possible leakages and explosions).

5.1. Experimental Setup

We perform experiments on a prototype of the storage facility problem. In this paper, we report results for a 7-agent problem responsible for the 7 doors (not including the main door). Each agent has two objectives, namely to minimize request tardiness (by minimizing the waiting time of the vehicles), and to minimize the heat caused by opening/closing. The constraints are: the exposure of the facility due to the opening of the doors must not exceed certain level. The exposure is measured by the number of road segments that are revealed because of the opening of the doors. In this case, it must not be more than 4. There are 3 traffic agents with 1 overlapping road segment. The congestion level of this segment is limited to 15 vehicles at any point of time. The traffic agents have a single objective, that is to minimize the delay of jobs. The outline of the system is given in Figure 1.

The environmental variables captured in the experiment are the war level and the temperatures of each door. These two inputs are fed into the fuzzy system. The domain of the temperature $[25, 65]$ is normalized to $[0, 1]$ and divided into five fuzzy sets {L,LM,M,MH,H} as shown in Figure 2. Similarly, the domain of the input war-level $[0, 100]$ is normalized and divided into three sets {L,M,H}. The output variables of the fuzzy system are: autonomy level, with domain $[0, 1]$, and objective function, with domain $\{0, 1\}$. The domain of the autonomy level is also normalized and divided into five sets {L,LM,M,MH,H}. The rule base inside the fuzzy system are obtained from empirical and historical data.

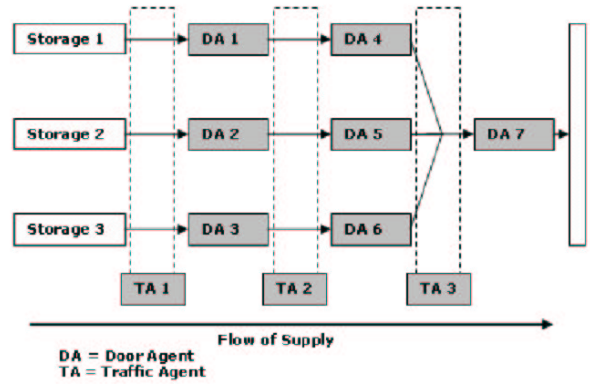

Figure 1. Storage Facility

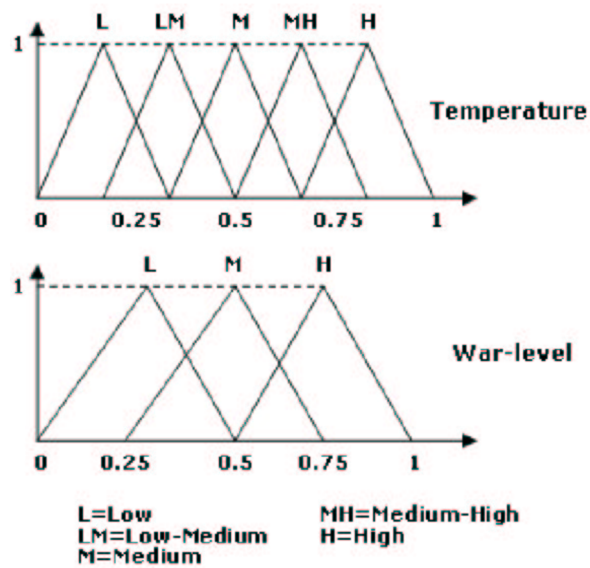

Figure 2. Fuzzification of Input Variables

A subset of the rules for determining the autonomy levael and objective function of the door agents is shown in Figure 3, where a low objective function refers to minimizing tardiness and high refers to minimizing temperature.

For each pair of environmental variables, we generate 3000 requests with different deadlines. The two output variables that we observed are: the tardiness of the requests (how late the requests are satisfied) and the safety level of the system, by measuring the exposure caused by the opening of the doors.

Input		Output	
Temperature	War-level	Autonomy Level	Objective Function
Low	Low	Low	High
Low	Medium	Low-Medium	Low
Low	High	Medium	Low
Low-Medium	Low	Low	High
Low-Medium	Medium	Low	Low
Low-Medium	High	Medium	Low
Medium	Low	Medium	High
Medium	Medium	Medium	Low
Medium	High	Medium	Low
Medium-High	Low	Medium-High	High
Medium-High	Medium	Medium-High	High

Figure 3. Subset of The Inferencing Rules

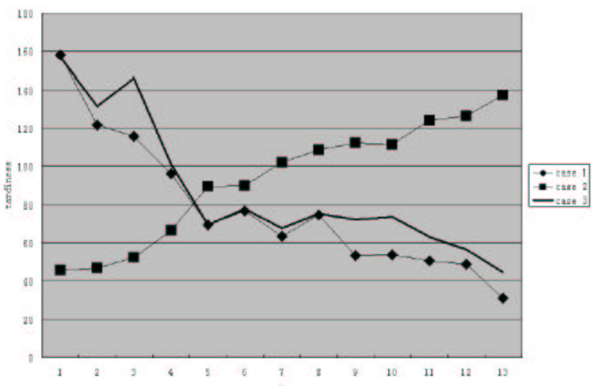

Figure 4. Comparison of Tardiness

5.2. Experimental Results

The test cases are divided into 3 parts. First, we fixed the average temperature to 25^{o}C, and varied the war level. Second, we fixed the war level, to 15% and varied the average temperature. Thirdly, we varied both the war level and the average temperature.

Figure 4 shows the measure of tardiness for three different cases. The first case shows that as the war level increases over time, tardiness decreases but at the same time the security level of the whole system also decreases, as seen in Figure 5. In the second case, as we increase the temperature over time, the effect is the opposite of the first case, as seen in both figures. For the third case, the variations in the observed output variables are shown in Figures 4 and 5. It can be observed that the system adapts to changes in environmental variables and respond accordingly.

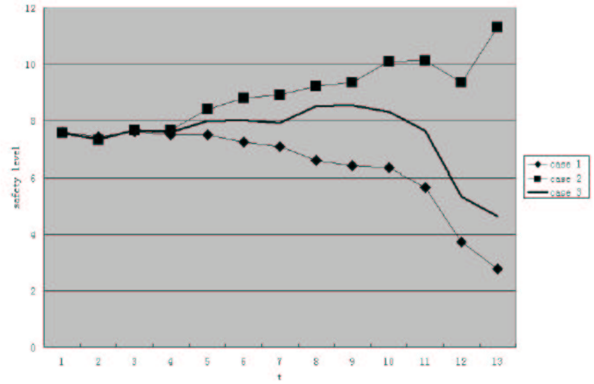

Figure 5. Comparison of Safety

6. Conclusion

We studiued the problem of partial adjustable autonomy in the context of sense and respond to dynamic environmental changes via changing automony levels and objectives. Our solution approach has shown promising results for a real problem involving the design of a control system for a military storage facility under a peace-to-war transition. We believe the proposed approach can be adapted to solve other real-time logistics management problems where drastic changes in environmental variables (such as peace-to-war) is an issue.

References

[1] K. Barber, A. Goel, and C. Martin. Dynamic adaptive autonomy in multi-agent systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 12 (2):129–148.

[2] C. Bessiere. Arc-consistency in dynamic constraint satisfaction problems. *Proc. AAAI*, pages 221–226, 1991.

[3] G. Dorais, R. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems on mars. *In Proceedings of the First International Conference of the Mars Society*, pages 397–420.

[4] D. Driankov, H. Hellendoorn, and M. Reinfrank. An introduction to fuzzy control, springer-verlag berlin, heidelberg. 1996.

[5] J. P. Gunderson. Adaptive goal prioritization by agents in dynamic environments. *IEEE Systems, Man, and Cybernetics*, pages 1944 – 1948, October 2000.

[6] R. Mailler and V. Lesser. Solving distributed constraint optimization problems using cooperative mediation. *Autonomous Agent and Multi Agent System*, 2004.

[7] Y. Ran, N. Roos, and J. van den Herik. Approaches to find a near-minimal change solution for dynamic csps. *Proc. CPAIOR*, pages 373–387, 2002.

[8] H. Wang, J. Rooda, and J.F.Haan. Solving problems with a fuzzy approach. *IEEE International Conference on Fuzzy Systems*, 1:194–198, 1996.