In IAS-7, 2002

Algorithms for Utility-based Role Exchange

Xin Zhang and Henry Hexmoor

Computer Science & Computer Engineering Department, Engineering Hall, Room 313,
Fayetteville, AR 72701
{xxz03, hexmoor}@uark.edu

Abstract. We present three algorithms for utility-based role exchange that are inspired by game theory. We introduce some methods for comparison of algorithms. While all three algorithms provide similar gains, the extent to which utilities of other agents are considered in role exchange, the number of exchanges is minimized.

1 Introduction

In multiagent systems, agent roles have been studied in regards to plans and distribution of tasks [2][3]. Little attention is paid to agent's preference over its roles, and reasoning about swapping roles with other agents. In this paper we will consider agents operating based on the social welfare of their group. As such role exchange between agents will be to benefit the collective. Therefore, some agents need to sacrifice their individual utility for the sake of their community [1].

Careful selection of a pattern of roles for adoption is a key point for improving the group utility. [12] presented the concept of *individual utility*, which measures each agent's utility in a specific role and use the *total utility*, which represents the team utility in the formation of roles. [12] also presented the concept of *role exchange values (REV)*, which measures each agent's utility gain during the hypothetical role exchange. We considered role exchange based on individual agent's utility gain and the algorithm allowed role exchange between any pair of agents who may experience net positive gain from the exchange. This *any role exchange algorithm* is not optimal for the consideration of the total numbers of role exchange times before reaching the maximum total utility.

In this paper we will show two other variants of the algorithm and compare their performance. At first we will introduce the related works in the field of formation-based roles in section 2. Then we will provide some assumptions, formula, and the conditions for role exchange. Based on the formula and conditions, we will introduce the *any role exchange algorithm* in section 4.1, *Individual Optimal Role Exchange Algorithm* in section 4.2, and *Group Optimal Role Exchange Algorithm* in section 4.3. The implementation and comparison of the three algorithms will be presented in section 5 with the results. In section 6, we will provide some concluding remarks.

2 Related Work

Formation as basis of collaboration among agents is introduced in [4][8]. Formations are commonly found in the game of soccer or Robocup [11]. Agents adopt an initial assignment of roles, but this assignment may need to be revised as the situation changes. Therefore, re-assignment of roles or some role exchanges become necessary [6]. Formations are dynamic and flexible, and it is a team structure to decompose the task into a set of roles. There is an initial formation, but there are also the run-time triggers for dynamic changes of the formations [5]. Formation-based role assignment ensures flexibility of role exchange, which was beneficial to the FC Portugal team [7].

Various formations as role models are discussed in [5]. A role defines a position and a set of responsibility within a role model, and roles are assigned to agents in an application. Role model can have various forms of dynamic behavior, which ensures the modeling mobility and adaptive behavior. Agent organizations can take on various formations, and role assignments can be assigned dynamically [5].

Consider a group of distinct *n* agents, $A = \{a_1, a_2, \ldots, a_n\}$, where a_i is the ith agent and a set of distinct n roles, $R = \{r_1, r_2, \ldots, r_n\}$, where r_i is the ith role, such that $i \neq j \Rightarrow r_i \neq r_j$. For any agent a_i and role r_j, there is a utiltiy $U(a_i, r_j) = u_{ij}$, where u_{ij} stands for the utility of adopting role j by agent i and function $U(a_i, r_j)$ is to get u_{ij}. Any formation F is a set of $\{< a_i, r_j, u_{ij} > \mid where\ i, j \in [1..n]$, for any pair of $< a_i, r_j, u_{ij} >$ and $< a_k, r_l, u_{kl} >$, $a_i \neq a_j \Rightarrow r_k \neq r_l\}$ and each agent is assigned a single distinct role[12].

3 The Assumptions and Properties for Role Exchange

Assume the agents we discuss in this paper obey Pareto-optimality. We make the following assumptions.

1. With N agent and N roles, there is a one to one assignment of roles to agents.
2. Each agent has a unique utility value per role. I.e., $V(A, R)$ is agent A's unique utility in role R.
3. An agent's adoption of a role will not affect the utility of another agent adopting another role.
4. The *total utility* of a number of agents is equal to the sum utilities from each of these agents. I.e. *Total Utility* = $\sum_{i=1}^{i=n} V(A_i, R_i)$
5. The role exchange process takes place only between a pair of roles at one time.
6. If the margin of gain from a hypothetical role exchange is positive for a pair of agents, they are obliged to exchange. This is due to the Pareto-optimality cooperative agents.

7. The time consumption or utility loss due to the process of role exchange is assumed negligible and will not be considered.

The concepts of *Role Exchange Value* (REV) and *Individual Utility Gain* (IUG) for each agent are computed when considering role exchange in an agent pair [12]. REV involves a pair of agents and a pair of roles, i.e., agent A, agent B and role R1 and role R2. We have the following concepts related with role exchange:

1. $V(A, R1)$ represents the unique utility of agent A taking role R1.
2. $V(A, R1, B, R2)$ represents the sum utility of agent A taking role R1 and agent B taking role R2. I.e., based on assumption 3 and 4, $V(A, R1, B, R2) = V(A, R1) + V(B, R2)$.
3. $REV_A(A, R1, B, R2)$ represents the role exchange value of agent A in the formation that agent A taking role R1 while agent B taking role R2.

In role exchange for agent pair (A, B), the REV of agent A in the formation that agent A taking role R2 while agent B taking R2 as equation (1):

$$REV_A(A, R1, B, R2) = 0.5 * \{V(A, R1) + V(A, R2, B, R1) - V(B, R2)\} \quad (1)$$

The *individual utility gain* for agent A in this role exchange formation is as equation (2):

$$IUG_A(A, R1, B, R2) = REV_A(A, R1, B, R2) - V(A, R1) \quad (2)$$

Based on equations (1) and (2), we use the following 3 conditions in Figure 1 to check if role exchange is applicable.

1. If $IUG_A(A, R1, B, R2) < 0$, role exchange will degrade to total utility for the entire group and original role formation is better.
2. If $IUG_A(A, R1, B, R2) = 0$, role exchange is not necessary. There is no difference between before and after role exchange.
3. If $IUG_A(A, R1, B, R2) > 0$, role exchange will be beneficial to the entire group.

Figure 1. Conditions for Role Exchange

4 The Algorithms for Role Exchange

The algorithms for utility-based role exchange take the following steps. In our algorithms, t is the time index. For instance, $t = 0$ is the time before any role exchange. $t = 1$ is the time at the 1st exchange. Function $add(< x, y, z >, S)$ adds the triple $< x, y, z >$ to set S. Function $delete(< x, y, z >, S)$ deletes the triple $< x, y, z >$ from set S. Function $IsMaxForAll(i, k, j, l)$ is true if $IUG_i(i, k, j, l)$ is the maximum among all the formations. Function $IsMaxFori(i, k, j, l)$ is true if $IUG_i(i, k, j, l)$ is the maximum among the formations, which correspond to role exchange with agent i . *stop* stands for termination of the algorithm. Predicate "formation" picks out a specific formation, e.g., F.

We discussed an algorithm for utility-based role exchange in [12]. In this paper we will present 2 other algorithms for role exchange and compare them to see what the advantage and disadvantage for each of them. The initial conditions are the following.

- Agent-role utility table has been set up and the utility value is randomly generated.
- Each agent adopts an initial role randomly. We assume that the *i*th agent adopts *i*th role prior to any role exchange.
- Pairs of agents are selected in order to explore applicability of role exchange.

4.1 Any Role Exchange Algorithm

Role exchange happens whenever the agent pair's IUG>0. Since any role exchange will occur when it is applicable, we call it "any role exchange" algorithm. This algorithm was presented in [12] and is given below in Figure 2:

1. There is no role adoption for any agent at the very beginning. $t = 0 \Rightarrow F = \varnothing$.

2. When role adoption starts, each agent adopts a role randomly, which means that the agent may adopt any role at first. $t = 1 \Rightarrow \forall i \exists j, agent(i) \wedge role(j) \wedge U(i,j) = u_{ij} \wedge add(<i,j,u_{ij}>, F)$.

3. Search the agent pairs from the first agent for role exchange. If the IUG of the given pair of agents is positive, the agent pair will make role exchange; otherwise search the next agent pair for role exchange.
$\forall i,j \forall k,l \forall F, agent(i) \wedge agent(j) \wedge role(k) \wedge role(l) \wedge formation(F) \wedge <i,j,u_{ik}> \in F \wedge <i,j,u_{jl}> \in F \wedge$
$IUG_i(i,k,j,l) > 0 \Rightarrow add(<i,l,u_{il}>, F) \wedge add(<j,k,u_{jk}>, F) \wedge delete(<i,k,u_{ik}>, F) \wedge delete(<j,l,u_{jl}>, F)$

4. Role exchanges will stop when the utility gain of any agent pair is no more than zero.
$\forall i,j \forall k,l\ agent(i) \wedge agent(j) \wedge role(k) \wedge role(l) \wedge \sum_{i=1}^{n} IUG_i(i,k,j,l) \le 0 \Rightarrow stop.$

Figure 2. Any Role Exchange Algorithm

4.2 Individual Optimal Role Exchange Algorithm

We start role exchange from the first agent. Search all other agents paired with first agent, find the pairing with the maximum IUG, perform the role exchange corresponding to the maximum IUG. Then repeat this step with other agents. If for all agent pairing produce IUG less or equal to zero, no role exchange is needed. Since for every agent, we find the best applicable role exchange and the agents will be checked in order (from the first one to the last one), we will call it "individual optimal role exchange" algorithm and it is shown in the following Figure 3.

1. There is no role adoption for any agent at the very beginning. $t = 0 \Rightarrow F = \varnothing$.

2. When role adoption starts, each agent adopts a role randomly, which means that the agent may adopt any role at first.
$t = 1 \Rightarrow \forall i \exists j, agent(i) \wedge role(j) \wedge U(i,j) = u_{ij} \wedge add(<i,j,u_{ij}>, F)$.

3. Search the agent pairs from the first agent for role exchange. Find the maximum IUG of the given agent pair related with the first agent, if the IUG of the given pair of agents is positive, the agent pair will make role exchange, repeat step 3; otherwise search the next agent pair start with the second agent for role exchange, repeat step 3.
$\forall i,j \forall k,l \forall F, agent(i) \wedge agent(j) \wedge role(k) \wedge role(l) \wedge formation(F) \wedge <i,j,u_{ik}> \in F \wedge <i,j,u_{jl}> \in F \wedge$
$IUG_i(i,k,j,l) > 0 \wedge IsMaxFori(i,k,j,l) \Rightarrow add(<i,l,u_{il}>, F) \wedge add(<j,k,u_{jk}>, F) \wedge delete(<i,k,u_{ik}>, F) \wedge$

Figure 3. Individual Optimal Role Exchange Algorithm

4.3 Group Optimal Role Exchange Algorithm

Here agents share knowledge of IUGs and refrain from early role exchanges. IUGs are computed and the best one is performed and this repeated until no more role exchange is applicable. I.e., search all the agent pairs, find the maximum IUG pair, do role exchange. Repeat this steps until the IUG for all the agent pairs are no more than 0. We can also call it “group optimal role exchange” algorithm in Figure 4.

1. There is no role adoption for any agent at the very beginning. $t = 0 \Rightarrow F = \varnothing$.

2. When role adoption starts, each agent adopts a role randomly, which means that the agent may adopt any role at first.
$t = 1 \Rightarrow \forall i \exists j, agent(i) \wedge role(j) \wedge U(i, j) = u_{ij} \wedge add(< i, j, u_{ij} >, F)$.

3. Search all the agent pairs from the first agent. Find the agent pair with maximum IUG, if the IUG of the given pair of agent is positive, do role exchange.
$\forall i, j \forall k, l \forall F, agent(i) \wedge agent(j) \wedge role(k) \wedge role(l) \wedge formation(F) \wedge < i, j, u_{ik} > \in F \wedge < i, j, u_{jl} > \in F \wedge$
$IUG_i(i, k, j, l) > 0 \wedge IsMaxForAll(i, k, j, l) \Rightarrow add(< i, l, u_{il} >, F) \wedge add(< j, k, u_{jk} >, F) \wedge delete(< i, k, u_{ik} >, F) \wedge$
$delete(< j, l, u_{jl} >, F)$

4. Role exchanges will stop when the utility gain of any agent pair is no more than zero. $\forall i, j \forall k, l \; agent(i) \wedge agent(j) \wedge role(k) \wedge role(l) \wedge \sum_{i=1}^{n} IUG_i(i, k, j, l) \le 0 \Rightarrow stop.$

Figure 4. Group Optimal Role Exchange Algorithm

5 Implementation and Comparison of Algorithms

Consider the following example involving role exchanges with 2 agents and their 2 roles. We can use a matrix to model this problem. Based on assumption 1 and 2, suppose there are N agents and N roles, we can use N*N matrix to represent the relationship between agents and roles. The rows represent agents such as agent A0 and columns represent roles, such as role R0. The value at the intersection of an agent row and a role column, such as element (i, j), represents the utility that agent i adopting role j. In an implementation of this algorithm, we use a 10*10 matrix shown in Table 1, whose utilities are randomly generated. Here we used the Wichiman-Hill algorithm to generate uniform distributed random number in (0,1), and we repeated the implementation for several times.

5.1 Implementation

According to the algorithm we discussed above, no role has been adopted at first. So we may just assign each agent A_i with role R_i, as the entities highlighted in the table. Based on assumption 3 and 4, at this time, the initial total utility of the group is $\sum_{i=0}^{i=9} V(Ai, Ri) = 71$.

Then based on assumption 5, 6 and 7, we will check each agent pair to decide if role exchange is necessary or not based on conditions we discussed in 2.1 and the condition of the algorithms.

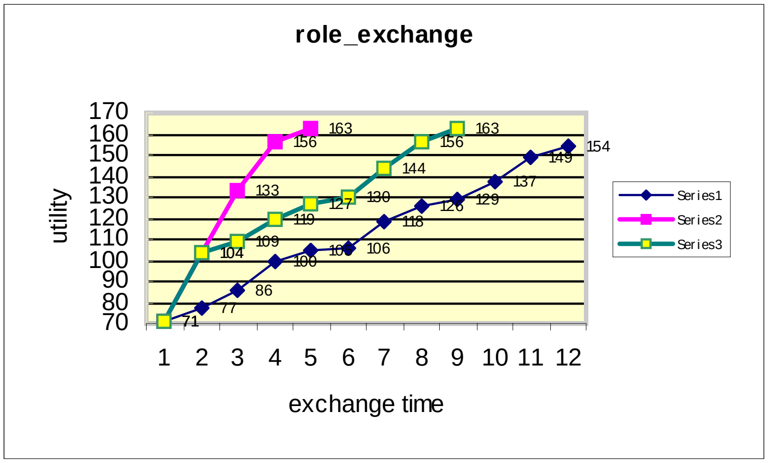
Table 1. Agent-Role Table Pairs

	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9
A0	1	7	14	0	9	4	18	18	2	4
A1	5	5	1	7	1	11	15	2	7	16
A2	11	4	2	13	12	2	1	16	18	15
A3	7	6	11	1	8	9	12	7	19	15
A4	14	3	11	2	13	13	4	1	11	13
A5	8	7	4	2	17	17	19	3	1	9
A6	18	16	15	10	2	8	6	0	2	4
A7	8	6	5	10	9	10	10	6	1	13
A8	8	9	3	4	14	16	0	6	16	11
A9	8	4	19	6	3	17	18	18	2	9

5.2 Comparison of the 3 Algorithms

Figure 5 shows the results of comparison between 3 utility-based role exchange algorithms. Series 1 represents the role exchange using algorithm 1, Series 2 represents corresponds to algorithm 2, and Series 3 represents algorithm 3.

By comparing our 3 algorithms, we find that algorithm 1's overall computation time



is the smallest, but it suggest the most number of role exchanges. Algorithm 2's overall computation time and the number of role exchanges are both at mod-level. Algorithm 3's overall computation time is the largest, but the number of role exchanges is the least. If we consider time consumption or utility loss due to the role exchange to be nontrivial, then algorithm 3 should be the best choice.

Figure 5 Role exchange with 3 algorithms

Since the utility values in agent-role table are randomly chosen, it's hard to decide which algorithm will yield the best formation after role exchange in a specific situation. But we know that for each agent-role table that there must be an optimum formation with regards to utility. Here, we define a term "optimization percentage" in equation (3).

"optimization percentage" = total_utility/optimized utility (3)

Total_utility is the final utility of the group of agents after each role exchange algorithm stops. Optimized_utility is the utility yielded by the optimized formation, which yields the maximum total group utility.

We manually (not by either of our algorithms) calculate this optimized value by considering all possible formations, which can be generated from the given agent-role table. We then compare this value with the output results from our 3 algorithms. For an empirical result, we repeated this 1000 times with different agent-role utility tables and accumulated the results. Our results of optimization percentage for 3 algorithms are 97.7274%, 97.5173%, and 98.2839% respectively as in Table 2. From these results we are inclined to suggest that the 3 algorithms' "optimization percentage" are statistically equivalent and does not provide an adequate reason for selection. We now define the term "unit_time_gain" in equation (4) to measure the average utility gain per role exchange:

"unit_time_gain" = total_utility_gain / role_exchange_times (4)

Total_utility_gain is the utility gain for a run of the algorithm. Role_exchange_times is the number of role exchanges at the end of the algorithm.

From Figure 5, we see that the total utility of the whole group changes from 71 to 154 with algorithm 1, from 71 to 163 with algorithm 2, and from 71 to 163 with algorithm 3. The *unit_time_gain* for algorithm 1 is 6.91, for algorithm 2 it is 10.2, and for algorithm 3 it is 23.

The comparison results are shown in Table 2. We compare those 3 algorithms in three aspects, such as the total utility, the optimization percentage and the unit time gain for each algorithm respectively. If based on a lot of experiments we can assume that all 3 algorithms yield the "same" total utility, the algorithm with the highest "*unit_time_gain*" will be the best one. So, the "group optimal role exchange" algorithm is the best one among these 3 algorithms for utility-based role exchange.

Table 2 Comparison results of the Algorithms

	Total Utility	Optimization Percentage	Unit Time Gain
Algorithm 1	154	97.7274%	6.91
Algorithm 2	163	97.5173%	10.2
Algorithm 3	163	98.2839%	23

6 Conclusion

We presented three utility-based role exchange algorithms. Two of these algorithms had not been reported prior to this paper. We presented the concept of "optimization percentage" and "unit_time_gain" to measure the performance of the algorithms for utility-based role exchange. We discussed the implementation of the algorithms and analyzed the advantage and disadvantage of each algorithm for utility-based role exchange. All three algorithms yield similar results. However, if we assume role exchange to involve non-trivial cost, a committee based choice of role exchange, we called "group optimal role exchange" algorithm produces the minimum exchanges and is preferred.

References

[1] Boman, M., Kummeneje, J., Lyback, D., Younes, H. L.: UBU Team. **RoboCup-99 Team Descriptions Simulation League**, (1999) pp 133-138

[2] Cavedon, L., and Sonenberg, L., On Social Commitments, Roles, and Preferred Goals, In Proceedings of 3rd Int Confernce on Multiagent Systems, pp. 80-87, (1998).

[3] Castelfranchi, C. and Falcone, R., From Task Delegation to Role Delegation, In Proedings of AI-IA 1997: Advances in Artificial Intelligence Congress, LNAI 1321, pp. 278-289.

[4] Hexmoor, H., Zhang, X., Norms, Roles and Simulated RoboCup. 2nd workshop on norms and institutions in multiagent systems **(Agents 2001)**, Montreal, CA, ACM press, (2001).

[5] Kendall, E., Role Models, Patterns of Agent System Analysis and Design. British Telecom Technical Journal Special Issue on Decentralized Business Systems, 1999.

[6] Marsella, S., Adibi, J., Al-Onaizau, Y., Kaminka, G., Musela, I., Tambe, M., Experiences acquired in the design of Robocup teams: A comparison of two fielded teams, Journal of Autonomous agents and Multi-agent Systems Journal of Autonomous agents and Multi-agent Systems (**JAAMAS),** special issue on "Best of Agents'99", (2000)

[7] Reis, L. P., Lau, N., FC Portugal Team Description: RoboCup 2000 Simulation League Champion. RoboCup 2000: **Robot Soccer World Cup IV**, Berlin, (2001)

[8] Stone, P., Veloso, M., Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth communication for Real-Time Strategic Teamwork. Artificial Intelligence Journal (**AIJ**), Volume 100, number 2, June (1999)

[9] Tu, M.T., Griffel, F., Merz, M., Lamersdorf, W., A plug-In Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents. Mobil Agents: Second International Workshop (MA'98), Stuttgart, Germany, Sep., 1998, pp 222-236.

[10] Veloso, M., Stone, P., Han, K., The CMUnited-97 Robotic Soccer Team: Perception and Multiagent Control. Agents'98, Minneapolis, May, 1998, pp. 78-86.

[11] Veloso, M., Pagello, E., Kitano, H., RoboCup-99: Robot Soccer World Cup III. **RoboCup**, (1999)

[12] Zhang, X., Hexmoor, H., Utility-based Role Exchange. In the Proceedings of Central and Eastern Europe on Multi-Agent Systems (**CEEMAS-01**), Poland. (2001), pp. 413-420.