Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

MULTI-AGENT LEARNING MODEL WITH BARGAINING

Haiyan Qiao
Jerzy Rozenblit

Electrical and Computer Engineering Department
University of Arizona
Tucson, AZ 85721-0104, USA

Ferenc Szidarovszky

Systems and Industrial Engineering Department
University of Arizona
Tucson, AZ 85721-0020, USA

Lizhi Yang

Electrical and Computer Engineering Department
University of Arizona
Tucson, AZ 85721-0104, USA

ABSTRACT

Decision problems with the features of prisoner's dilemma are quite common. A general solution to this kind of social dilemma is that the agents cooperate to play a joint action. The Nash bargaining solution is an attractive approach to such cooperative games. In this paper, a multi-agent learning algorithm based on the Nash bargaining solution is presented. Different experiments are conducted on a testbed of stochastic games. The experimental results demonstrate that the algorithm converges to the policies of the Nash bargaining solution. Compared with the learning algorithms based on a non-cooperative equilibrium, this algorithm is fast and its complexity is linear with respect to the number of agents and number of iterations. In addition, it avoids the disturbing problem of equilibrium selection.

1 INTRODUCTION

Interactive decision problems with multi-agents are common in real life, e.g, peace-keeping, wireless sensor networks, team robot, etc. A difficulty in such decision problems is in the uncertainty in model parameters, i.e., the probability of a state transition and agents' payoff function. Multi-agent reinforcement learning is an innovative approach to solve this type of problems. It combines the learning process in an unknown environment with the interactive decision process of multiple agents.

In multi-agent systems, if each agent acts independently to maximize its individual payoff, the final payoff of each agent is usually undesirable in an unknown decision environment. In fact, even when the environmental parameters are known, in the prisoner's dilemma-like decision problems,

		agent 2	
		cooperate	defect
agent 1	cooperate	(2,2)	(0,3)
	defect	(3,0)	(1,1)

Figure 1: The Prisoner's Dilemma

non-cooperative action of each agent may lead to undesirable payoffs. Consider the well known prisoner's dilemma represented in Figure 1 as a matrix where each row and column represent an action of player 1 or 2, and each cell represents the payoffs to each player for every combination of actions. If each agent plays its dominant strategy, then the game terminates with Nash equilibrium payoffs (1,1). If the two players cooperate with each other and play an optimal joint action, then a better payoff (2,2) is achieved, which is Pareto-optimal (i.e., none of the payoffs/objectives can be improved without worsening another). However, when the agents act independently, they have the fear of being taken advantage of by another agent. Therefore, they will not cooperate with each other. If the agents are able to make an enforceable contract, then they will have a strong desire to cooperate with each other.

In this paper, a type of an interactive decision problem among multiple agents, which has the same essential characteristics as the prisoner's dilemma, is studied and a cooperative game model is adopted to solve it. Since the Nash Bargaining Solution (NBS) is an attractive solution to cooperative games, an NBS based multi-agent learning algorithm is designed. Simulations are performed on a testbed of general-sum stochastic games. Compared to multi-agent

learning algorithms based on noncooperative equilibriums, our algorithm has the following features: (a) there is no equilibrium selection problem, (b) the algorithm ensures convergence to a Pareto-optimal solution, (c) the algorithm is linear to the number of agents and the number of iterations, therefore much faster than the equilibrium based multi-agent learning algorithms.

In Section 2, the related work in multi-agent learning is reviewed briefly. In Section 3, the Nash bargaining solution-based multi-agent learning algorithm is presented and analyzed. In Section 4, the simulation results are shown. Finally, conclusions and future work are summarized in Section 5.

2 RELATED WORK

Multi-Agent Reinforcement Learning (MARL) has been an active research area in Artificial Intelligence for more than a decade and some innovative results have been obtained.

Unlike single agent learning problems, in multi-agent settings there is no single utility function to optimize. Each agent has a different objective and its payoff is determined by the joint action of multiple agents. In such interactive decision problems, there exists many solution concepts, some of which are drawn from game theory, and others are from decision theory. In MARL, the focus is on which solution concept to use to guide the learning.

One type of MARL is value iteration learning (Sutton and Barto 1998) based on different concepts of equilibrium in game theory. Littman (1994) first introduced a Minmax solution-based learning algorithm in zero-sum stochastic games. Hu and Wellman (1998) presented a Nash Equilibrium-based learning algorithm, which extended Littman's algorithm to the general-sum games. Greenwald and Hall (2003) considered the possibility of action correlation among agents and proposed a Correlated Equilibrium based learning algorithm.

Another type of MARL is mainly motivated by multiple-person decision theory. It assumes that each agent plays a best-response against stationary opponents, and requires the joint action of agents to converge to Nash Equilibrium in self-play. Bowling and Veloso (2002) first proposed such an MARL algorithm with a variable learning rate, i.e., learn quickly while losing and slowly while winning. Conitzer and Sandholm (2003) introduced AWESOME, a learning algorithm for repeated matrix games that learns best response when opponents are stationary, otherwise move to equilibrium.

Shoham and Powers (2003) criticized the equilibrium-based MARL algorithms for the disturbing problem of equilibrium selection, and proposed a set of learning criteria (Powers and Shoham 2005) to guarantee that an agent's average payoff is near to best response against stationary opponents, close to an equilibrium in self-play, and at least better than the minimax payoff against the opponents. Correspondingly, a hybrid strategies based-learning algorithm is devised for repeated matrix games. Banerjee and Peng (2005) proposed a no-regret algorithm which satisfied the same set of criteria without the assumption that the learners know game matrices.

In a social dilemma, non-cooperative equilibrium is not always desirable. The agents are able to improve payoffs by cooperation or bargaining. In Stimpson and Goodrich (2003ab), a special learning approach based on non-cooperative bargaining model is designed to solve the social dilemma. It is similar to the equilibrium based MARL algorithms, but still has the problem of equilibrium selection.

In this paper, cooperative games are adopted to solve the social dilemma and a corresponding learning algorithm is designed based on the Nash Bargaining Solution concept.

3 THE NBS BASED MULTI-AGENT LEARNING

3.1 The Nash Bargaining Solution

Agent behaviors are different in different games. Compared to equilibrium, the solution concept in non-cooperative games, the Nash Bargaining Solution (Nash 1953) is a core solution concept in cooperative games.

The Nash Bargaining Solution selects the actions that maximize the product of the utility gains of the agents in comparison to the "status quo". This solution is derived based on a certain set of axioms representing the fairness of the solution. The "status quo" point can be chosen as the noncooperative Nash Equilibrium or from the worst possible outcomes for the agents.

In an unknown environment it is is difficult to find NBS even with an enforceable contract. The agents have to learn throught iterations how to reach the NBS.

3.2 The NBS Based Multi-Agent Learning Algorithm

We adopt a stochastic game framework to multi-agent systems. An n-player stochastic game is described with a tuple

$$(N, S, \{A_i\}_{i\in N}, \{r_i\}_{i\in N}, P),$$

where $N = 1, 2, ..., n$ is the set of agents, S is the state space, A_i is action set of agent i, $r_i : S \times \prod_{j\in N} A_j \to R$ is the payoff function for agent i, and $P : S \times \prod_{j\in N} A_j \to \Delta(s)$ is the transition probability over the state space S.

In multi-agent Q learning, the Q function of agent i is defined as

$$Q_i(s, \vec{a}) = r_i(s, \vec{a}) + \gamma \sum_{s'\in S} p(s'|s, \vec{a}) v_i(s', \pi_1, \ldots, \pi_n), \tag{1}$$

where s' is the new state after joint action $\vec{a}$ is taken at the state s, $r_i(s,\vec{a})$ is the one-period reward at state s under joint action $\vec{a}$, $\gamma \in [0,1)$ is the discount rate, $p(s'|s,\vec{a})$ is the transition probability from state s under $\vec{a}$ to new state s', and $(\pi_1,\ldots,\pi_n)$ is joint strategy which is decided by a specific decision mechanism. By this definition, $Q_i(s,a)$ is agent i's total discounted reward of taking action $\vec{a}$ in state s and then following the specific policy $(\pi_1,\ldots,\pi_n)$ thereafter.

There is a selection function f which defines specific strategy for each agent to play, i.e.,

$$v_i(s,\pi_1,\ldots,\pi_n) = f_i(\vec{a})(Q_1(s,\vec{a}),\ldots,Q_n(s,\vec{a})). \quad (2)$$

When the selection function is a Nash equilibrium, i.e.,

$$V_i(s) \in NASH_i(Q1(s),\ldots,Q_n(s)), \quad (3)$$

where $Q_i(s)$ is agent i's reward matrix at state s, and $V_i(s)$ represents agent i's reward at state s when every agent follows the Nash equilibrium strategy at s.

In reality, the overall reward function and the state transition probability distribution are typically unknown. The advantage of value iteration learning is that the agents can learn even when the model is unknown. The Q function is updated through explorations of the state and action space according to

$$Q_i^{t+1}(s,\vec{a}) = (1-\alpha)Q_i^t(s,\vec{a}) + \alpha(r_i^t + \gamma V(s')). \quad (4)$$

Different selection functions result in different learning algorithms, i.e., Minmax-based learning (Littman, 1994), Nash equilibrium-based learning (Hu and Wellman 1998) or correlated equilibrium (Greenwald and Hall 2003). When we adopt equilibrium as the solution concept, the solution is not guaranteed to be Pareto-optimal.

In a social dilemma, the agents are usually rational and would like to reach a win-win situation. In many situations, each agent is able to improve its objectives without preventing others from improving their objectives. The agents are more prone to coordinate in such situations and willing to play cooperative games. The Nash Bargaining solution to cooperative games has a unique and Pareto-optimal solution. Therefore, NBS is applicable to learning in social dilemma modeled by cooperative games. In NBS based learning, v value is calculated as

$$v_i(s,\pi_1,\ldots,\pi_n) = NBS_i(s)(Q_1(s,\vec{a}),\ldots,Q_n(s,\vec{a})), \quad (5)$$

where

$$NBS(s) = Max_{\vec{a}}(Q_1(s,\vec{a}) \times \ldots \times Q_n(s,\vec{a})). \quad (6)$$

$NBS(s)$ represents the payoff matrix obtained by Nash Bargaining Solution at state s and NBS_i is NBS payoff to agent i.

The learning algorithm is given in Table 1. The learning starts from an initial state. At each iteration, the learning starts from the new state resulting from a previous joint action. If one agent reaches the goal, a random state is chosen at next iteration to continue the exploration of the state space.

Like Nash-Q learning, in NBS based learning the agent learns the strategy without knowing the transition function and the reward function of the game, i.e., the model is unknown. Unlike Nash-Q learning which makes a strong assumption that there is a unique equilibrium, the Nash Bargaining solution is unique at each stage and therefore does not have the equilibrium selection problem.

The algorithm results in mutual cooperation through Nash Bargaining. At each stage of the game, a Pareto-optimal solution is obtained. Thus, the overall solution of the game is also Pareto-optimal. This property is the key for discriminating between various learning algorithms. Stimpson and Goodrich (2003a) demonstrated that many multi-agent learning algorithms fail to discover a Pareto efficient solution.

Table 1: The NBS Learning Algorithm

Inputs:
learning rate α
discount factor γ
T total training iterations
Initialization:
Index the agents by $i, i = 1,2,...,n$
For all $s \in S$ and $a_i \in A_i$,$i = 1,2,...,n$,
let $Q_i(s,a_1,...,a_n) = 1$, $Q^*(s) = min_a Q_i(s,a) = 0$.
$s = s_0$
Loop:
for $t = 1$ to T
Choose action $a_i \in A_i$
Observe $\vec{r} = r_1,...,r_n$; $\vec{a} = a_1,...,a_n$; and new state s'
Update
$Q_j(s,\vec{a}) = (1-\alpha)\cdot Q_j(s,\vec{a}) + \alpha\cdot(r_j + \gamma\cdot V_j(s'))$
$V_j(s') = NBS_j(Q_1(s',\vec{a}),...,Q_n(s',\vec{a}))$
$NBS(Q_1(s,\vec{a}),...,Q_n(s,\vec{a}))$
$=max_{\vec{a}}(Q_1(s,\vec{a}) - Q_1^*(s))\cdot ... \cdot (Q_n(s,\vec{a}) - Q_n^*(s))$
where $NBS(Q_1(s,\vec{a}),...,Q_n(s,\vec{a}))$ denotes NBS payoff matrix and $V_j(s)$ denotes the NBS based payoff to agent j
$s = s'$

3.2.1 Complexity of the Algorithm

According to the algorithm, the learning agent needs to maintain one Q-function for each agent. Let $|S|$ be the number of states in state space, and $|A_i|$ be the size of agent i's action space A_i, then the total number of entries

(a) grid game 1 (b) grid game 2

Figure 2: Grid World Games

in Q_i is $|S| \cdot \prod_{i \in N} A_i$, where $N = 1, 2, \ldots, n$. The total memory space required for n Q-functions is $n \cdot |S| \cdot \prod_{i \in N} A_i$, the same as in Nash-Q learning.

At each iteration, each Q-function is updated and the product of Q values is compared with V-value, so the running time of NBS based learning at each state is $O(n)$, where n is the number of agents. Therefore, the total running time with t iterations is $O(n \cdot t)$, linear to the number of iterations t and the number of agents n.

In Nash-Q learning, the total running time is dominated by the calculation of Nash equilibrium at each iteration and the computation complexity of finding an equilibrium in matrix game can be done with linear programming, which is more expensive than linear. Because of the linear running time, the NBS learning algorithm is much faster than non-cooperative equilibrium based learning algorithms.

3.2.2 Convergence of the Algorithm

The proof can be given along the lines of the convergence proof in Hu and Wellman (2003) with a minor modification.

4 EXPERIMENTS

The commonly used framework of multi-agent systems includes stochastic games and matrix games. Since we model a dynamic process instead of a one-shot play, we adopt stochastic games to model state transition. In the multi-agent learning with stochastic game framework, grid games are commonly used for the experiments. The two grid games (Hu and Wellman 2003, Greenwald and Hall 2003) shown in figure 2 are simulated. In the games, two agents take actions simultaneously. We assume that the agents do not know the location of the targets and the overall payoff function. They can however observe previous actions taken by another player, the new state that results from the joint action, and the immediate payoff of each action to each player.

4.1 Configuration of the Experiments

The state of the game is decided by two agents' joint location (l_1, l_2) where $l_i = 1 \ldots 9, i = 1, 2$. Each agent's action space A_i consists of four one-step movements, $A_i = \{up, down, left, right\}, i = 1, 2$. The payoff function is defined as

$$r_i = \begin{cases} 100 & \text{if } L(l_i, a_i) = goal_i \\ -1 & \text{if } L(l_1, a_1) = L(l_2, a_2) \text{ and} \\ & L(l_j, a_j) \neq goal_j, j = 1, 2 \\ 0 & \text{otherwise} \end{cases}$$

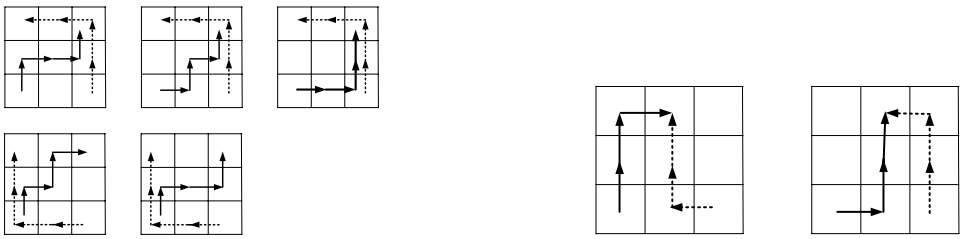
If both agents attempt to move into the same cell, they have to back off and both lose one point. If an agent reaches its goal, it is rewarded 100 points and the game is over. If none of the agents reaches its goal and they do not conflict then each agent scores zero, neither one is rewarded or punished.

The difference between grid game 1 and grid game 2 is that there are two different goals in grid game 1, and one common goal in grid game 2. In addition, game 1 is deterministic, i.e., each action results in a transition to a specific state but game 2 is non-deterministic in cells 1 and 3. In game 2, there is a barrier above cells 1 and cell 3. So when an agent is located in cell 1 or 3 and selects action to move up, it has 50% possibility to succeed and 50% possibility to fail, i.e., to move back to original cell.

The objective of each agent is to reach its goal with the minimum number of steps of movement.

4.2 Experiment Results

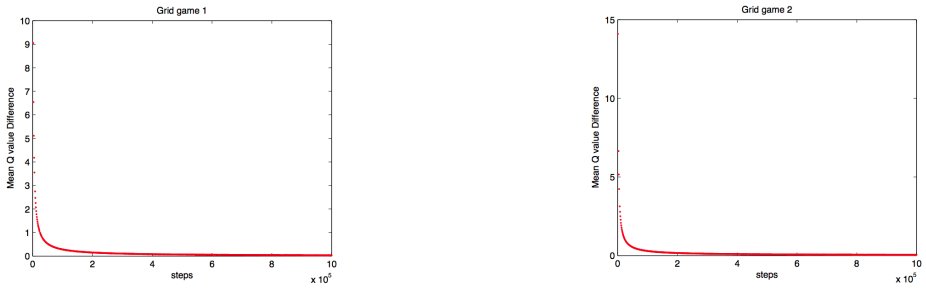
In these two grid games, the Nash Bargaining solution is the same as the equilibrium solution as shown in Hu and Wellman (2003). However, the calculation is much easier and faster since Nash-Q has to calculate equilibrium at each stage while NBS learning just does a simple comparison. In addition, there is no equilibrium selection problem as in Hu and Wellman (2003), Greenwald and Hall (2003). Figure 3 shows the paths from the Nash Bargaining solution. Figure



(a) grid game 1: solution paths

(b) grid game 2: solution paths

Figure 3: Solution Paths in Grid Games



(a) Convergence of grid game 1

(b) Convergence of grid game 2

Figure 4: Convergence in Grid Games

3(a) shows some paths from NBS in grid game 1, and the other NBS paths are symmetric to the ones in Figure 3(a).

The experimental results are presented in Figure 4. The x-axis represents time, and the y-axis represents the means of the distribution of the errors err_i^t for all t and $err_i^t = |Q_i^t(s,\vec{a}) - Q_i^{t-1}(s,\vec{a})|$. The experimental results show that NBS based Q learning converges in both deterministic and nondeterministic grid games. In these two grid games, like equilibrium based learning, there are multiple NBS solution options. In equilibrium based Q learning, such as Nash-Q and CE-Q, there is the question of what equilibrium policies the algorithms converge to. However, in NBS based Q learning, the solution converges to a unique solution without centralized control. At different runs, it may reach different NBS solutions. When there are multiple optimal solutions with the same value, one of the solutions is reached depending on the state transition during the iterations.

The experiments are benchmarks used in a MARL study. In comparing the convergence speed and the number of iterations in Figure 4 with those in Nash-Q learning (Hu and Wellman 2003) and in CE-Q learning with the same experiments, the number of iterations are similar, in the order of 10^5. Therefore, the comparison has to be made on the number of operations per iteration step. At each iteration, the calculation of NBS is linear, but the calculation of Nash equilibrium and correlated equilibrium can be calculated using linear programming. Therefore, the overall computation time of NBS based learning is less and it is much more computationally efficient.

For different applications, the state space, actions set, and short-term reinforcements have to be modeled differently, but the memory space and algorithm complexity are the same as it was analyzed in previous section.

5 CONCLUSION

In this paper, we have applied cooperative game theory to model and analyze a certain type of interactive decision problem with the prisoner's dilemma features. A Nash Bargaining Solution-based learning algorithm has been presented. The experiments have been performed on the testbed of stochastic games. The experimental results have shown that the algorithm always converges to the Nash Bargaining solution.

In most social dilemmas, when the agents make enforceable contracts to resolve their conflict, the power of agents makes a big difference. Therefore, a possible extension of the algorithm is to explore the influence of the agent's power on multi-agent learning in cooperative games, i.e., the multi-agent learning with asymmetric agents.

One problem with the value iteration reinforcement learning algorithms is the scalability. In the proof of con-

vergence, the assumption is made that every state and action have been visited infinitely often. When the state and action spaces are large, it is not efficient to obtain a solution. Therefore, another future work is to make agents have bounded memory in order to speed up convergence.

REFERENCES

Banerjee, B., and J. Peng. 2005. Efficient No-Regret Multiagent Learning. In *Twenty National Conference on Artificial Intelligence* (AAAI 2005), 41-46.

Bowling, M. and M. Veloso. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136: 215-250.

Claus, C., and C. Boutilier. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Fifteen National Conference on Artificial Intelligence*, 746-752. Madison, WI.

Conitzer, V. and T. Sandholm. 2003. AWESOME: A General Multiagent Learning Algorithm that Converges in Self-Play and Learns a Best Response Against Stationary Opponents. In *Proceedings of the twentieth International Conference on Machine Learning* (ICML 2003), 83-90. Washington, DC.

Greenwald, A. and K. Hall. 2003. Correlated Q-learning. In *Proceedings of the Twentieth International Conference on Machine Learning* (ICML 2003), 242-249. Washington DC.

Hu, J. and M. P. Wellman. 1998. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning* (ICML 1998), 242-250. Madison, WI.

Hu, J. and M. P. Wellman. 2003. Nash Q-learning for general-sum stochastic games, *Journal of Machine Learning Research* 4:1039-1069.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Eleventh International Conference on Machine Learning* (ICML 1994), 157-163. New Brunswick, NJ.

Littman, M. L. and P. Stone. 2001. Implicit Negotiation in Repeated Games. In *Proceedings of the Eighth International Workshop on Agent Theories, Architectures, and Languages*, 393-404.

Nash, J. F. 1953. Two-person Cooperative games. *Econometrica* 21:128-140.

Powers, R, Y. Shoham. 2005. New criteria and a new algorithm for learning in multi-agent systems. In *Nineteenth International Joint Conference on Artificial Intelligence* (IJCAI 2005), 817-822. Edinburgh, Scotland.

Shoham, Y., R. Powers, and T. Grenager. 2003. Multi-Agent Reinforcement Learning: a critical survey. *Technical Report*

Stimpson, J. L. and M. A. Goodrich. 2003a. Learning To Cooperate in a Social Dilemma: A Satisficing Approach to Bargaining. In *Twentieth International Conference on Machine Learning* (ICML 2003): 728-735. Washington DC.

Stimpson, J. L., and M. A. Goodrich. 2003b. Nash Equilibrium or Nash Bargaining? Choosing a Solution Concept for Multi-Agent Learning. *Proceedings of the 2003 AAMAS* Workshop on Game Theoretic and Decision Theoretic Agents. Melbourne, Australia, July, 2003.

Sutton, R. S. and A. G. Barto. 1998. *Reinforcement Learning: An introduction*. MIT press.

AUTHOR BIOGRAPHIES

HAIYAN QIAO is a PhD student in Electrical and Computer Engineering Department at the University of Arizona. She holds a MS degree of Computer Science from North Dakota State University and a BS degree of Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China. Her research interests are multiagent systems, machine learning, and game theory. Her e-mail address is <haiyanq@ece.arizona.edu>.

JERZY ROZENBLIT is Professor and Head of the Electrical and Computer Engineering Department at the University of Arizona. During his tenure, he has established the Engineering Design Laboratory with major projects in design and analysis of complex, computer-based systems, software engineering, embedded systems, and symbolic visualization. The projects have been funded by the National Science Foundation, US Army, Siemens, Infineon Technologies, Rockwell, McDonnell Douglas, NASA, Raytheon, and Semiconductor Research Corporation. He has extensive teaching experience and conducts a vigorous graduate program as evidenced by many successful PhD and MSc students and Best Teacher awards. Jerzy is active in professional service in capacities ranging from editorship of ACM and Society for Computer Simulation Transactions, program and general chairmanship of major conferences, to participation in various university and departmental committees. Among several visiting assignments, he was a Fulbright Senior Scholar and Visiting Professor at the Institute of Systems Science, Johannes Kepler University, Austria, Research Fellow at the US Army Research Laboratories, Visiting Professor at the Technical University of Munich, and Fulbright Senior Specialist in Cracow, Poland. Over the years, he has developed strong associations with the private sector and government entities. His management and project experience includes over $8 million in externally funded research. He had served as a research scientist and visiting professor at Siemens AG and Infineon AG Central Research and Development Laboratories in Munich, where over the last decade he was instrumen-

tal in the development of design frameworks for complex, computer-based systems. For the last eleven years, he has led a vigorous research program at the University of Arizona in visualization, human-computer interaction, and artificial intelligence funded by the US Army. Currently, jointly with the Arizona Simulation Technology and Education Center, he is developing virtually assisted surgical training methods and systems. Co-author of several edited monographs and over a hundred publications, Jerzy holds the PhD and MS degrees in Computer Science from Wayne State University, Michigan and the MSc in Computer Engineering from the Technical University of Wroclaw, Poland. His e-mail address is <jr@ece.arizona.edu>.

FERENC SZIDAROVSZKY is a professor of Systems and Industrial Engineering at the University of Arizona. He received his BS, MS, and PhD degrees in Mathematic from the Eotvos University of Science, Budapest, Hungary, where he became a professor in 1968 in the Department of Numerical and Computer Mathematics. In 1977 he received a second PhD degree in Economics from the University of Economic Science, Budapest, Hungary. The Hungarian Academy of Sciences awarded him with the degrees of Candidate in Mathematics in 1975 and Doctor of Engineering Sciences in 1986. In 1988 he joined the University of Arizona. His research interests are systems theory, game theory, conflict resolution, multicriteria decision making, computational mathematics and their applications in economics and natural resources management. His e-mail address is <szidar@sie.arizona.edu>.

LIZHI YANG is a PhD student in Electrical and Computer Engineering Department at the University of Arizona. He holds a MS degree in System and Industrial Engineering from the University of Arizona and a BS degree in Auto Engineering from Zhejiang University, China. His research interests are applications in distributed wireless sensor networks, learning theory, stochastic process, and discrete event modeling methodologies. His email address is <lzyang@ece.arizona.edu>.