IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
21-23 September 2009, Rende (Cosenza), Italy

On Modeling Asymmetric Multi-Agent Scenarios

Tomasz Owczarek
Silesian University of Technology, Faculty of Organization and Management, Zabrze, Poland,
tomasz.owczarek@polsl.pl, http://www.roz6.woiz.polsl.pl/en/index.php?id=Owczarek

***Abstract* – The article presents an example of an extension of Multi-agent influence diagram (MAID) proposed by Koller and Milch with some concepts derived from Sequential influence diagram (SID) described by Jensen, Nielsen and Shenoy. Its main aim is to test if these two graphical languages can be merge in order to deal with an asymmetric multi-agent scenario. An example of such a scenario is given and it is modeled and solved using techniques derived from these two languages. The obtained solution is verified using traditional game-theoretic method.**

***Keywords* - asymmetric scenarios, game theory, structured games, influence diagram**

I. Introduction

Game theory has been widely used in modeling multi-agent systems in which autonomous agents interact with each other. However, although two basic forms of representing games, i.e. normal form (game matrix) and extensive form (game tree) are able to model every real-life situation, they suffer from unnecessary exponential growth and lack the ability of describing the dependence relationship between variables [1]. During the last few years there appeared some new concepts of game representation, such as game network [2], graphical game [3] and multi-agent influence diagram (MAID) [4], which allow to represent games in a more compact way. These new forms of game representation (described in [5] as "structured games") arose out of the attempts of connecting the single-agent graphical models of reasoning under uncertainty with techniques from game theory, and adapting them to multi-agent scenarios [6].

Derived from the probabilistic graphical models such as Bayesian nets or influence diagram, structured games inherits their difficulties with modeling asymmetric situations – when possible variables' states depend on the previous moves. There are many graphical techniques for dealing with asymmetric single agent decision problems [7]. One of the most recent developments in this area is sequential influence diagram (SID) presented in [8]. SID can be seen as an influence diagram with some nodes connected together (creating a cluster of nodes which allows to represent a partial temporal ordering between variables) and with structural arcs, each associated with a condition (called a guard) under which the next variable in the scenario is the variable associated with the node that the arc points to. Solving an SID requires an additional computational structure called a decomposition graph.

The aim of the paper is to extend the multi-agent influence diagram framework with some methods used in SID. We take the concepts derived from SID and give an example of how to use them to model and solve an asymmetric multi-agent scenario (represented as a game with incomplete information) when the decision variables are split variables, i.e. agents' decisions changes structure of the scenario.

II. Multi-Agent Influence Diagram

Multi-agent influence diagram (MAID) was proposed by Koller and Milch in [4] as a new representation of games, especially games with incomplete information. In general, MAID is an influence diagram (ID) (see [9, 10]) extended to the multi-agent environment. Like ID, MAID is a directed acyclic graph consisting of three types of nodes: decision nodes $\boldsymbol{D}$ (represented as rectangles), chance nodes $\boldsymbol{C}$ (ovals) and utility nodes $\boldsymbol{U}$ (diamonds). Each node in the graph corresponds to a variable in the scenario (game). Decision and utility nodes (variables) are partitioned into disjoint sets $\boldsymbol{D_i}$ and $\boldsymbol{U_i}$ for each $i \in \boldsymbol{N}$, where $\boldsymbol{N}=\{1,2,\ldots,n\}$ is a set of agents. Each variable $C \in \boldsymbol{C}$ is associated with conditional probability distribution (CPD), which specifies probability distribution over a set of possible outcomes of C. For a utility node $U \in \boldsymbol{U}$ game specifies a deterministic real-valued function which represents the utility (payoff) of agent.

A CPD associated with decision variable is called a decision rule. An assignment of decision rules to all $D \in \boldsymbol{D_i}$ is called a strategy of agent i. A strategy profile σ is an assignment of decision rules to all decision variables. It defines a joint probability distribution P_σ over all variables in $\boldsymbol{X}=\boldsymbol{C} \cup \boldsymbol{D} \cup \boldsymbol{U}$ and we can calculate this using the chain rule for Bayesian network [9]. The expected utility $EU_i(\sigma)$ that agent i receives under σ is then equal to the sum of the expected utilities of all i's utility nodes. The aim of each of the agents in MAID is to maximize their expected utility by choosing optimal strategy with respect to the strategy of all other agents. Given a MAID $\boldsymbol{M}$ if, for all $i \in \mathbf{N}$, σ_i is optimal for the strategy profile σ in $\boldsymbol{M}$, then σ is a Nash equilibrium in $\boldsymbol{M}$ (see Def. 3 and 4 in [4]).

Koller and Milch proved in [4] that a game represented as MAID is equivalent to its extensive form and showed how to convert one into another. But it is also shown that MAIDs have some advantages over game trees. Firstly, MAID is usually much smaller than the equivalent game tree (size of a MAID grows linear in the number of variables while size of a game tree is exponential). So

MAIDs are often much more convenient for representing a multi-agent scenario. Secondly, graphical part of a MAID can give us information about dependencies between variables which can simplify the computations of Nash equilibrium.

Koller and Milch provide a definition of a *strategic relevance* – intuitively, a decision node $D' \in \boldsymbol{D}_{-i}$ is strategically relevant to a decision node $D \in \boldsymbol{D}_i$ if in order to optimize a decision rule for D agent i must know the decision rule for D' (a graphical criterion for finding strategic relevance is called *s-reachability*). Strategic relevance allows us to construct a *relevance graph*, a directed graph with only decision variables as nodes and with arcs indicating strategic relevance between them. If the relevance graph is acyclic then the game can be solved simply by choosing the optimal decision rules for the decision variables in the reverse topological order of the decision nodes. If there are cycles then the game is decomposed into smaller ones, according to the strongly connected components (SCC) of the graph. Each of these games can then be solved individually (either by transforming into an extensive form like in [4] or by applying the algorithm from [5]). The computational effort of finding the Nash equilibrium in MAIDs is often much smaller than in game trees.

Example of a game in the MAID form is shown in Figure 1. It is a simple game of incomplete information called "signaling game" [11]. Nature moves first and chooses a "type" for agent 1. This move is represented by a chance node *Type*. Agent 1, knowing his type (this knowledge is represented by informational arc from node Type to node *D1*), makes decision *D1*. The true state of the variable *Type* is not revealed to agent 2 (no arc from node *Type*) who can only observe agent 1's action when making decision *D2*. Agent 1's payoff *U1* depends on the variables *Type*, *D1* and *D2*. Agent 2's payoff *U2* depends on the variables *Type* and *D2*. The relevance graph of this game is cyclic – all agents need to know opponent's decision rule in order to optimize their payoff (even though agent 2 can observe action in *D1*, he does not know the true state of the variable *Type*; knowledge about the decision rule for *D1* would reveal this information and allow to maximize *U2*).

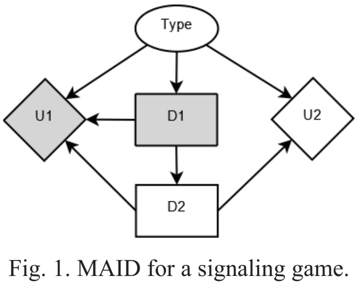

Fig. 1. MAID for a signaling game.

III. Sequential Influence Diagram

Sequential influence diagram (SID) is a language proposed by Jensen, Nielsen and Shenoy in [8] for modeling asymmetric decision problems. In this section we will briefly describe its features. As we are interested only in structural asymmetry, we will omit some of the parts connected with order asymmetry (such as the conception of cluster of nodes), referring to [8] for more details.

In general, an SID is an influence diagram with additional arcs called structural arcs, represented by dashed arrows. These arcs describe the information precedence and structural asymmetry. Each structural arc (X,X') is associated with an annotation $g_{(X,X')}$ called a guard which describes the conditions (the state of the variable X) under which the next node after X is X'.

If a variable belongs to the domain of some guard in SID, it is called a split variable. Each SID induces a partial temporal order $\prec$ on the nodes. Split variable X is called an initial split variable if there is no split variable X' so that $X' \prec X$. Instantiating a split variable means setting its value to a specific state. By instantiating a split variable X to some value x in an SID $\boldsymbol{I}$ we get a new SID $\boldsymbol{I}[X \to x]$.

Solving an SID requires an additional structure, a directed acyclic graph called a decomposition graph. Each of its nodes is associated with pair (X, S), where X is a subset of the decision and chance variables, and S is either a split variable or an empty set. Jensen, Nielsen and Shenoy give a recursive algorithm for constructing a decomposition graph with a tree structure.

Each node in the decomposition graph is initialized with proper probability (tables of conditional or prior probabilities associated with nodes) and utility potentials. After this initialization the decomposition graph is traversed from the leaves towards the root. When visiting a node, using the marginalization and multiplication of the potentials, we eliminate the split variable and free variable associated with the node (see [9] for more information about the algebra of potentials). By recursively eliminating the variables we can get the optimal policy for an agent.

IV. Asymmetric Multi-Agent Influence Diagram – A Simple Blackjack Game

We will begin with presenting a game which we called a "simple blackjack" [12]. Two agents play a card game with a deck containing six cards. The cards are of the value 0, 1 or 2 (two of each kind). Agent 1 draws a card and after seeing its value decides whether to draw another one. After that agent 2 (who saw the agent 1's decision) draws a card and makes a similar decision. A game is won by the agent whose hand has value of 2 or less and either he has more points than his opponent or his opponent has more points than 2. If both agents' hands are of equal values or both have more than 2 points – there is a tie.

An asymmetric MAID for this scenario is shown in Figure 2. There are two decision nodes *D1?* and *D2?* (rectangles) representing decisions made by agent 1 and agent 2 respectively. Four chance nodes *D11*, *D12*, *D21* and *D22* (ovals) represent nature's moves. Variables *D11*, *D12* represent values of the cards from the first and possibly the second drawing of agent 1; similarly *D21*, *D22* represent values of the cards of agent 2. There are also four utility nodes *U1*, *U2*, *U3* and *U4*. Each agent gets 2 if he wins, 1 if there is a draw and 0 when he loses. Since this is a sum-zero game (agents can either win/lose or draw) the utility nodes represent the payoffs of both agents. They are conditioned by the states of the decision variables, e.g. *U3|n,y* means that *U3* is considered in the scenario only when *D1?=no* and *D2?*=yes.

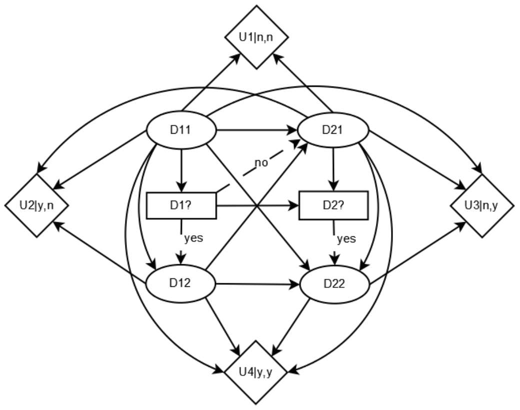

Fig. 2. Asymmetric MAID for "simple blackjack" game.

You can see some structural arcs in the graph. Structural arcs from *D1?* mean that when agent 1 chooses to draw another card, variable *D12* appears in the scenario and makes an impact on the probability distribution of the variable *D21* and possibly *D22*. If *D1?=no*, then *D12* is not present in the scenario and the probability potentials are changed, by marginalizing *D12*: ϕ_{21} changes into

$$\begin{aligned}&\phi_{21}^{-D12}(D11)=\\&=\sum\nolimits_{D12}[P(D21\mid D11,D12)P(D12\mid D11)]=\\&=P(D21\mid D11)\end{aligned}\quad(1)$$

and ϕ_{21} changes into

$$\begin{aligned}&\phi_{22}^{-D12}(D11,D21)=\\&=\frac{1}{\phi_{21}^{-D12}}\sum\nolimits_{D12}[P(D22\mid D11,D12,D2\ 1)\cdot\\&\quad\cdot P(D21\mid D11,D12)P(D12\mid D11)]=\\&=P(D22\mid D11,D21)\end{aligned}\quad(2)$$

Notice that, unlike in SID, informational arcs in this asymmetric MAID are solid arcs to distinguish them from the structural arcs with guards.

A multi-agent influence diagram for the "simple blackjack" game is very similar except that there is just one utility node (representing the payoff of both players) connected by additional arcs with *D1?* and *D2?*, and the structural arcs are replaced by conditional arcs. Comparing to MAID, the asymmetric MAID consists of more variables, but their potentials are often much smaller (e.g. the largest utility potential *U4* has 3^4=81 entries while the utility potential in MAID has $3^4 2^2$=324 entries (It can be reduced by putting some additional dummy variables into the model.). Secondly, to deal with asymmetry some variables in the MAID must contain dummy states which make the graphical representation of the problem not as clear as in the asymmetric MAID.

V. Solving an Asymmetric MAID

Solving a game in asymmetric MAID form requires constructing the decomposition graph and initializing it with potentials. Like in SID, we begin with eliminating the free variables in the leaf nodes and with calculating (for each agent) the new utility potentials. These messages are sent upwards to the parent node. When the potentials come to the node associated with the decision split variable *S*, for each instantiation of the parents of *S* a domination criterion is applied.

Definition *Let S be a decision split variable of agent A with possible states $s_1,s_2,\ldots,s_n$, let $\{\psi_i(\mathbf{X_u},\mathbf{X_o},s_i)\}$ be the set of A's utility potentials differed only in S, where $\mathbf{X_u}$ is the set of the variables unobserved when making a decision in S and $\mathbf{X_o}$ is the set of the variables which state is observed before making a decision in S. Then we say that decision k dominates decision $j \neq k$ in instantiation $x_0 \in dom(\mathbf{X_o})$ if for each $x_u \in dom(\mathbf{X_u})$*

$$\psi_k(x_u,x_o,s_k)\geq\psi_j(x_u,x_o,s_j)\quad(3)$$

If (3) is fulfilled for each $j \neq k$ then we say that decision k is dominating.

A domination criterion is analogous to the classic game-theoretical term [11]. It identifies strategies which are the best responses (considering the current knowledge) regardless of the state of the unobserved variables. When decision is dominating then in the behavior strategy the equivalent action is played with probability 1.

If there are dominating strategies, the partial decision rules are assigned to variable *S*. If the root node is reached and dominating strategies were found, we move downward the tree, again applying a domination criterion (in some cases we can even calculate the expected utility of the decisions using the probability inference). We end this process when there were no dominating strategies found during the iteration. After the iteration process, if all decision rules were assigned, the resulting strategy profile constitutes the Nash equilibrium. A decomposition graph for the "simple blackjack" asymmetric MAID is shown in Figure 3.

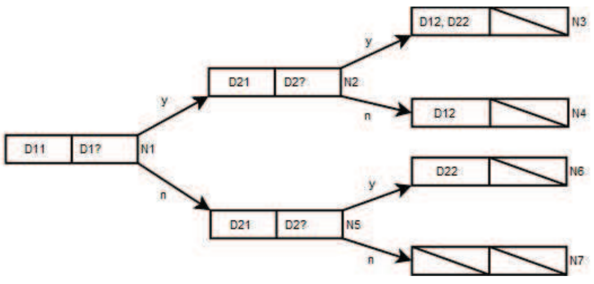

Fig. 3. Decomposition graph for "simple blackjack" game.

It is a tree with internal nodes associated with split variables (there are two split variables: *D1?* and *D2?*). The utility potentials and some free variables (those which are not observed until all decisions are made so their values have no impact on agents' decisions) are associated with the leaves.

We begin with the elimination of the free variables in the leaf nodes and calculate the new utility potentials ψ, e.g. in node *N3* we have:

$$\begin{aligned}&\phi_3^{-D22}(D11,D12,D21)=\\&=\sum\nolimits_{D22}[P(D22\mid D11,D12,D21)\cdot\\&\quad\cdot P(D21\mid D11,D12)P(D12\mid D11)P(D11)],\end{aligned}$$

$$\begin{aligned}&\psi_3^{-D22}(D11,D12,D21)=\\&=\sum\nolimits_{D22}[P(D22\mid D11,D12,D21)\cdot\\&\quad\cdot U4(D11,D12,D21,D22)],\end{aligned}$$

$$\phi_3^{-D12}(D11,D21)=\sum\nolimits_{D12}\phi_3^{-D22}(D11,D12,D21),$$

$$\begin{aligned}&\psi_3^{-D12}(D11,D21)=\\&=\sum\nolimits_{D12}[P(D12\mid D11,D21)\cdot\\&\quad\cdot\psi_3^{-D22}(D11,D12,D21)]=\\&=\sum\nolimits_{D12}[\frac{P(D11,D12,D21)}{P(D11,D21)}\cdot\\&\quad\cdot\psi_3^{-D22}(D11,D12,D21)]=\\&=\frac{1}{\phi_3^{-D12}}\sum\nolimits_{D12}[\phi_3^{-D22}(D11,D12,D21)\cdot\\&\quad\cdot\psi_3^{-D22}(D11,D12,D21)].\end{aligned}$$

In a similar way we eliminate variable *D12* in node *N4*. The calculated potentials ψ_3 and ψ_4 (in the form of 3x3 tables) are then sent upwards to node *N2*, combined and conditioned on the state of the split variable *D2?*. After that, the domination criterion is applied. Table I presents this situation. Here *D2?* is the decision split variable, observed variables are *D21* and *D1?* (which has value "yes") and there is one unobserved variable *D11*. When we compare values in the first two rows we get that regardless of the value of agent 1's first card, it is always better for agent 2 to draw if he gets 0 in the first drawing (in case agent 1 drew second card).

The domination criterion gives us that agent 2 should draw if *D21* equals 0 and he should not draw if *D21* equals 2. Then we do the same with nodes *N6* and *N7* and we get exactly the same decision rule for agent 2 if agent 1 did not draw second card. The potentials are then reduced according to the decision rules we obtained. Next, the variable *D21* is eliminated and the four potentials are sent to the node *N1* and combined, so the resulting potentials are in the form ψ(*D11*, *D1?=yes*, *D2?*) and ψ(*D11*, *D1?=no*, *D2?*). By domination criterion we get the strategy for agent 1: drawing the second card only when the first was 0. This information is then sent downward to the nodes *N2* and *N5* and the utility potentials are reduced according to the agent 1's strategy. Once again the domination criterion gives us that agent 2 should not draw if *D21* equals 1 and agent 1 drew. If *D21* equals 1 and agent 1 did not draw then the domination criterion is not helpful. In this case however it is possible to make probability inference *P*(*D11* | *D21*, *D1?* = no) and this way the expected utility of the two possible actions in *D2?* can be calculated.

Table I
VALUES OF POTENTIALS Ψ_3 AND Ψ_4

		D11		
D21	*D2?*	0	1	2
0	y	1	1.5833	1.5833
	n	0	1	1.5
1	y	0.4167	1	1.083
	n	0.75	1.5	1
2	y	0.4167	0.9167	1
	n	1.75	1.75	1.5

Following the steps described above we get strategies in equilibrium presented in Table II. The same strategy profile we get when solving the "simple blackjack" game tree with a GAMBIT [13].

Table II
STRATEGIES IN EQUILIBRIUM

Agent 1	draw second card only if 0 in the first drawing
Agent 2	1st *strategy*: draw second card only if 0 in the first drawing
	2nd *strategy*: draw second card if 0 in the first drawing OR 1 in the first drawing and the opponent did not draw second card

VI. DISCUSSION AND FUTURE WORK

Using the techniques presented, we will get Nash equilibrium only in limited cases – when we can find best strategies by applying the domination criterion. Even if we do not find equilibrium, we can still take advantage of the calculations to build an extensive form (often with a smaller size than the straightforward representation of a scenario) and use the traditional game-theoretic methods.

The main advantage of using this graphical language is more compact representation of a scenario. Notice that the extensive form of the presented game is a tree with 111 leaf nodes while the asymmetric MAID consists of 10 nodes only. The next thing is the better visualization of the structure of a scenario, a relationship between variables is more explicit and all can be read from the graphical part of a model. With an extensive form sometimes we cannot deduce about variables' dependencies without looking at the quantitative part of a model.

The presented conception of asymmetric MAID opens perspectives for future work. Firstly, we did not give the precise algorithm for solving a game in this form. New problems which we do not deal with will probably arise when formulating this. Next, there are some open questions connected with an SID (e.g. problem of finding the optimal structure of the decomposition graph) which are relative to an asymmetric MAID. It is also still unclear how the structural arcs influence the strategic relevance. Research in this area, especially finding a graphical criterion like s-reachability would be a great extension. Finally, another extension would be finding an algorithm which can directly compute the equilibrium without transforming the game to an extensive.

REFERENCES

[1] J. Kałuski, *Teoria Gier*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2002. (In Polish)

[2] P. La Mura, "Game networks," *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 335-342.

[3] M. Kearns, M. L. Littman, S. Singh, "Graphical models for game theory," *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, 2001, pp. 253-260.

[4] D. Koller, B. Milch, "Multi-agent influence diagram for representing and solving games," *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 2001, pp. 1027-1034.

[5] B. Blum, D. Koller, C. L. Shelton, "A Continuation method for nash equilibria in structured games," *Journal of Artificial Intelligence Resarch*, vol. 25, 2006, pp. 457-502.

[6] S. Parsons, M. Wooldridge, "Game theory and decision theory in multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 5, 2002, pp. 243-254.

[7] C. Bielza, P. P. Shenoy, "A comparison of graphical techniques for asymmetric decision problems," *Management Science*, vol. 45, 1999, pp.1552-1569.

[8] F. V. Jensen, T. D. Nielsen, P. P. Shenoy, "Sequential influence diagrams: a unified asymmetry framework," *International Journal of Approximate Reasoning*, vol. 42, 2006, pp.101-118.

[9] F. V. Jensen, Bayesian Networks and Decision Graphs. Springer-Verlag, 2001.

[10] T. Owczarek, "Modelowanie sytuacji decyzyjnych przy wykorzystaniu diagramów wpływu," *Zeszyty Naukowe Politechniki Śląskiej "Organizacja i Zarządzanie"*, vol. 45, Gliwice, 2008, pp. 305-323. (In Polish)

[11] D. Fudenberg, J. Tirole, *Game Theory*. MIT, Cambridge, 1991.

[12] T. Owczarek, "Czy gracz racjonalny naprawdę jest racjonalny, czyli o równowadze w grach karcianych," *Zeszyty Naukowe Politechniki Śląskiej "Organizacja i Zarządzanie"*, vol. 45, Gliwice, 2008, pp. 293-304. (In Polish)

[13] R. D. McKelvey, A. M. McLennan, T. L. Turocy, Gambit: Software Tools for Game Theory, Version 0.2007.01.30, http://gambit.sourceforge.net, 2007.M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.