A New Approach for Threat Evaluation and Weapon Assignment Problem, Hybrid Learning with Multi-Agent Coordination

Mustafa AZAK
Department of Computer Engineering
Middle East Technical University
Ankara – TURKEY
azak.mustafa@gmail.com

Ahmet Engin BAYRAK
Department of Computer Engineering
Middle East Technical University
Ankara – TURKEY
enginbayrak@gmail.com

***Abstract*—The use of intelligent agents is one of the popular topics in defense industry. Agent usage would be beneficial for defense industry especially in decision making phase of the domain procedures. In Threat Evaluation Weapon Assignment System (TEWAS), we tried to develop learning agents working in coordination for the decision process of command and control systems. This paper describes all details of TEWAS Project in the scope of machine learning techniques.**

I. INTRODUCTION

Defense Industry is one of the biggest consumers of Software Industry. Improving technology allows the development of complex software systems that meets the complex requirements of Defense Industry. Nowadays, many intelligent software products are commonly used for critical mission tasks and for other technological support requirements in militaries. Command and Control (C2) systems mostly use those intelligent software products. Up to now, in that domain only decision support systems were developed, each aims to support people about a problem in mission accomplishment. For those systems, there exists a dependency to human and a risk of performance loss because of human abilities. Ongoing researches have the motivation of producing a decision making system without any human dependency for complex military tasks; such as Threat Evaluation and Weapon Assignment problem of C2 (defense) systems.

Threat Evaluation Weapon Assignment System (TEWAS) tries to provide support for tactical decision making in complex and highly dynamic scenarios where time for decision making and action execution is crucial. Our aim was to design and develop agents to perform threat evaluation and weapon assignment in defense systems without a human dependency.

Being able to reach a novel solution using learning agents was our main motivation. Our approach would be beneficial with an evolving defence performance for the defined problem.

All agents in TEWAS have two goals:

- *Defending the owned platform:* Each agent is located on an armed platform and responsible for using weapon sources to maximize the defense performance.
- *Defending unarmed platforms:* There could be some unarmed platforms that should be defended in military forces. Agents are also responsible for defending those unarmed platforms against the threats.

In order to achieve these defined goals, agents should find solutions for following problems:

- *Threat Evaluation:* Each agent is responsible for evaluating threats to find “which threats are more dangerous for owned platform and for the defended platforms?” [1].
- *Weapon Assignment:* After prioritizing threats, agents in defense system should coordinate to make weapon assignments according to threat priorities. Aim of the coordination is to achieve unique assignment and weapon usage effectiveness [1].

In TEWAS project, each agent performs threat evaluation individually. Agents learn threat evaluation using Backpropagation algorithm with Artificial Neural Networks. Moreover, they learn to coordinate with each other in decision making process of weapon assignment by the help of Reinforcement Learning technique.

TEWAS environment is a board with 16*16 cells. On the board, there are three agents and several threats. Agents are located on the middle of the board in a 4*4 defence zone; on the other hand, each threat can be located on any cell except from defence zone on the board. Only two agents are armed and are aware of all data about threats and their attributes. Therefore, environment can be considered as a fully observable environment. In addition, we assumed that environment is static and it would not change while treat evaluation and weapon assignment phases are being performed.

978-1-4244-2881-6/08/$25.00 ©2008 IEEE

II. PROBLEM DEFINITION AND ALGORITHM

A. Task Definition

TEWAS project was initiated in order to find machine learning solutions for following problems of defense systems:

1) Threat Evaluation

Each agent in a defense system should be able to classify threats according to their potential danger for itself and for the system. This classification should be made individually by each agent. Input set for threat evaluation problem contains the main attributes of threats which is called sensor information as gathered by the help of sensors:

- Distance to agent,
- Speed of threat,
- Distance to defence zone,
- Orientation of threat.

Each agent should create an evaluation output which is one of following classification categories:

- Critical → threat is very dangerous for system
- Potential → threat has a potential for being critical
- Neutral → total danger for a threat is over a threshold but it is not expected to become critical in a short time
- Negligible → threat has a total danger less than defined threshold. It seems unlikely to be critical

2) Weapon Assignment

Armed platforms can use weapons to defend the team. So each agent in a defense system, that controls an armed platform, should be able to make a threat selection to destroy. However, those agents should coordinate while making decisions in order to minimize total weapon usage and maximize defense success. In our weapon assignment problem, each agent is not aware of other agents' threat selection before selecting the threat. However, according to sensor information received, agents can learn which threat is destroyed by other agents afterwards. Input set for weapon assignment problem contains classifications for each threat in a state, which are generated by threat evaluation phase. We used three agents in TEWAS project (a, b and c) and our input for Weapon Assignment phase can be represented as following:

Input Set: $\{n, [<C_{1a}, C_{1b}, C_{1c}>, <C_{2a}, C_{2b}, C_{2c}>, \ldots\ldots, <C_{na}, C_{nb}, C_{nc}>]\}$

Where:

- "**n**" is the number of threats classified
- "C_{ij}" represents the classification made for target number "**i**" by the agent "**j**".

In Weapon Assignment problem, output expected from each armed agent is making a threat selection and destroying it in a given state.

There have been many TEWA problem applications and researches. However, applying a hybrid learning solution which combines different learning algorithms for defined processes is a new approach. Also realization of multi-agent coordination in a learning solution makes our project very interesting.

B. Algorithm Definition

For the problems defined in previous section, we used two different machine learning solutions. For Threat Evaluation problem we used Backpropagation algorithm with an Artificial Neural Network and for Weapon Assignment problem we preferred a modified Q-Learning algorithm. Let's explain the details of solutions.

1) Threat Evaluation

Backpropagation algorithm with an Artificial Neural Network was the best learning technique alternative for the Threat Evaluation because threat classification is performed according to some complex equations and evaluations based on sensor information of the threat. Also there was no delay for getting a reward or output for the classification. Therefore, we designed a feed forward Neural Network where information moves in one direction.

In TEWAS, we used four inputs as input neurons in Neural Network:

- Distance to agent
- Speed of threat
- Orientation of threat.
- Distance to defence zone

In our Neural Network, there are four output units; each corresponds to one of defined classification categories:

- Critical
- Potential
- Neutral
- Negligible

In our Neural Network, each output neuron unit generates an output value between 0 and 1. The unit with the highest output is the classification category for the threat. In addition, there is also a layer for two hidden neurons whose outputs are directly produced by defined threat inputs in the network. Calculated outputs for hidden units are the input values for equations, generating classification output. [2, 3]

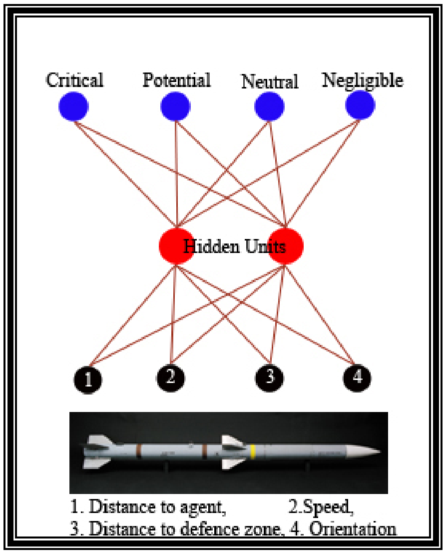

Figure 1 Threat Evaluation Neural Network

Figure 1 shows our complete Neural Network design for threat evaluation learners.

TRACE

In order to trace the solution by an example, assume that there exists a threat in a state with following attributes (inputs):

T = (Distance to agent: 6, Speed of threat: 2, Orientation of threat: -2, and Distance to defence zone: 4)

Our inputs for the neural network would be the given numbers. Then we calculate the values for hidden layer neuron outputs. Assume they are as followings:

Output_H1 : 0.7 and Output_H2 : 0.33

Then assume following output values are calculated for classification output units:

Critical → 0.35, Potential → 0.66, Neutral → 0.42 and Negligible → 0.2

According to given inputs, our neural network classifies the threat as "Potential" as its output is the greatest.

2) Weapon Assignment

In weapon assignment phase, we are training agents by the help of Q learning, which is a well known Reinforcement learning technique. After the training, agents learn how to coordinate with other agents and how to select a threat to destroy in any state of defense system.

Q-learning was suitable for the weapon assignment problem that includes delayed reward and punishment for the performed actions. However, we needed to add agents' coordination task to the standard Q learning algorithm. In our implementation, agents are aware of the state of the defense system at any time but they perceive other agents' actions only after the action is carried out, not before its execution. So each agent also considers the actions that other agents may perform when selecting an action. [4] During threat selection process, each agent assumes that other agents would also make a threat selection according to the state of defense system. In any state, selections are made and actions are performed by agents individually according to the Q-table. However, each agent collects the statistics about other agents' actions and updates a probability vector which is specific for that state after any action is performed. Probability vector and Q values construct a customized Q table for future weapon assignment decisions. [5]

Policy for the weapon assignment solution approach, selects an action that has maximum Q-value for any state of the defense system as defined in standard Q-learning algorithm. But action is not performed by the agent if the Q-table tells that selected action has the greatest probability for being performed by other agents in that state of defense system. Our customized Q-learning solution's policy selects another action from the remaining ones according to their Q-values. [6]

Let's define our Q-learning more formally:

State Representation

For representing a state in Q table we would use:

- Input set for Weapon Assignment defined in previous section:

Input Set: $\{n, [<C_{1a}, C_{1b}, C_{1c}>, <C_{2a}, C_{2b}, C_{2c}>, \ldots\ldots, <C_{na}, C_{nb}, C_{nc}>]\}$

- Status of learning agent that is one of followings:

State_Destroyed, State_NoWeapons, State_HasWeapon

- Status of unarmed platform that is one of followings:

State_Destroyed, State_NoWeapons

Action Selection

In each state of defense system, agent, which owns an armed platform and has remaining weapons, can select a threat from the remaining threats and destroy that one. Action is selected according to a policy that depends on Q-table.

Q-Table Design

Our Q table design can be represented as following:

$S1 \rightarrow [(Q_{A1}, P_{A1}),(Q_{A2}, P_{A2})\ldots(Q_{AK}, P_{AK})]$
$S2 \rightarrow [(Q_{A1}, P_{A1}),(Q_{A2}, P_{A2})\ldots(Q_{AK}, P_{AK})]$
………………………………………….
$SN \rightarrow [(Q_{A1}, P_{A1}),(Q_{A2}, P_{A2})\ldots(Q_{AK}, P_{AK})]$

Where:

- "***SN***" is any state for defense system,
- "Q_{AK}" is the Q-value of action K in corresponding state,
- "P_{AK}" is the probability of action K for being performed by other agents in corresponding state.

Q-Table Update

Q-Values in Q-tables are updated according to the rewards and punishments which are given to each agent in following states:

- All Threats Destroyed: Agent is rewarded. Immediate reward of this state is 100.
- Unarmed Platform Destroyed: As each agent's mission also includes defending the unarmed platform, agent is punished. Immediate reward of this state is -50.
- Agent Destroyed: Agent is punished. Immediate reward of this state is -100.

Other agent's action statistics are updated in customized Q-table after the performance of any action by other agents.

TRACE

In order to trace the solution by an example, assume we have a state of defense systems, in which there are three agents with given statuses and three threats with given classifications:

A1 : State_NoWeapons,
A2 : State_HasWeapon,
A3 : State_HasWeapon
T1 : <Neutral, Negligible, Neutral>,
T2 : <Potential, Critical, Potential>,
T3 : <Critical, Potential, Negligible>

Where:
- C1 is threat classification for A1
- C2 is threat classification for A2
- C3 is threat classification for A3)

in the set <C1, C2, C3>.

Assume Q-table for defined state is as following:

	Q(T1)	P(T1)	Q(T2)	P(T1)	Q(T3)	P(T3)
A2	-0.4	0.01	0.78	0.75	0.62	0.24
A3	0.2	0.05	0.90	0.45	-0.1	0.5

Where:

- Q(T) is Q-Value for the state for selecting threat T by corresponding agent and
- P(T) is probability of threat T for being destroyed by agents other than the corresponding one in table.

Action selection according to given state and Q-tables:

- **Action of A1:** As A1 does not have remaining weapons, that agent does not perform threat selection and destroying.
- **Action of A2:** According to the Q-values; destroying T2 has maximum reward for A2. However, as T2 is the most probable candidate for being destroyed by other agents; A2 selects T3 and destroys it.
- **Action of A3:** According to the Q-values; destroying T2 has maximum reward for A3. In addition, as T2 is not the most probable candidate for being destroyed by other agents; A3 selects T2 and destroys it.

Q-table updates:
- Q-values for each agent's action are updated according to rewards and punishments
- Probability values are updated according to the action performed by agents in defined state.

New Q-table can be similar to:

	Q(T1)	P(T1)	Q(T2)	P(T1)	Q(T3)	P(T3)
A2	-0.4	0.01	0.78	*0.8*	*0.65*	*0.19*
A3	0.1	*0.04*	*0.92*	*0.43*	-0.1	*0.53*

Two armed agents in the given trace, make threat selection for a state of defense system in coordination and destroyed two different threats. However, only T2 would be destroyed if they made selections individually.

III. EXPERIMENTAL EVALUATION

We designed our system in two phases to solve threat evaluation and weapon assignment problems.

A. Methodology

Threat Evaluation phase evaluation: In this phase of system, we are expecting from each agent to classify the threats as accurate as possible according to their sensor information about threats. The outputs of threat evaluation phase play an important role in the weapon assignment phase because target classification is one of inputs in decision making mechanism of agents in weapon assignment. As we discussed in previous section, we used artificial neural networks to solve this classification problem and with training examples, we trained agents to learn how to classify targets by Backpropagation algorithm. Therefore, our main criterion for system performance is how our agents evolve with the increasing number training data and their success rates in classification.

Weapon Assignment phase evaluation: In weapon assignment, armed agents' goal is to be able to make a threat selection to destroy with its weapon. Our main criterion is selecting the most appropriate target for reaching a better state for defence system. Moreover, agents have to behave in coordination to achieve the maximum utilization for their resources. We used a modified Q-learning technique to teach agents how to work with coordination among them.

To be able to evaluate the performance of the agents and our learning algorithm's success, we tried to analyze the performance of the system with increasing number of training data as we did in the threat evaluation phase but we also create two different working modes for system to evaluate achievements in coordination more precisely:

1. Using coordination between agents and updating the Q-tables accordingly in order to observe the performance of coordination.
2. Removing the coordination factor in Q tables and observing independent agents' performance.

We used the following strategy in weapon assignment process for each working modes:

- Train with increasing number of training scenarios.
- Simulate test scenarios until one of the termination conditions reached.
- Collect the amount of scenarios in which agents destroy all of the threats without any loss of agents.
- Calculate win rate and evaluate learning performance.

B. *Results*

Threat Evaluation phase evaluation: After the implementation, we generated 1500 threats that take part in 250 scenarios using scenario generator. We arranged 500 threats for test data and 1000 threats for training data. We assumed that our training data has no noisy data. We divided the training phase into five parts according to training data used in order to evaluate performance of classification made by neural network.

At the end of training and testing session, we observed the following results:

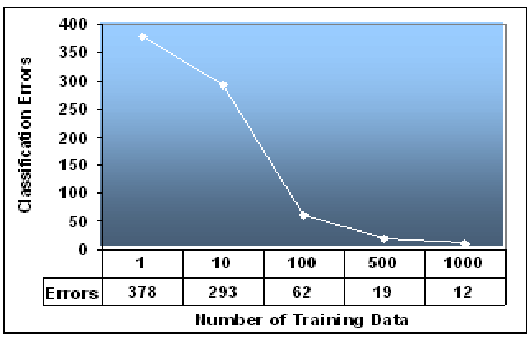

Figure 2: Training data effect on classification

Weapon Assignment phase evaluation: We generated 150 scenarios having 12 threats and we distribute 16 weapons among two armed agents in each scenario using scenario generator. We arranged 100 scenarios for test data and 50 scenarios for training data.

At the end of training and testing session, we observed the following results for each mode:

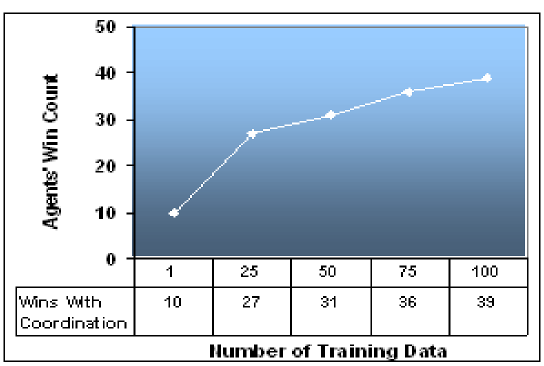

Figure 3: Training data effect on weapon assignment success with coordination

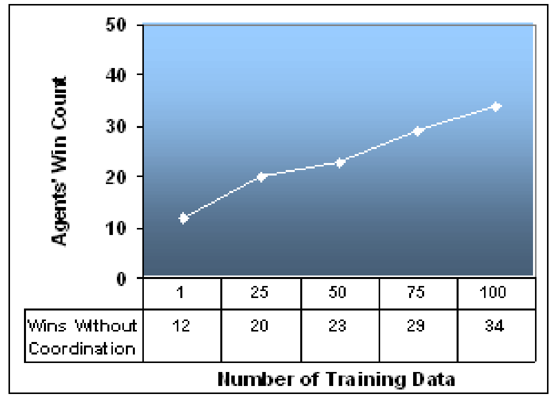

Figure 4: Training data effect on weapon assignment success without coordination

C. *Discussion*

Threat Evaluation phase evaluation: As we expected, error rate decrease with increasing number of training data. However, after some point, performance increase in system slows down and it would probably converge to a constant error rate after sufficient training data supplied. We also observed that Backpropagation algorithm with artificial neural network is successful up to some point, but we have to provide more information about threats to achieve more accurate classification.

Weapon Assignment phase evaluation: After analyzing training and testing session, we concluded that agents benefits from coordination and they can complete scenarios more successfully than independent agents. In addition, we have seen that our default Q-learning algorithm implementation for weapon assignment without coordination is also successful and agents become more mature with sufficient number of training data. We may consider a weakness of our method that although coordination seems to improve agents' performance, it can be claimed that it is not sufficient experimental achievement.

IV. RELATED WORK

There have been many researches and products about TEWA systems. This part of our paper contains a survey of literature and recent developments in TEWA decision support systems within the defence domain. It is observed from various TEWA systems, operator is the clearly central to the overall defence processes. [7] TEWA process result is given to operator and operator perceives this information, processes it in different ways and then decides on a relevant action to take within the given tactical situation. In other words; TEWA systems, understand the tactical situations, evaluate the alternatives and finally make a decision and act accordingly. Decisions are still generally made by under some degree of uncertainty. [8]

Moreover, in recent years, agent and multi-agent technology has become more and more important in this field. Researches in multi agent systems are concerned with the behavior of pre-existing agents' collection that aims to solve a given problem. A multi agent system can be defined as a loosely coupled network of nodes (agents) that work together to solve problems that are beyond the individual capabilities or knowledge of each node. [9] These problem solvers are autonomous and may be heterogeneous in nature. Multi-agent systems are ideally suited to represent problem solving, but have the additional advantage of offering a sophisticated pattern of interaction.

To conclude, we could gather the strengths and weaknesses of existing researches in the area. Following statements are some of their strengths:

- *Experience on Issue:* Computer usage in decision making for defense industry is an old task; experience on this issue is very high. So that their products are advanced and talented.
- *Detailed Domain Knowledge:* Most of the analyzed researches were also supported by government and military. Using the advantage of military support, those researches had probably improved themselves in domain knowledge related with TEWA issue.
- *Real Time Decision Support:* Current researches mostly give importance to real time support for decision making. Decreasing time required for human operators while evaluating tactical situation in dynamic scenarios, are one of the popular issue for those researches.

In addition to the strengths we have mentioned, those researches also have some weaknesses. Following statements are some of those weaknesses:

- *Predefined and Not Evolving Procedure:* Developing an agent to behave intelligently while supporting operators for decision making is predefined procedure and any improvement on their maximum performance is not applicable. Using learning agents in this are would be beneficial as they would create a chance for improving defense system performance.

- *Dependency to Human Operator:* Many TEWA projects are semi-automated that requires human operator to give final decision. That dependency would limit the overall performance of the system to human abilities. In complex scenarios, using full automated defense system would easily achieve defending however semi-automated ones would wait for operators' decisions and probably perform worse than the others. [1]

V. FUTURE WORK

As we discussed in task definition part of paper, we have limited number of inputs for the threat evaluation process and classify the threat with four different categories. These are some of our shortcomings because real life threat evaluation processes are much more complicated and we can extend our input parameters considering real sensors' information and classify threats more accurately according to them and improve our system in future. In addition, in current system we have three agents and we assumed that they have fixed positions. For future work, we may increase the number of armed and unarmed agents to experience the effects of having more agents in multi agents systems and analyze the performance of coordination between agents. Agents that are capable of changing positions will also make our problem more complex but more interesting. With our future work, we can simulate a complex defense system which is completely controlled by software agents.

VI. CONCLUSION

In TEWAS project, we have implemented learning agents for decision making in Threat Evaluation and Weapon Assignment problem of command and control systems. Problem domain was very complex and problem itself has many sub-problems that should be solved. Our motivation in this project was optimizing the performance in decision making for TEWA coordination problem of multi armed platform.

Our implementation includes threat evaluating learners that could coordinate with each other to increase defending performance and weapon usage effectiveness. Our method combines two different machine learning algorithms that were chosen according to their suitability for the related problem. We used Backpropagation algorithm with an Artificial Neural Network for learning of threat evaluation problem and Reinforcement Learning technique for weapon assignment problem.

We can conclude that TEWAS project rationalizes the agent performance in Threat Evaluation and Weapon Assignment decision making by adding learning and multi-agent coordinative deciding features to current researches. Future researches on the same domain can use also some hybrid learning techniques with customization to solve similar problems.

REFERENCES

[1] Threat Evaluation and Weapons Allocation in Network-Centric Warfare, Stephane Paradis, Abderrezak Benaskeur, Martin Oxenham and Philip Cutler

[2] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[3] Tom M. Mitchell, Machine Learning. McGraw-Hill Press, 1997

[4] R.S. Sutton and A.G. Barto. Reinforcement Learning. The MIT Press, Cambridge, MA, 1998

[5] N. R. Jennings, "Coordination techniques for distributed artificial intelligence," in Foundations of Distributed A.I., G. M. P. O'Hare and N. R. Jennings, Eds. NY: Wiley, 1996

[6] M. Wooldridge, An Introduction to Multiagent Systems. Chichester, U.K.: Wiley, Feb. 2002.

[7] P. Beaumont, "Multi-platform coordination and resource management incommand and control",Laval Univ., Pavillon Pouliot, QC, Canada, 2004.

[8] Combat resource allocation planning in naval engagements, A. Benaskeur É. Bossé DRDC Valcartier

[9] Ahuja RK, Kumar A, Jha KC & Orlin JB, 2003, Exact and heuristic algorithms for the weapon target assignment problem, Univ. of Florida Press, Gainesville (TE).