# Image Cover Sheet

| CLASSIFICATION | SYSTEM NUMBER |
| --- | --- |
| UNCLASSIFIED | 154741 |

**TITLE**

USING TEMPORAL DIFFERENCE LEARNING FOR CLOSE-LOOP RESOURCE ALLOCATION

**System Number:**

**Patron Number:**

**Requester:**

**Notes:**

**DSIS Use only:**

**Deliver to:**

UNCLASSIFIED

DEFENCE RESEARCH ESTABLISHMENT
CENTRE DE RECHERCHE POUR LA DÉFENSE
VALCARTIER

DREV - TM-9504
Unlimited Distribution/Distribution Illimitée

# USING TEMPORAL DIFFERENCE LEARNING FOR CLOSED-LOOP RESOURCE ALLOCATION

by

J. Berger, S. Roy, S. Gallant

November/novembre 1995

Approved by/approuvé par

Gaétan Ricard

Director/Directeur

25/10/95

Date

SANS CLASSIFICATION

## ABSTRACT

Closed-loop resource allocation in naval tactical defence is very complex and often relies on the use of specific heuristics showing random variations in the expected solution quality. As a result, problems where combinatorial issues over time are critical to resource allocation require proper models to be defined and suitable algorithms to be developed. A neural network algorithm using the temporal difference method as a learning technique is proposed to solve a closed-loop dynamic weapon-target allocation problem. Directed to select efficient strategies during the planning phase, the learning process is driven by the estimation of the outcome prediction of a battle scenario at a specific time given the sequence of action decisions already taken so far. Combined with past experience the neural network progressively modifies its internal representation of the solution space in order to improve the quality of the decision in timely planning resource allocation. Partial results obtained through computer simulation conducted within the context of naval anti-air warfare for a single problem representation show that the proposed approach fail to match the performance of a greedy heuristic but outperforms a random resource allocation policy.

## RÉSUMÉ

L'affectation de ressources en boucle fermée dans un contexte de défense navale tactique est très complexe et se base souvent sur des heuristiques spécifiques montrant des variations aléatoires dans la qualité des solutions obtenues. Par conséquent, des problèmes pour lesquels les aspects combinatoires associés au passage du temps sont critiques à l'affectation des ressources nécessitent le développement de modèles et d'algorithmes adéquats. Un algorithme de réseau de neurones utilisant la méthode d'apprentissage par différences temporelles pour la résolution d'un problème dynamique d'affectation armes-menaces représenté par un modèle en boucle fermée est présenté. Principalement consacré à la sélection de stratégies efficaces durant la phase de planification, le processus d'apprentissage est guidé par la prédiction du résultat associée avec un scénario de bataille à un moment donné et basée sur la séquence de décisions prises jusqu'à maintenant. Le réseau de neurones modifie progressivement sa représentation interne afin d'améliorer la qualité de la décision liée à la planification des ressources. Des résultats de simulation partiels obtenus dans un contexte de guerre anti-aérienne pour une représentation simple du problème montrent respectivement une performance moindre et supérieure de l'approche proposée en comparaison à une heuristique et une politique aléatoire d'affectation des ressources.

TABLE OF CONTENTS

## EXECUTIVE SUMMARY

Closed-loop resource allocation in naval tactical defence is very expensive by virtue of the intrinsic complexity of the problem. The procedure to solve this problem is generally biased toward a specific policy and the use of empirical methods (heuristics) showing random variations in the solution quality. As a result, problems such as missile launching or illuminator scheduling, where the involved combinatorics and contention over time are critical to resource allocation require proper models to be defined and suitable algorithms to be developed. A learning technique based on knowledge acquired from previous experiments conducted through off-line training of a neural network for a variety of problem classes represents an interesting alternative to be explored, compromising optimality at the expense of execution time, while potentially improving the quality of the computed solution.

The objective of this document is to present a neural network algorithm using the temporal difference method as a learning technique in order to solve a closed-loop resource allocation problem within the naval anti-air warfare context. The algorithm proposed is an incremental learning procedure targeted to prediction problems. The performance of the method is compared with a "greedy" technique and a random-based resource allocation procedure. The strengths and weaknesses of the algorithm are briefly discussed.

Partial results obtained through computer simulation conducted within the context of naval anti-air warfare for a single problem representation show that the proposed approach fails to match the performance of a "greedy" heuristic but outperforms a random resource allocation policy. It is believed that any combination pertaining to improper problem representation, incomplete represented knowledge, inadequate information encoding and inappropriate training adversely contributed to the observed performance. The modest performance obtained through simulation allowed an investigation of the potential benefits of the method in trading off speed and solution quality through off-line learning for a single problem model. Even though a more extensive effort would be required to explore many other alternatives in order to assess the value of the approach for resource planning within tactical naval defence, this work can nonetheless be used as a reference to guide further research using neural networks to perform other prediction tasks within the military domain.

# 1.0 INTRODUCTION

A closed-loop resource allocation problem is generally very difficult to solve by virtue of its dynamic nature and its inherent computational complexity. Typical resolution methods involve approaches such as operational research or decision theory aimed at solution optimality but are somewhat expensive to use. Other approaches rely on specific heuristic procedures. In that case, the cost is usually related to the unpredictable behavior of the heuristic in providing a satisfactory solution for a particular problem instance. However, as combinatorial issues over time are critical to closed-loop resource allocation, a learning technique compromising optimality at the expense of execution time while potentially improving the quality of the computed solution based on knowledge acquired from previous off-line training for a variety of problem classes represents an interesting alternative to be explored.

In this document, a neural network algorithm using the temporal difference method (Ref. 1) as a learning technique is proposed to solve a closed-loop resource allocation problem. The temporal difference method that has been successfully used in selecting game strategies (Refs. 2 and 3) is being explored to address dynamic weapon-target allocation. Directed to select efficient tactical strategies during the planning process, learning is achieved on the basis of the estimation of the anticipated outcome (survival, defeat) prediction of a battle scenario instance at a given time associated with a sequence of action decisions taken over a certain period of time. Combined with past experience the neural network progressively modifies its internal representation of the solution space in order to improve the quality of the decision in timely planning resource allocation. The objective of the method is to determine in what extent this approach can benefit closed-loop resource allocation.

The organization of this document is as follows. In Chapter 2.0 we give a general description of the closed-loop resource allocation problem model to be investigated. The basic assumptions are outlined and the high-level defence model is presented. Then, the technical approach involving neural networks with the temporal difference method used as a learning technique is described in Chapter 3.0. In Chapter 4.0 we present some results from a computational experiment conducted within the context of naval anti-air warfare. The performance of the method is compared with a greedy technique and a random-based resource allocation policy. The strengths and weaknesses of the algorithm are briefly

discussed. The value of the proposed algorithm is then discussed. Finally, we conclude with a short summary in Chapter 5.0.

This effort is part of an ongoing iterative process to develop and analyze new models, advanced algorithms, techniques and methods, single/multiple cooperative intelligent adaptive agent architectures, in order to support real-time resource (defence and computational) allocation and scheduling of renewable and non-renewable resources within the context of tactical naval defence applicable to single and multiple platforms as well as point and area defence scenarios. Initial work was carried out at DREV between June 1994 and August 1994 under PSC 12C, Ship Combat System Integration.

## 2.0 PROBLEM DESCRIPTION

This chapter presents the resource allocation problem to be investigated. A short description of the illuminator scheduling problem is first presented. General assumptions as well as characteristics of the defence model required to represent the problem under study are then given.

The launch or illuminator scheduling problem consists in scheduling target interception using a scarce number of illuminators and interceptors in order to optimize certain criteria. As a critical resource, the illuminator must be allocated for a certain period of time $I$ to a target just before its interception by a surface-to-air missile in its terminal homing phase. A shoot-look-shoot (SLS) doctrine is assumed to be in effect. This means that the outcome of a target engagement must be known before another interceptor is launched against that target.

The problem model being investigated has the following characteristics. A number n of weapon platforms must ensure air-defence against m targets. Each weapon platform i has $N_i$ interceptors. The probability $P_{ijt}$ that an interceptor launched from a weapon platform i destroys a target j being illuminated during time interval t is determined by the battle management, command and control, communications (BM/C3) system. This probability depends on the characteristics of the weapon, the target, the illumination time interval as well as the relative weapon-target geometry. The objective of the defence is to allocate and schedule defensive resources to targets (weapon, illuminator, target, time) over a given period of time $\Delta T$ so as to maximize survivability subject to the resource

constraints of availability and demand of interceptors as well as the constraints set imposed by illuminator scheduling.

## 2.1 General Assumptions

The problem model under study includes the following assumptions regarding the battle environment.

Battle management environment:

- Naval anti-air warfare.
- Ideal meteorological conditions.
- Single ship or task group (convoy/fleet) with weighted assets under heavy sequential or simultaneous anti-ship missile (ASM) attack.
- Constant ship configuration.
- Constant speed for defensive weapons. Straight line trajectory between ship and intercept point.
- Constant speed trajectory for threats.
- Multiple sequential or simultaneous attack from any direction.

Defence:

- Threat size of the attack unknown to the defence.
- Nature of the offensive launch platforms unknown to the defence.
- Offensive launch platform(s) not engageable (submarine, aircraft, ships).
- Size of the offensive stockpile unknown to the defence.
- Targeted assets (threat intentions) before and during the attack unknown to the defence. Defence may only predict at a further stage at which target an offensive weapon is directed.
- Defence periodically knows which threats have been destroyed and ceases to allocate weapons to those threats.
- Each side allocates in ignorance of each other.
- The observation rate is greater than the change rate of the world state, over the entire battle.

- Perfect communication link between the command ship and subordinate ship units (high communication rate, efficient and reliable network).
- No distortions between perceived (blue force) and real universe.

Systems (defensive units):

- Any ship component except the target evaluation and weapon assignment (TEWA) element is considered as a perfect process, there is no noisy or imprecise data, no uncertainty. However, the TEWA process must deal with information incompleteness and knowledge uncertainty.
- Utilization of concurrent resources is possible (under specific constraints).
- Reliable resource.

Based on this context definition, the next section will now introduce the basic defence model.

## 2.2 High-level Defence Model Overview

This section presents the fundamental components of a generic conceptual high-level defence model based on a system environment scheme in order to capture the main characteristics of the problem to be investigated. The environment accounts for the air threats while the system represents the force (ship unit). The model architecture considered, SARA (Situation Assessment and Resource Allocation), is pictured in Fig. 1.

The closed-loop simulator comprises six different types of processes or entities, namely, threat scenario, sensor, weapon, track manager, action manager, and TEWA embodied in the battle manager and portrayed as a particular battle management function. Message passing in SARA is represented in Fig. 1 by self-explanatory underlined inscriptions, attached to directed edges joining two node components (entities). The inscription delineates the nature of the interaction that takes place between two elements.

The environment is defined via the threat scenario entity in which a collection of air threats, namely anti-ship missiles emerging from hostile platforms, is directed toward predetermined targets (ships). Initially specified by the user, the threat scenario characterizes the deterministic kinematic behavior of the air threats.

The system unit including the five remaining elements is made up of sensor(s), weapon(s), a track manager, a centralized TEWA decision-making process, and an action manager. The following paragraphs of this section briefly explain how the system component of the simulator operates and interacts with the environment, from detection to engagement management.

Targets are periodically detected by one or numerous sensors responsible for data gathering and track generation (initiation, update, drop). Tracks generated and later updated by the sensors carry kinematic information (position, speed) depicting the evolution of the target state (state vector). In order to determine or supplement target allegiance, further information is added to the track during the TEWA decision-making process. These operations are periodically accomplished by each sensor entity. Sensory information encoded as a set of tracks is then sent to the track manager entity as an input.

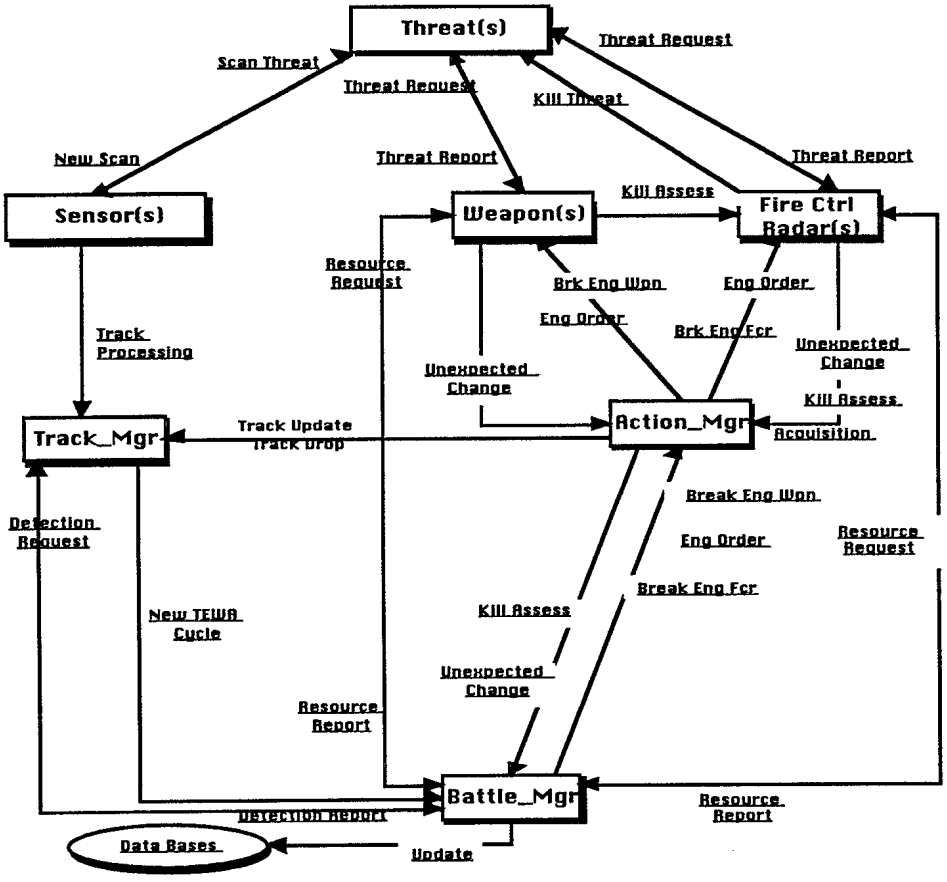


FIGURE 1 - SARA

The track manager is responsible for carrying out track processing and data fusion based on the multiple sensor outputs. This process is assumed to be perfectly executed. The reason for such an assumption is to eliminate the distortions and the uncertainty arising from noisy data and imprecise measurements, in order to match the perceived universe of the system with the real universe (environment). Thereby, the number of tracks obtained after reduction corresponds to the number of threats observed. When a track processing iteration is completed a new TEWA cycle is triggered. Besides, the track manager must regularly handle information request from the battle manager and updated track data from the action manager.

The TEWA component executed in the battle manager entity is the command and control decision-making process responsible for target evaluation and weapon assignment (resource planning task). Being connected to any other system components and based on a centralized concept of operations, it constitutes the cornerstone of the simulation environment. This entity module is the key element of the whole concept demonstrator, making possible the study of various models, algorithms and schemes. Dealing with a set of tracks provided as an input, the TEWA process first diagnoses potential and direct threats and evaluates the threat level posed to the blue force. Based on this information, a suitable weapon suite, properly combined with fire support resources (illuminators), is selected in a timely fashion in order to generate a set of resource allocation plans. As a result, engagement/disengagement orders are issued and sent to a subordinate echelon, the action manager.

The action manager entity accounts for engagement management command and control. As the situation unfolds, this system unit has the responsibility to execute orders and provide feedback information to the battle manager (TEWA) during multiple threat engagements. A typical activity consists of expanding high-level allocation plans in order to initiate concurrent or sequential engagements. This engagement management process relies on proper resource coordination in setting up an appropriate sequence of actions and task scheduling before and during plan execution. As a result, lower level orders are sent to semi-autonomous agents (operators, fire-control computers) in order to initiate weapon deployment (plan execution).

Plan execution is currently realized by weapon and fire-control radar effectors (entities). Resource entities simulate the various operations and related time delays for fire support resources (fire-control radar/illuminator) commitment (start-up, slew, search,

acquisition, lock-on, illumination, kill assessment) and weapon deployment (start-up, slew, search, pointing, fire-control computation, fire). The various actions accomplished by the effectors account for the system environment interactions and determine threat evolution states, closing the simulation loop. Feedback information on the situation such as kill assessment is then immediately reported to the action manager.

The closed-loop event-driven simulation is initiated according to predetermined air-threat scenarios and system specifications. Then, each entity simulation runs concurrently achieving communication via message passing as illustrated in Fig. 1. Central to this communication scheme, the battle manager entity constitutes the most important element, playing a key role on the dynamics of other system components. The simulation goes on until a termination condition is met. Termination conditions are based on overall destruction of the threat or completion of the air attack mission. Simulations may be run in either a deterministic or random (non-deterministic) mode. SARA provides the user with some capability to perform survivability calculations automatically. Through a configuration file, estimated or exact values for ship or force survivability can be determined by statistical analysis or by exhaustive generation of an engagement decision tree. Other measures of performances may also be easily investigated using collected simulation data, namely resource expenditures, utilization, reaction time, number of defeated threats, overall system performance, etc. SARA (Ref. 4) has been implemented using Sim++ (Ref. 5), a process-oriented discrete-event simulation language embedded in the object-oriented language C++.

## 3.0 TECHNICAL APPROACH

### 3.1 Neural Network Description

A neural network algorithm using the temporal difference method (TD) (Ref. 1) as a learning technique is proposed to solve a closed-loop resource allocation problem. Temporal difference learning is a way of extracting information from observation of sequential stochastic processes in order to improve predictions of future outcomes. The algorithm proposed by Sutton (Ref. 1) is an incremental learning procedure specialized for prediction problems. This algorithm uses the difference between the estimated outcome and the real outcome in order to modify the weights of the network. One of the most successful applications using this algorithm is the computer program called TD-Gammon, a neural network able to teach itself to play backgammon solely by playing against itself and

learning from the result. The level of expertise of the network is estimated to be at a strong master level.

Temporal difference methods apply to the training of a system which learns to predict outcome of a temporal sequence of events. Suppose the following sequence having T events:

$$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_{T-1} \rightarrow x_T$$

where $x_i$ is the $i^{th}$ event of the sequence.

The outcome related to this sequence of events is denoted by Z. The task of the system is to provide for each element of the sequence, an estimated value of the outcome. For example, the estimated value of the outcome associated with the event $x_i$ of the sequence is $P_i$ which is an estimate of the real outcome value Z. For a traditional supervised learning system, the computed and desired value pairs of a sequence used to train the network would be:

$$(P_1, Z), (P_2, Z), (P_3, Z), \ldots, (P_T, Z)$$

where $P_i$ represents the new prediction associated with the $i^{th}$ event.

By contrast, the temporal difference method TD(0) would generate the following training pairs:

$$(P_1, P_2), (P_2, P_3), (P_3, P_4), \ldots, (P_T, Z)$$

Each observation is related only to the next time step's prediction and is not associated with the final outcome as in the supervised method. So, the temporal difference method provides the ability to train a network without knowing the outcome of the sequence beforehand. Unlike the back-propagation technique relying on the difference between each prediction as well as the eventual outcome, the training of a neural network based on the temporal difference method is mainly driven by the difference between successive predictions only.

### 3.2 Algorithm Scheme

The learning procedure for updating the network weights is achieved incrementally. For each observation an increment $\Delta w_t$ is computed. A weight is changed by summing all the sequence's increments:

$$w \leftarrow w + \sum_{t=1}^{T} \Delta w_t \tag{3.1}$$

For supervised learning the weight change related to the event t of the sequence is given by:

$$\Delta w_t = \alpha(Z - P_t)\nabla_w P_t \tag{3.2}$$

where $\alpha$ is the learning rate.

The weight change is proportional to the difference between the final outcome and the prediction of the $t^{th}$ event of the sequence times the gradient of the predicted outcome over the weight. The learning rate defines the magnitude of the weight update. The selection of a good learning rate is a very important factor in achieving a satisfactory learning level in a reasonable amount of time.

In order to solve this equation, the final outcome of the sequence has to be known. All observations and predictions made during a sequence must therefore be remembered until the end. In order to remove the actual outcome Z from the weight change equation, equation (3.2) could be modified to be computed incrementally as follows:

$$\Delta w_t = \alpha(P_{t+1} - P_t)\sum_{k=1}^{t} \nabla_w P_k \tag{3.3}$$

This equation indicates an identical contribution associated with each event. However, Sutton defined a whole class of TD algorithm, TD($\lambda$), considering exponential weighting with recency in which alterations to the predictions of observation occurring earlier in the sequence are less significant. $\lambda$ is an adjustable parameter ranging between 0 and 1. The general equation is formulated as follows:

$$\Delta w_t = \alpha(P_{t+1} - P_t)\sum_{k=1}^{t} \lambda^{t-k}\nabla_w P_k \tag{3.4}$$

TD(0) clearly shows a single contribution associated with the last event whereas TD(1) evenly involves all previous events. Note that for $\lambda = 1$ the equation is similar to equation (3.2) which corresponds to the temporal difference implementation of the supervised learning method. The algorithm has been implemented in C++ on an HP9000/730 computer.

# 4.0 COMPUTATIONAL EXPERIMENT

In this chapter, we present some results of a computational experiment conducted within a context of naval anti-air warfare. The simulation experiment first involves a training process in which the neural network builds its internal representation of the world based upon each new battle outcome, and develops its own strategy based on previous experience. The investigation is nonetheless limited to a single neural network configuration. The performance of the algorithm is then compared with some heuristics on the basis of the quality of the computed solution, that is defence survivability.

## 4.1 Simulation

The proposed problem representation involves a multi-layer neural network. The neural network considered includes three layers namely an input , a hidden and an output layer respectively. The first layer encodes information input on the number of targets m and the number of weapon platforms (including fire support resources (illuminators)) n, describing a class of battle scenarios to be investigated. The corresponding neuron elements of this layer are divided into n groups. These groups are then subdivided into m subgroups referring to a specific weapon-target pair. Each subgroup finally involves three neuron elements describing various target features in reference to a weapon platform, namely, relative range, radial velocity and engagement status (or probability of kill of a target being engaged). The encoded information has been restricted to a small number of features and scaled for convergence and normalization purposes respectively. Moreover, it is assumed that the force is composed of units (weapon platforms) with near-similar properties (coverage zones, weapon velocity, set-up time, etc.). The input layer therefore contains approximately 3mn nodes. The second layer containing hidden neurons consists of an intermediate zone and is assumed to be larger than the input layer. It typically comprises one hundred neuron elements or more. As for the output layer, it includes a single neuron element representing an estimation of the prediction associated with force survivability at the current simulated time of the battle. This prediction computed by the neural network can be characterized as the probability of the force to survive an air attack given the history of the battle scenario so far. Details on the implementation and the interface with SARA can be found in Ref. 6.

The training procedure explored during the learning process of the neural network used various combinations of self-learning, supervised learning (heuristic) and random learning (random supervision). Typical strategies emphasized random (or heuristic-

based) supervision during the first phase and then later moved to self-learning within the last phase to refine and attempt to stabilize or generalize concepts of the network associated with its internal representation of the solution space. Given a set of parameters defining the neural network, the number of simulations to be conducted gradually increased with the growing number of targets to be progressively addressed. Accordingly, four, twelve and twenty thousand simulation runs have been performed for one, two, four and eight target scenarios respectively taking into account the various permutations of the neuron elements to be generated during the operation by virtue of the natural notion of symmetry to be recognized and learned by the network. The training experiment aimed to focus on stressful situations (battle scenarios) in order to speed-up the learning process. At each step of the training procedure the intermediate neural network was extracted to monitor its progress and perform statistical analysis. Additional details about the experiment can be found in Ref. 6.

### 4.2 Results

The performance of the algorithm is compared with some heuristics or alternative policies on the basis of the quality of the computed solution, that is defence survivability. Preliminary performance comparisons involves an optimization-based approach (heuristic) used to solve a static weapon-target allocation problem (myopic) and a random-based policy over a sample (10,000) of baseline scenarios randomly generated. The former consists in pairing weapons (and supporting illuminators) to prioritized engageable targets (earliest deadline) in order to maximize the probability of kill over time of intercept ratio. The latter randomly allocates weapons to targets according to constraints imposed by layer defence.

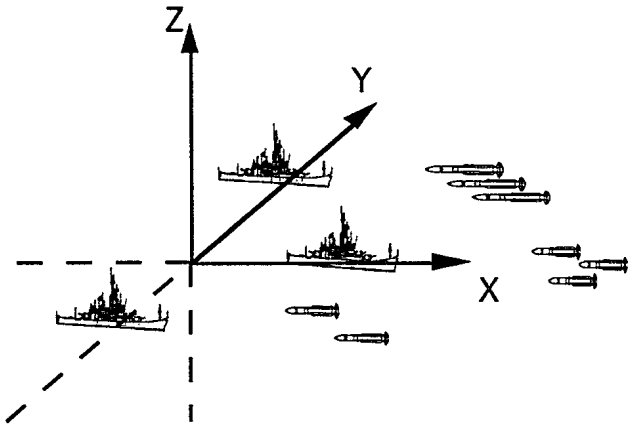


FIGURE 2 - Scenario Instance

The class of scenarios under investigation includes three weapon platforms with predetermined configuration and system parameters coming under attack by eight threats (similar anti-ship missiles) with separation time uniformly distributed in the interval [0,T] where T runs into the interval [32,40] seconds. In fact, based upon discretization of the input space, the spatial (geometry) and temporal (scheduling) distributions of the targets constitute the main degrees of freedom in generating admissible battle scenarios. An instance of such a scenario is represented in Fig. 2.

The near two hundred neuron neural network scheme proposed for this experiment used the basic following reference parameters: $\lambda = 0.7$, $\alpha = 0.02$. The average computational results obtained for a different number of targets over a sample of 10,000 simulation runs are summarized in Table I:

TABLE I - Defence Survivability Average - Comparative Performance

| Number of Targets | Heuristic (myopic policy) | Random-based policy | Neural Network TD($\lambda$) |
|---|---|---|---|
| 1 | 99.6% | 77.1% | 99.1% |
| 2 | 90.2% | 46.8% | 76.6% |
| 4 | 76.0% | 30.4% | 45.7% |
| 8 | 32.0% | 2.5% | 2.7% |

Based on computer simulations achieved for two, four and eight targets, the myopic-based heuristic for resource allocation outperforms the TD-based method despite the progressive improvement of solution quality during the learning phase as monitored until performance saturation. However, better performances are obtained for the TD method when compared with the random policy. The outcome shown by the neural network approach for four and eight targets indicates the difficulty to be overcome by the training procedure to successfully address the computational complexity associated with the problem space. The incidence of this limitation on the generalization capability of the network implies that more intensive training efforts would be required given the constraints imposed by the neural network configuration earlier defined.

Further investigation regarding the impact on the overall performance related to the sensitivity of adjustable parameters (e.g. $\alpha$, $\lambda$), some input variables and the reward

function within the current setting, has not shown significant changes, suggesting that any combinations including improper problem representation, incomplete represented knowledge, network topology, inadequate information encoding and inappropriate training may adversely contribute to the observed performance. This fact appears particularly important for scenarios with a growing number of targets. Other possible explanations include intrinsic problem sensitivity and the nature of state-dependent feedback (non-Markovian process) for the system considered, making the learning process more difficult.

## 5.0 SUMMARY AND CONCLUSION

A neural network algorithm using the temporal difference method as a learning technique has been proposed to solve a closed-loop resource allocation problem. The temporal difference method that has been previously used in successfully selecting game strategies was explored to address dynamic weapon-target allocation. Directed to select efficient strategies during the planning phase, the learning process is driven by the estimation of the outcome (survival, defeat) prediction of a battle scenario at a specific time given the sequence of action decisions already taken so far. Combined with past experience the neural network progressively modifies its internal representation of the solution space in order to improve the quality of the decision in timely planning resource allocation.

Partial results obtained through computer simulation conducted within the context of naval anti-air warfare for a single problem representation show that the proposed approach fail to match the performance of a greedy heuristic but outperforms a random resource allocation policy. It is believed that any combinations pertaining to improper problem representation, incomplete represented knowledge, inadequate information encoding and inappropriate training adversely contributed to the observed performance. This fact appears particularly important for scenarios with a growing number of targets. Other possible explanations include intrinsic problem sensitivity and the nature of state-dependent feedback (non-Markovian process) for the system considered, making the learning process more difficult. Despite the modest performance obtained through simulation, the investigation allowed to probe the potential benefits of the method although limited to a single network configuration, to ultimately trade-off speed and solution quality through off-line learning. It also gave some indications on its capability for generalization, its sensitivity and general limitations.

## 6.0 REFERENCES

1. Sutton R.S., "Learning to Predict by the Methods of Temporal Differences", Machine Learning, 3, pp. 9-44, 1988.

2. Tesauro G., Sejknowski T.J., "A Parallel Network that Learns to Play Backgammon", Artificial Intelligence, 39, 1989.

3. Tesauro G., "Practical Issues in Temporal Difference Learning", Machine Learning, 8, pp. 257-277, 1992.

4. Berger, J., and Chouinard L., "SARA - An Object-Oriented Target Evaluation and Weapon Assignment Demonstrator", The Military, Government, and Aerospace Simulation Conference, San Diego, CA, 1994.

5. JADE Simulations Int. Corp., "Sim++ - A Discrete-Event Simulation Language", JADE Simulations International Corporation, Calgary, Canada, 1991.

6. Gallant S., "Sélection de Stratégies par Réseau de Neurones pour un Problème d'Allocation des Ressources", Université Laval, Quebec City, Canada, August 1994.

**INTERNAL DISTRIBUTION**

**DREV TM-9504**

1 - Deputy Director General

1 - Director Command and Control Information Systems Division

6 - Document Library

1 - Mr. J. Berger (author)

1 - Mrs. M. Bélanger

1 - Mr. R. Carling

1 - Dr. B.A. Chalmers

1 - Mr. D. Demers

1 - Mrs. M. Gauvin

1 - Dr. K. Heaton

1 - Mr. P. Labbé

1 - Mr. L. Lamontagne

1 - Mr. G. Picard

1 - Mr. J. Roy

## EXTERNAL DISTRIBUTION

## DREV TM-9504

2 - DSIS

1 - CRAD

1 - C/DCIEM

1 - C/DREA

1 - C/DREO

1 - C/DRES

1 - DMCS 2

1 - DMCS 4

1 - DMCS-7

1 - DMCS-8

1 - DMCS-9

1 - DMOR

1 - DNR

1 - DRDA

1 - DRDM

1 - Royal Military College
Department of Mathematics and Computer Science
Kingston, Ontario, K7K 5L0
Attn: Command and Control Project

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(Highest classification of Title, Abstract, Keywords)

# DOCUMENT CONTROL DATA

| | |
|---|---|
| . ORIGINATOR (name and address)<br><br>DREV<br>2459 Boul. Pie XI nord<br>Courcelette, Qué.<br>G0A 1R0 | 2. SECURITY CLASSIFICATION<br>(Including special warning terms if applicable)<br><br>UNCLASSIFIED |

. TITLE (Its classification should be indicated by the appropriate abbreviation (S,C,R or U)

Using Temporal Difference Learning for Closed-Loop Resource Allocation (U)

:. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

BERGER, Jean, ROY, Serge, GALLANT, Stressy

| | | |
|---|---|---|
| . DATE OF PUBLICATION (month and year) | 6a. NO. OF PAGES | 6b. NO. OF REFERENCES |
| 1995 | 22 | 6 |

. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. Give the inclusive dates when a specific reporting period is covered.)

Memorandum

. SPONSORING ACTIVITY (name and address)

N/A

| | |
|---|---|
| a. PROJECT OR GRANT NO. (Please specify whether project or grant)<br><br>0112C12A | 9b. CONTRACT NO. |
| 0a.ORIGINATOR'S DOCUMENT NUMBER<br><br>DREV TM-9504 | 10b. OTHER DOCUMENT NOS.<br><br>N/A |

1. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

- [X] Unlimited distribution
- [ ] Contractors in appoved countries (specify)
- [ ] Canadian contractors (with need-to-know)
- [ ] Government (with need-to-know)
- [ ] Defence departments
- [ ] Other (please specify) :

2. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM

13. ABSTRACT ( a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

A neural network algorithm using the temporal difference method as a learning technique is proposed to solve a closed-loop resource allocation problem. The temporal difference method that has been previously used in successfully selecting game strategies is being explored to address dynamic weapon-target allocation. Directed to select efficient strategies during the planning phase, the learning process is driven by the estimation of the outcome (survival, defeat) prediction of a battle scenario at a specific time given the sequence of action decisions already taken so far. Combined with past experience the neural network progressively modifies its internal representation in order to improve the quality of the decision in timely planning resource allocation. Partial results obtained through computer simulation conducted within the context of naval anti-air warfare for a single problem representation show that the proposed approach fail to match the performance of a greedy heuristic but outperforms a random resource allocation policy.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the calssification of each sould be indicated as with the title.)

Neural Network
Learning
Temporal difference
Planning
Closed-loop resource allocation
Simulation

154741

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM