Int. J. Appl. Math. Comput. Sci., 2001, Vol.11, No.2, 493–513

PERIODIC COORDINATION IN HIERARCHICAL AIR DEFENCE SYSTEMS

PIOTR ARABAS*, KRZYSZTOF MALINOWSKI*

The subject of this work is the defence planning of a point target against an air attack. The defence system is decomposed into a number of sectors. A direct method of coordination is used at the upper level, while the sectors use a discrete-time event-based model and the description of uncertainty by multiple scenarios of an attack. The resulting problems are solved using linear programming. A comparison of two coordination strategies for realistic attack scenarios and an analysis of effectiveness are provided.

Keywords: missile defence, hierarchical systems, coordination, decision support systems

1. Introduction

The second half of the 20th century brought about a rapid development of various kinds of missiles, initiated by the first operational use of V1 and V2 during the Second World War. While it was to some extent possible to destroy relatively slow V1s flying at medium attitudes and similar winged missiless, the problem of the anti-ballistic missile defence remained in fact unsolved until the introduction of the first effective surface-to-air missiles (SAMs) in the 1950s (Zdrodowski, 1998). The high speed of both targets and weapons created a demand for control systems which, although the first anti-missile systems were designed for a point defence (e.g. Russian A-35 for defence of Moscow (Lenox, 1998; Zdrodowski, 1999)), would usually cover the whole area of a country. Recent conflicts (especially the wars in the Persian Gulf and former Yugoslavia) demonstrated the role played by the systems responsible for the defence against weapons of a shorter range, such as tactical and cruise missiles.

The objective of this paper is to develop a model for simulating and controlling an anti-missile defence system. Most solutions to this problem (Piasecki, 1968; Piasecki and Boratyn, 1968) take advantage of the probabilistic approach, control theory (e.g. the Lanchester model) (Kimbleton, 1969; Parkhideh and Gafarian, 1996; Przemieniecki, 1994), or game theory (Ardema *et al.*, 1985; Przemieniecki, 1994). The main idea here is to build a hierarchical model which could help to describe real system dependencies and understand their importance better than in the case of a centralized

* Institute of Control and Computation Engineering, Warsaw University of Technology, ul. Nowowiejska 15/19, 00–665 Warsaw, Poland, e-mail: {parabas,K.Malinowski}@elka.pw.edu.pl

one. Another significant issue is that decomposition allows us to use parallel computations and thus to increase the efficiency. The multiscenario algorithm (Warchoł and Malinowski, 1993; 1995) used in this model, although based on classic control theory, introduces new ideas of representing uncertainty.

2. System Description

The system considered is dedicated to defend an important object of moderate size, e.g. an air base or a factory, against an air attack. The weapons are dislocated near the defended object and the protected space can be outlined by a semi-sphere. The radius of this semi-sphere equals the range of weapons. Such a scheme is known as a point defence (Hill, 1988; Zdrodowski, 1998) and is used when limited resources do not allow an area defence system to be organized. The assets of the system may consist of various types of SAM missiles with radar guidance and appropriate command stations. The system is focused on fighting against various kinds of missiles, including ballistic (tactical) and cruise ones. The most important assumption is that the attacking objects come in a limited number of groups. Such an assumption is similar to the situation when the attacker tries to eliminate the SAM radars first (e.g. using anti-radar missiles) to allow better execution of the second phase of the attack, i.e. the attack on the main target. An argument for this can be actions against Iraqi airfields during the war in the Persian Gulf, when within several minutes after eliminating the air defence (often with HARM anti-radar missiles) the runways were mined and the airplanes destroyed.

The proposed command system comprises two levels. The lower level consists of a number (3–5) of sectors commanded by Weapon Directors. The aim of each Weapon Director is to protect his sector using the resources allocated. The upper level—the Air Defence System Commander—coordinates the efforts of the Weapon Directors by allocating resources (mainly weapons) and assigning tasks to sectors. It is important to note that it does not directly define physical borders of the sectors—this allows a more flexible management and is a solution to some common problems (e.g. overlapping sectors). For illustrative purposes, it is still possible to identify the range of the sector as a sector of the fire of its batteries (see Fig. 1).

The framework for the defence system is provided by the use of a hierarchical control scheme and a direct method of coordination. Such a technique is chosen not only for its effectiveness and simplicity, but also for better modelling of the information and responsibility pattern in the real system. That makes this model a better tool for an analysis.

The solution to the decision problem of a Weapon Director is computed with the use of a multiscenario algorithm (Warchoł and Malinowski, 1993) which implies special construction of a model for the sector. As the algorithm assumes multiple scenarios of the attack (uncontrolled inputs), the state values may evolve in several ways following an attack scenario. In other words, at the moments when a particular attack scenario allows for multiple variants of the attack, multiple variants of the state values must also be considered. This is the reason behind marking all variables

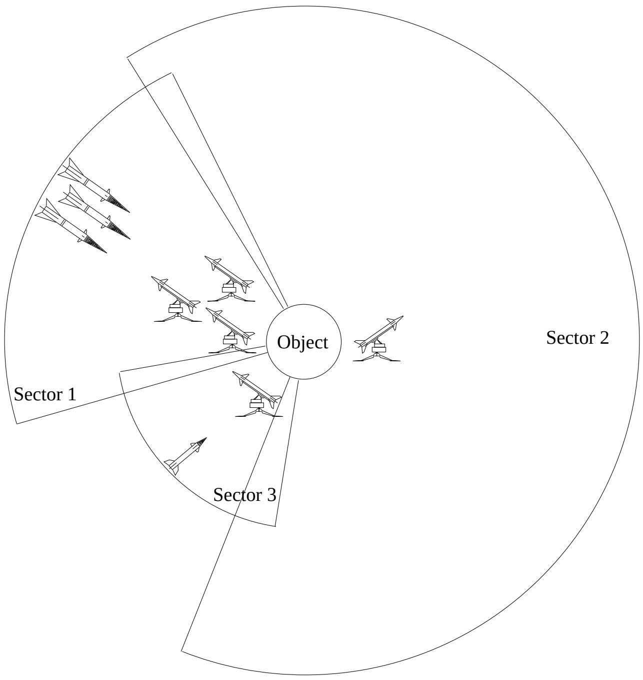


Fig. 1. Schema of the air defence system.

with upper index i denoting the number of the node in the scenario graph instead of time (as is typical for dynamic systems). The idea of multiple scenarios is described more precisely in the next section. Formally, the decision problem of the j-th Weapon Director can be defined as follows: Find

$$\widehat{Q}^j\left(\alpha^j, o^j\right) = \min_{\{\boldsymbol{x}^j\},\{u^j\}} Q^j\left(\{\boldsymbol{x}^j\},\{u^j\},\alpha^j,o^j\right), \tag{1}$$

subject to

$$\boldsymbol{x}^{j,s} = f^j\left(\boldsymbol{x}^{j,i}, u^{j,i}, z(\alpha^j), o^j\right), \quad i=1,\ldots,N, \quad s \in S(i), \tag{2}$$

$$g^j\left(\boldsymbol{x}^{j,i}, u^{j,i}, z(\alpha^j), o^j\right) \leq 0, \qquad i=1,\ldots,N, \tag{3}$$

where $\boldsymbol{x}^{j,i}$ is the state vector at node i, $S(i)$ is the set of successor nodes of node i, i.e. the set of nodes in which the system can evolve due to future actions of the enemy, so $\boldsymbol{x}^{j,s}$ for $s \in S(i)$ represents one of the possible values of the state variable at the time instant subsequent to i. Furthermore, $u^{j,i}$ is the decision vector at node i, $\{u^j\}$ is the decision sequence over the full decision horizon ($i=1,\ldots,N$), z is the prediction of the attack actions, α^j denotes a task assignment, o^j are resources allocated by the coordinator, and $z(\alpha^j)$ denotes the part of the global prediction of attack actions relevant for the j-th Weapon Director. The function $f(\cdot)$ describes the state transformation, and the mapping $g(\cdot)$ represents local constraints. The function

$\widehat{Q}^j(\alpha^j, o^j)$ denotes the optimal value of the performance index of the j-th Weapon Director.

At the upper-level, the coordinator minimizes the overall performance index which is equal to the sum of the Weapon Directors' performance indices by choosing suitable values of variables o and α, i.e. by allocating tasks and resources:

$$\widehat{Q} = \min_{\alpha,o} Q(\alpha, o) = \min_{\alpha,o} \sum_{j=1}^{J} \widehat{Q}^j\big(\alpha^j, o^j\big), \tag{4}$$

subject to

$$h(o) = 0, \tag{5}$$

where J is the number of sectors, and eqn. (5) describes global resources constraints. The coordination can be performed once, at the beginning of the attack (initial coordination), or it can be repeated during the operation of the system (periodic coordination). The moments suitable for computing a new allocation of resources are connected with the arrival of new information and important changes in the situation, e.g. observation of a new group of enemy objects or destruction of a battery allocated to one of the sectors. This allows us to take advantage of the repetitive control scheme applied at the lower level of the system. The special case of periodic coordination discussed in this article is the coordination repeated at each stage, i.e. synchronously with the repetitions of the multiscenario algorithm used by Weapon Directors.

3. Decision Problem of the Defence Sector

As the decision problems of sectors constitute the lower level of the hierarchical system, the model of a sector must provide a compromise between the precision and efficiency. The effect of this is a partially linear model with a simplified description of the process of destroying enemy objects and weapons of the sector by casualty functions (described, e.g. in (Piasecki, 1968; Przemieniecki, 1994)), thus avoiding time-consuming computations typical of stochastic models. An algorithm used for finding optimal decisions also tries to provide better effectiveness than the traditional ones (e.g. stochastic dynamic programming) by the use of multiple scenarios (Warchoł and Malinowski, 1993). This allows us to solve a simpler problem, similar to the deterministic case, without loosing the ability to construct a decision policy.

Modelling the sector involves modelling two different but closely interconnected subsystems: the attacking objects and the local weaponry of the sector. For the attacking objects important information seems to be the distance from the centre of the system to any group and the number of missiles in it. The weaponry of the sector is organized in batteries which may differ in type, number of missiles and capabilities (sector of fire, range, effectiveness, etc.). This implies storing the state of each battery as an independent state variable. For more convenient notation, the state vector $\boldsymbol{x}^j$ is divided into the following variables connected with the attack:

$w_k^{j,i}$ – the number of the attacking objects observed in the k-th group before the time event associated with node i,

$r_k^{j,i}$ – distance from the defence centre to the k-th group of the enemy objects,

and variables connected with the weapons of the sector:

$x_l^{j,i}$ – the number of missiles in the l-th battery available at node i in sector j.

A link between the two groups of state variables is provided by variable $u_{k,l}^{j,i}$ describing the actions taken by the Weapon Manager to defend the sector (i.e. launching missiles) and variable $\varphi_k^{j,i}$ representing the number of enemy objects shot down.

3.1. State Equations

3.1.1. State Equations of Attacking Objects

The main role for attacking object subsystems is played by the equations describing the interaction between the number of attacking objects and decisions represented by the number of missiles launched from the batteries of the sector. The equations describing the changes in the distances to different groups of objects are less complicated as the Weapon Director cannot influence the movement of these objects (in no way other than by destroying them). The following equations describe the changes in the number of attacking objects:

$$w_k^{j^0} = 0$$
$$w_k^{j,s} = \begin{cases} w_k^{j,i} + z_k^i - \varphi_k^{j,i} & \text{if } r_k^{j,i} - v_k^i \tau^{j,i} \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

for $s \in S(i)$, where z_k^i is the number of objects in the k-th group (part of attack prediction), $\tau^{j,i}$ is the length of the time period between nodes i and s, $\varphi_k^{j,i}$ is the random variable describing the number of objects shot down, and $S(i)$ is the set of successor nodes of node i.

It is important to note that the state variable $w_k^{j,i}$ describes the number of attacking objects as observed before (prior to entering) node i. This implies the zero initial condition. The total number of attacking objects present at node i is $w_k^{j,i} + z_k^i$.

The presence of the random variable $\varphi_k^{j,i}$ implies calculation of the expected value of the performance index during optimization. To avoid this inconvenience, the casualty function approach was used, which is equivalent to substituting the expected value of the variable $\varphi_k^{j,i}$ into (6). Although such a solution is only suboptimal, it is widely used (Piasecki, 1968; Przemieniecki, 1994) and allows us to significantly simplify the model as well as speed up the calculations.

The proposed casualty function of the enemy objects is based on the Bernoulli schema and can be calculated as follows:

$$\varphi_k^{j,i}\left(u_{k,l}^{j,i}, e_{k,l}^{j,i}, m_{k,l}^{j}\right) = \mathrm{E}\,\varphi_k^{j,i} = \sum_{l=1}^{L}\left(1 - {e_{k,l}^{j,i}}^{m_{k,l}^{j}}\right)\frac{u_{k,l}^{j,i}}{m_{k,l}^{j}}, \tag{7}$$

where $e_{k,l}^{j,i}$ is the percentage of the objects surviving an attack by a single missile (part of the attack prediction), and $m_{k,l}^{j}$ is the maximal number of missiles which can be used to engage a single enemy object. An important advantage of using such a function is that it is linear and allows us to apply fast and reliable computational methods.

The object movement is described by the equations below. The velocity of objects (v_k^i) should be treated as a mean value and may vary between the nodes:

$$r_k^{j^0} = \begin{cases} f_k^0 & \text{if } z_k^0 > 0, \\ 0 & \text{if } z_k^0 = 0, \end{cases}$$
$$r_k^{j,s} = \begin{cases} f_k^{i+1} & \text{if } z_k^s > 0, \\ \max\left(r_k^{j,i} - v_k^i \tau^{j,i}, 0\right) & \text{if } z_k^s = 0, \end{cases} \tag{8}$$

for $s \in S(i)$, where f_k^i is the distance to the k-th group of objects when spotted (part of the attack prediction), and v_k^i is the velocity of the k-th group of objects (part of the attack prediction).

3.1.2. State Equations of the Weapon

The state of the weapon is defined as a vector representing the number of missiles in the batteries of the sector. It depends not only on decisions (numbers of missiles launched from the batteries), but also on the enemy actions. The targets of the enemy missiles are relatively difficult to predict, and hence a simplified approach was adopted. It relies on the assumption that the defended object occupies a relatively small area and the weapons are placed close to it. This leads to a model in which all the batteries assigned to the sector may suffer from damage. The damage of batteries is calculated with the use of the casualty function whose parameters may vary for different groups so as to describe their abilities:

$$x_l^{j^0} = X_l^{j^0}, \tag{9}$$
$$x_l^{j,s} = \max\left(x_l^{j,i} - \sum_{k=1}^{K} u_{k,l}^{j,i} - \zeta^i\left(w_k^{j,i}, z_k^i, a_k^i\right),\ 0\right), \quad s \in S(i),$$

where $X_l^{j^0}$ is the initial value, i.e. the number of weapons allocated by the coordinator, and $\zeta^i(\cdot)$ is the casualty function of weapon.

A typical casualty function for the area targets can be constructed using a cookie cutter approximation of explosion effects and a Gaussian distribution of the shooting error (Piasecki, 1968; Przemieniecki, 1994):

$$\zeta_\star^i\left(x_l^{j,i}, w_k^{j,i}, v_k^i, a_k^i\right) = x_l^{j,i} \sum_{k \in K^{i\star}} \left(1 - (1 - a_k^i W_{s0})^{(w_k^{j,i} + z_k^i)}\right), \tag{10}$$

$$W_{s0} = 1 - \exp\left(-\frac{R_\mu^2}{\sigma_x^2 + \sigma_y^2}\right), \tag{11}$$

where R_μ is the lethal radius, σ_x and σ_y are respectively the standard deviations of the point of hit in the direction parallel and perpendicular to the line of flight, $K^{i\star}$ is the set of the indices of the groups of objects which cover their targets at node i, z_k^i is the number of objects in the k-th group, and a_k^i is the coefficient describing destructive abilities of an object of the k-th group (both are parts of the attack prediction). The drawback of such a function is that it is nonlinear, and thus the resulting decision problem cannot be solved directly with linear methods. Another possible solution is to use the linear approximation

$$\zeta^i\big(w_k^{j,i}, z_k^i, a_k^i\big) = \min\left(\sum_{k\in K^{i\star}} \frac{a_k^i(w_k^{j,i} + z_k^i)}{L}, x_l^{j,i}\right), \tag{12}$$

where $K^{i\star}$ is the set for indices for the groups of the objects which cover their targets at node i, z_k^i is the number of objects in the k-th group, and a_k^i is the percentage of the objects which hit their targets (both are parts of the attack prediction).

The possibility of using such a function in a repetitive control scheme was proved by experiments. A further discussion of the properties of both the functions together with numerical results can be found in (Arabas *et al.*, 1999).

3.2. Constraints

Constraints describe various physical limitations of both the weapon and detection systems of the sector. Among other things, they provide a limit for the maximum number of missiles that can be launched in particular periods by linking the decision with the number of missiles in batteries and the number of attacking objects:

$$\forall\, l = 1, \dots, L, \qquad \sum_{k=1}^{K} u_{k,l}^{j,i} \le x_l^{j,i}, \tag{13}$$

$$\forall\, k = 1, \dots, K, \qquad \sum_{l=1}^{L} \frac{1}{m_{k,l}^{j}} u_{k,l}^{j,i} \le w_k^{j,i} + z_k^i, \tag{14}$$

$$\forall\, l = 1, \dots, L, \qquad \sum_{k=1}^{K} \frac{u_{k,l}^{j,i}}{m_{k,l}^{j}} \le t_l, \tag{15}$$

$$\forall\, k = 1, \dots, K, \quad \forall\, l = 1, \dots, L, \qquad u_{k,l}^{j,i} \le x_l^{j,i} \delta_l\big(r_k^{j,i}\big), \tag{16}$$

where $\delta_l(\cdot)$ is the function returning 1 when the argument (distance to the enemy object) is less than the range of weapon l, and 0 otherwise. Furthermore, t_l is the maximum number of the enemy objects which can be tracked by radars allocated to the weapon l.

The constraint (13) describes weapon resources of the sector, as it allows the use of no more missiles than available at the particular stage. The constraint (14) limits the number of the missiles which can be used to engage a single enemy object to $m_{k,l}^{j}$ (for some types of weapon it is a typical procedure to launch more then one missile against a single target to attain a higher hit probability). The constraint (15) describes the tracking ability of radars (or other means of detection) allocated to the particular weapon, and the inequality (16) forbids firing missiles to the target outside the range of the weapon.

4. Multiscenario Algorithm

Modelling uncertainty plays an important role in the model of the sector. There are several sources of uncertainty in the proposed model:

1. the number of new attacking (enemy) objects observed at node i and their parameters (speed, distance, azimuth, etc.),
2. the number of attacking objects shot down, and
3. damages caused by the enemy objects.

The last two groups of random variables can be modelled by a purely random process (the values corresponding to subsequent time instants are independent of the previous realizations), and the damage functions described previously are examples of such an approach. The first group of variables describes an attack which is deliberately planned by the enemy and can be changed according to the observations taken during its execution. Such a process involves correlated variables and thus must be specially modelled. As most dynamic optimization algorithms are constructed for disturbances in the form of a white noise, the common solution is state augmentation which allows us to model correlated disturbances as an output of a dynamic system and apply standard methods. The drawback of such a procedure is enlarging the dimension of the whole system, which limits the possibility of using dynamic programming only to simple problems. Since some variables of the proposed model are discrete, the resulting problem cannot be solved analytically. Its dimension (5–8 for a typical attack and system setup) causes a difficulty for dynamic programming even without using time- and memory-consuming techniques like state augmentation. A solution seems to be choosing a method which limits the number of realizations of the uncontrolled variables to a set of a few scenarios considered important for the purpose of defending the object. An algorithm proposed in (Warchoł and Malinowski, 1993) takes advantage of a forecast in the form of multiple scenarios. It is important to note that, although the methods used in the algorithm resemble those used in the deterministic case, the solution is in the form of a control law giving various values of decisions for various realizations of the uncontrolled variables.

Attaching probability to each scenario allows us to construct a situation graph (see Fig. 6), the nodes of which correspond to the time events important for the system. The idea of the algorithm is to treat the nodes located in parallel scenarios (e.g. nodes 2 and 5 in Fig. 6) independently and define state variables at every node (i.e. $x^{j,2}$ at node 2 and $x^{j,5}$ at node 5). Every node needs a different decision, so a natural solution (taking account of the arrangement of the state variables) is to assign to them decisions numbered in the same way as the state variables. As every node provides values of uncontrolled inputs and the initial conditions of state variables are known, it is possible to define the state variables as functions of decisions. Substituting these functions into the performance index allows us to express it in terms of decisions and to solve the resulting problem in decision space only. The solution techniques range from a symbolic method, suitable for some classes of nonlinear problems (Warchoł and Malinowski, 1993), to the use of classical static optimization algorithms. The approach presented in this paper takes advantage of the linear structure of both the model and the performance index, and uses mixed linear programming which is relatively fast and reliable. An additional advantage is that the algorithm can be used in repetitive control scheme, i.e. the computations can be repeated when new information (graph) is available.

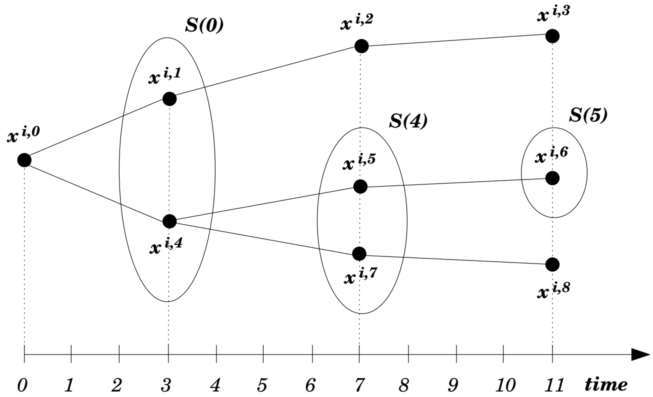


Fig. 2. Possible evolution of the state.

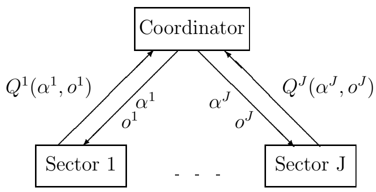


Fig. 3. Hierarchical air defence system.

4.1. Attack Prediction

The possible evolutions of an attack are described by a graph. The example presented in Fig. 6 shows the situation when two scenarios are possible at node 0 ($S(0) = \{1, 4\}$): one with probability $P^1 = 0.7$, leading to the branch starting at node 1, and the other, leading to node 4, with probability $P^4 = 0.3$. Every node is associated with one of the following two types of events:

- A new group of attacking objects appeared. The appearance of the k-th group at node i is indicated by $z_k^i > 0$;
- Objects were hit by missiles; it is possible to observe results of shooting and to launch new missiles.

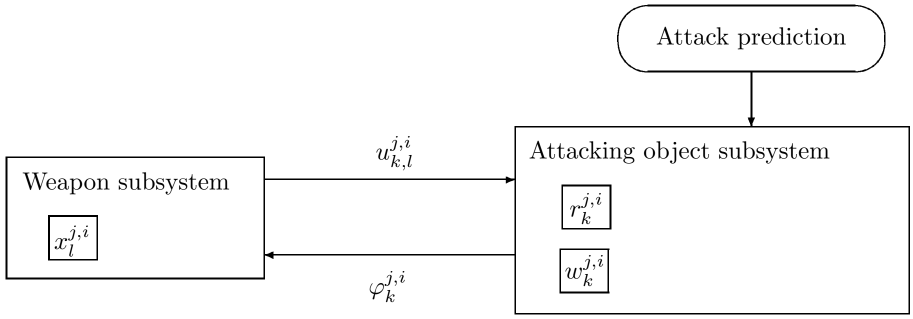


Fig. 4. Components of the model of the sector.

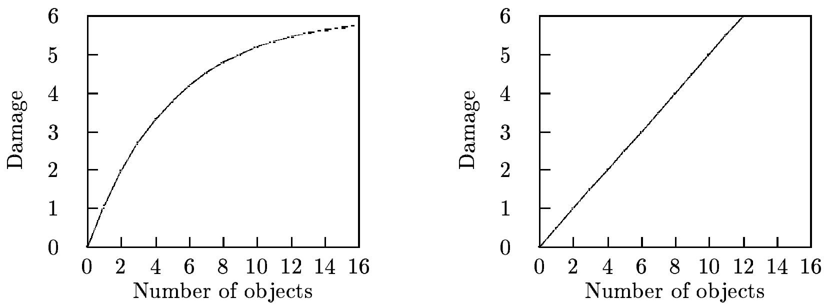


Fig. 5. Nonlinear (left) and (right) linear casualty functions.

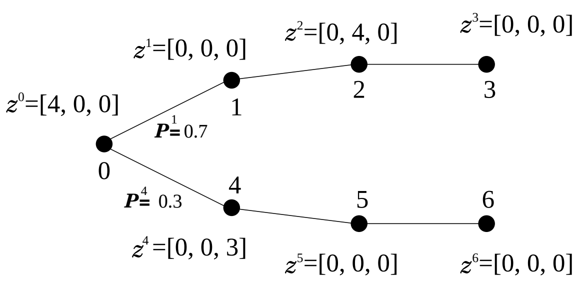


Fig. 6. Example of the prediction graph for an attack consisting of 3 groups ($K = 3$). The numbers next to the nodes are values of z^i ($z^i = [z_1^i, z_2^i, z_3^i]$) and the numbers near arcs are probabilities.

The following values are defined for each node:

- z_k^i – the number of objects appearing in the k-th group at node i; a non-zero value indicates the number of objects at the moment of detection and we have $z_k^i = 0$ for the rest of time,
- f_k^i – the distance to the detection point of the k-th group of objects,
- $\gamma^{j,i}$ – the azimuth at which the k-th group of objects was detected,
- v_k^i – the velocity of the k-th group of objects (mean value for the period of time between nodes i and s, $s \in S(i)$),
- $e_{k,l}^i$ – the probability that an object of the group k will evade weapon l,
- a_k – the probability that an object of group k will hit its target (assuming that it was not destroyed before),
- τ^i – the length of the time period between node i and the next node (node s, $s \in S(i)$),
- P^i – the probability of transition to node i.

4.2. Performance Index

Similarly to stochastic versions of dynamic programming, the performance index of node i is the sum of the stage performance function of this node and of the optimal performance indices of the descendant nodes weighted by the probabilities of those nodes:

$$Q^{j,i}\left(u_k^{j,i}, x_l^{j,i}, w_k^{j,i}, z_k^i\right) = q^{j,i}\left(u_k^{j,i}, x_l^{j,i}, w_k^{j,i}, z_k^i\right) + \sum_{s\in S(i)} P^s \widehat{Q}^{j,s}\left(u_k^{j,s}, x_l^{j,s}, w_k^{j,s}, z_k^i\right), \tag{17}$$

where $Q^{j,i}(\cdot)$ is the performance index of node i, $q^{j,i}(\cdot)$ is the stage performance function of node i, $\widehat{Q}^{j^s}(\cdot)$ is the optimal value of the performance index of node s, $S(i)$ is the set of indices of descendants of the node i, and P^s is the probability of transition to node s.

The stage performance index has the form

$$q^{j,i}\left(u_k^{j,i}, x_l^{j,i}, w_k^{j,i}, z_k^i\right) = \lambda^i\Bigg(p_0\sum_{k=1}^{K}\left(w_k^{j,i}+z_k^i-\varphi_k^{j,i}(u_{k,l}^{j,i}, e_{k,l}^{j,i}, m_{k,l}^j)\right) + \sum_{l=1}^{L} p_l u_{k,l}^{j,i} + p_*\vartheta\left(u_{k,l}^{j,i}, w_k^{j,i}, e_{k,l}^{j,i}, m_{k,l}^j\right)\Bigg), \tag{18}$$

where p_l is the cost per unit of the l-th weapon, p_0 is the 'cost of the presence' of enemy objects in the air space controlled by the Weapon Director, p_* is the cost of damage that can be caused by a single enemy object hitting the defended target, λ^i

is the weight of node i, $\vartheta(\cdot)$ is a function returning the number of the objects hitting their targets at node i:

$$\vartheta\left(u_{k,l}^{j,i}, w_k^{j,i}, e_{k,l}^{j,i}, m_{k,l}^{j}\right) = \begin{cases} w_k^{j,i} - \varphi_k^{j,i}\left(u_{k,l}^{j,i}, e_{k,l}^{j,i}, m_{k,l}^{j}\right) + z_k^i & \text{if } r_k^{j,i} - v_k^i \tau^i \le 0, \\ 0 & \text{otherwise.} \end{cases} \tag{18a}$$

The decision problem of the sector can be defined as minimization of the function obtained by substituting functions defining the state variables in terms of decisions $u_{k,l}^{j,i}$ and prediction factors into the performance index (17) of node 0. The resulting function depends only upon the decision scenario ($\{u^j\}$) and can be optimized in the space of decision scenarios:

$$\widehat{Q}^j(\alpha^j, o^j) = \min_{\{u^j\}} Q^{j,0}\big(\{u^j\}\big) \tag{19}$$

subject to constraints (13)–(16). The result is an optimal decision scenario and an optimal value of the local performance index for allocation of tasks and resources given by the coordinator.

The advantage of such a definition of the decision problem is its relatively simple structure—the constraints (13)–(16) are linear and the performance index is piecewise linear (if a linear approximation of the weapon casualty function (12) is used). Since the casualty function is composed of only two linear pieces, the second of which has the form of a bound (see Fig. 5), it is possible to replace it with the simplified function

$$\bar{\zeta}^i\big(w_k^{j,i}, z_k^i, a_k^i\big) = \sum_{k \in K^{i\star}} \frac{a_k^i(w_k^{j,i} + z_k^i)}{L} \tag{20}$$

and additional constraints defining a new variable $\zeta_l^{j,i}$ denoting the damages of the weapon:

$$\zeta_l^{j,i} - \bar{\zeta}^i\big(w_k^{j,i}, z_k^i, a_k^i\big) \le 0, \tag{21}$$

$$x_l^{j,i} - \zeta_l^{j,i} \ge 0. \tag{22}$$

Substituting $\zeta_l^{j,i}$ in place of the casualty function (12) and adding constraints (21) and (22) results in a linear problem which can be solved with standard methods. As decisions are integer variables and additional variables $\zeta_l^{j,i}$ are continuous, it is a problem of mixed linear programming. The solutions were obtained using a branch-and-bound scheme based on the simplex method chosen for its simplicity and effectiveness.

5. Coordinator

As described in Section 2, the coordination is achieved by the allocation of both weapons and tasks to the sectors. The weapons are allocated in battery units, which seems to be the best solution from the viewpoint of an army organization. Dividing weapons into smaller units, e.g. single launchers, is more difficult as these usually need to share radars and other equipment—this might lead to the situation when radars and launchers of the same battery belong to different sectors. It is important to note that batteries may differ not only in the type of weapons, but also in some other aspects like the number of missiles, sector of fire or effectiveness (i.e. the ability to destroy enemy objects). During operation of the system the number of missiles in the batteries may change (such changes may be caused by launching them or by enemy actions), but even such units can still be reallocated. This means that it is necessary to distinguish batteries, which results in a multidimensional variable o describing the allocation of batteries. Particular tasks are assigned by associating groups of enemy objects with sectors. Formally, the decision problem of the coordinator can be defined as follows: Find

$$\widehat{Q} = \min_{\alpha,o} Q(\alpha, o) = \min_{\alpha,o} \sum_{j=1}^{J} \widehat{Q}^j(\alpha^j, o^j) \tag{23}$$

subject to

$$\forall\, l = 1, \dots, L, \qquad \sum_{j=1}^{J} o_l^j \le 1, \tag{24}$$

$$\forall\, j = 1, \dots, J, \quad \forall\, k\colon\ \alpha_k^j = 1 \qquad \kappa\big(\beta_{\max,1}^j, \beta_{\max,2}^j, \gamma_k^i\big) = 1, \tag{25}$$

$$\forall\, j = 1, \dots, J, \quad \forall\, l\colon\ o_l^j = 1, \qquad \kappa\big(\beta_{\max,1}^j, \beta_{\max,2}^j, \epsilon_{l,1}\big) + \kappa\big(\beta_{\max,1}^j, \beta_{\max,2}^j, \epsilon_{l,2}\big) \ge 1, \tag{26}$$

$$\forall\, j = 1, \dots, J \qquad \sum_{l=1}^{L} o_l^j \ge 1, \tag{27}$$

$$\forall\, k = 1, \dots, K \qquad \sum_{j=1}^{J} \alpha_k^j = 1, \tag{28}$$

where $\widehat{Q}^j(\alpha^j, o^j)$ is the optimal value of the performance index of the j-th sector for given values α^j (task assignment) and o^j (battery allocation), J is the number of sectors, L is the number of batteries. Moreover, $\beta_{\max,1}^j$ and $\beta_{\max,2}^j$ denote the maximum scope of sector j, $\epsilon_{l,1}$ and $\epsilon_{l,2}$ describe the sector of fire of the l-th battery, and γ_k^i is the azimuth of the k-th group of enemy objects at node i. If the function $\kappa(\cdot)$ equals 1, then its third argument is an azimuth lying in the sector defined by the first two arguments, otherwise it equals 0. The allocation of batteries is defined by the matrix

$$o = \begin{bmatrix} o_1^1 & o_2^1 & \cdots & o_L^1 \\ o_1^2 & o_2^2 & \cdots & o_L^2 \\ \vdots & \vdots & & \vdots \\ o_1^J & o_2^J & \cdots & o_L^J \end{bmatrix}, \tag{29}$$

where $o_l^j = 1$ means that the l-th battery was allocated to the j-th sector. Similarly, the $J \times K$ matrix α (K is the number of groups of enemy objects) denotes the task assignment and is composed of the elements α_k^j such that $\alpha_k^j = 1$ when the k-th group of enemy objects is assigned to the j-th sector:

$$o = \begin{bmatrix} \alpha_1^1 & \alpha_2^1 & \cdots & \alpha_K^1 \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_K^2 \\ \vdots & \vdots & & \vdots \\ \alpha_1^J & \alpha_2^J & \cdots & \alpha_K^J \end{bmatrix}, \tag{30}$$

The constraint (24) describes a limitation imposed on the number of batteries—a battery may be assigned to one sector only. The constraint (25) forbids assigning to the defence sector the groups of enemy objects observed outside this sector. In turn, the constraint (26) allows us to avoid the situation when a battery sector of fire lies outside the maximum scope of the defence sector. The constraint (27) tries to minimize the risk related to an unexpected change in the attack (i.e. different than that provided by the attack prediction) by implying allocation of at least one battery to every sector. Such a procedure provides a kind of protection for the rear sector of the system against an unexpected attack. The constraint (28) implies an assignment of all the groups of enemy objects to the sectors (one group may be assigned to one sector only while it is possible for this sector to fight with more than one group).

5.1. Algorithm for Finding an Optimal Allocation

Finding an optimal allocation of tasks and batteries is performed in two stages. First, an optimal allocation of batteries for a given assignment of tasks is determined. Then a new task assignment is generated and the computations are repeated to obtain the optimal solution.

An attack scenario describes several variants of the attack by defining groups of enemy objects. To allocate tasks to the sectors the coordinator needs to assign these groups to the sectors. This involves simple graph processing—reduction of branches without a specified group. As a typical attack consists of a few (2–4) groups, the number of possible assignments is not high (e.g. 26 for the graph used in experiments) and can be further reduced if global constraints are defined.

Finding an optimal allocation of batteries is more difficult as the number of variants may be much higher, e.g. for the system consisting of 3 sectors and 5 batteries (see Fig. 7) there are $4^5 = 1024$ variants. For each variant the local problems for all sectors should be solved. The proposed solution generates all the possible allocations of the batteries for a single sector, as considered separately without taking into account the global part of the resource constraints (constraint (24)). In our example this means that for every sector all possible allocations are considered, i.e. from allocating none of 5 batteries (formally this can be coded as $o^j = [00000]$) to allocating all of them ($o^j = [11111]$). There are $2^5 = 32$ such allocations for the example discussed; some of them must be skipped as violating the constraint (26), thus reducing the total number of variants. The solutions to the sector problems for these allocations can be stored in tables associated with sectors. To find an optimal allocation for the whole system, it is necessary to generate all the combinations of variants stored in sector tables skipping these violating global resource constraints (24, 27), and to compute sums of optimal performance indices (the overall performance index (23) is the sum of the sector performance indices). Comparing these sums allows us to find the optimal one. The advantage of the method is that the number of necessary computations for the sector problems is substantially reduced (32 times in our example), and they can be carried out in parallel, as during computations no communication between sectors is needed (due to relaxing the global constraint). The sequential part finds the optimal sum, which is a relatively simple arithmetic task and takes very little time when compared with the optimization algorithms used to solve sector problems.

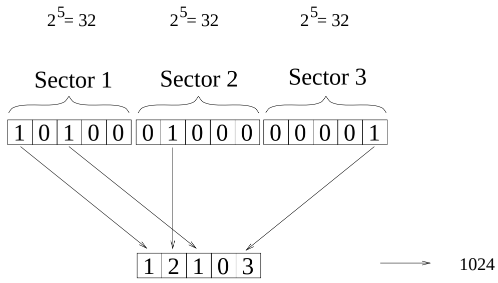


Fig. 7. Scheme of battery allocation.

6. Experiments

The idea of computational experiments was to check the ability of the system to provide a suitable protection under conditions different than predicted in the attack scenario graph, and to check the possibility of using an initial coordination in place of the periodic coordination repeated at each stage. To make it possible, a simulation program incorporating a simulation model was built. The simulation model was based on the model used in computations—(6–10, 12), with an additional modification of the azimuth of attacking groups (γ_k^i) and the number of objects observed ($w_{k,l}^{j,i}$) by normally-distributed random variables. In the case when two variants of an attack scenario were possible, their selection was performed at random.

6.1. Two Scenarios

Simulated scenarios were constructed to correspond to a real situation when the main part of the attack is to be preceded by an anti-radar missile strike. Each attack consisted of two stages. The target of the first were the radar and weapon systems—this was achieved by a large value of a_k^i (the ratio describing the ability to destroy the weapon of the sector) for the first group of enemy missiles. The second part of the attack might arrive from one of two possible directions thus creating two variants represented by two branches of the graph. The value of a_k^i was lower for these groups as they were aimed at the main target. The difference between the two scenarios is that in the first one the second part of the attack occurs after a short delay, so for most of the time both the groups of attacking objects are present. In the second scenario the final part of the attack is carried out after the time needed by the missiles of the first group to reach their targets (radars etc.), making it less demanding when compared with the first situation when the defence system has to fight with two groups concurrently.

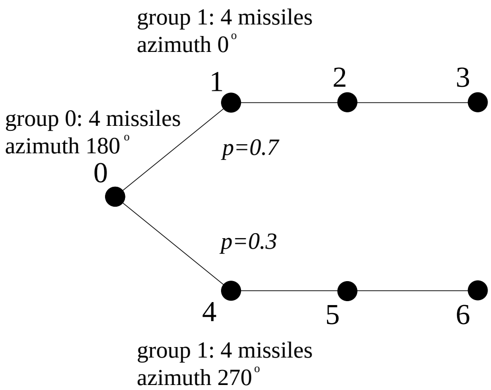
The weapon of the sector consisted of one long-range (80 km) SAM battery and three medium range (45 km) SAM batteries. In order to assess the number of missiles needed to defend the object efficiently, three variants were considered; the corresponding details are provided in Table 1.

Table 1. Battery characteristics.

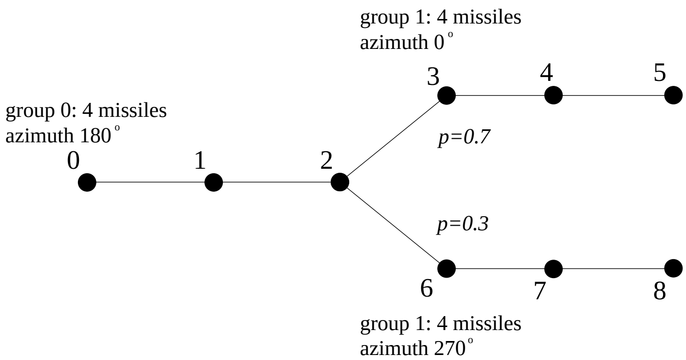
Battery No.	1	2	3	4
Range	80	45	45	45
Sector of fire	360°	0°–170°	80°–200°	160°–360°
Number of missiles—Variant 1	4	4	4	4
Number of missiles—Variant 2	4	8	8	8
Number of missiles—Variant 4	8	16	16	16

6.2. Results

The first set of experiments was carried out using the attack graph presented in Fig. 8(a). The simulations were repeated 30 times for five values of σ_γ being the standard deviation of the disturbance added to the azimuth prediction. As the mean value was equal to zero, $\sigma_\gamma = 0$ represents the situation of a perfect prediction. The number of missiles observed ($w_k^{j,i}$) was also modified by adding a normally-distributed random variable with zero mean and a standard deviation of 0.6. The number of hits by the enemy objects was chosen as the most objective criterion—it is the aim of the system to protect the defended object against destruction. Multiple hits are very dangerous as they increase exponentially the probability of the target destruction—they were shown separately. The results for periodic and initial coordination are included in Tables 2–4.



(a)



(b)

Fig. 8. Two variants of the attack.

Table 2. Results for the first variant of batteries (cf. Fig. 8(a)).

	Periodic coordination					Initial coordination				
	σ_γ					σ_γ				
	0°	45°	90°	135°	180°	0°	45°	90°	135°	180°
# of hits by 1 object	7	7	9	9	4	5	2	3	1	1
# of hits by 2 objects	10	3	9	3	7	7	8	7	7	4
# of hits by 3 objects	6	10	2	5	7	12	6	7	9	7
# of hits by more than 3 objects	0	3	1	2	5	4	13	12	12	17
Maximum # of hits	3	5	4	5	8	4	7	7	7	7
Maximum # of missiles launched	16	14	16	16	16	12	12	12	12	12

Table 3. Results for the second variant of batteries (cf. Fig. 8(a)).

	Periodic coordination					Initial coordination				
	σ_γ					σ_γ				
	0°	45°	90°	135°	180°	0°	45°	90°	135°	180°
# of hits by 1 object	5	8	4	3	5	14	5	4	6	7
# of hits by 2 objects	1	0	0	0	0	4	8	3	7	4
# of hits by 3 objects	0	0	0	1	2	0	2	1	3	2
# of hits by more than 3 objects	0	0	0	1	1	0	3	8	5	10
Maximum # of hits	2	1	1	4	4	2	5	5	6	6
Maximum # of missiles launched	20	20	22	22	22	18	18	16	18	16

Table 4. Results for the third variant of batteries (cf. Fig. 8(a)).

	Periodic coordination					Initial coordination				
	σ_γ					σ_γ				
	0°	45°	90°	135°	180°	0°	45°	90°	135°	180°
# of hits by 1 object	0	0	0	0	0	14	9	8	6	9
# of hits by 2 objects	0	0	0	0	0	3	1	1	5	5
# of hits by more than 2 objects	0	0	0	0	0	0	3	3	6	7
Maximum # of hits	0	0	0	0	0	2	5	5	5	5
Maximum # of missiles launched	18	18	18	18	18	17	18	16	17	17

The analysis of the results for periodic coordination allows us to find out a suitable size of batteries. Although in the first case (Table 2) all or nearly all SAM missiles were launched, it is likely that the defended object will be seriously damaged as a large part of the attacking objects (once even all) avoided a counter-fire. The second variant provided much better results and not all missiles were launched, even though not all enemy objects were destroyed either. This suggests that the structure of the batteries (i.e. the balance between long- and medium-range defences) is not correct. In the last variant, due to an increasing number of missiles in Battery 1, it is possible to attain success without using all medium-range missiles. The example shows the importance of providing a long-range defence as it gives the time necessary to asses the results of the first SAM salvo and then to repeat the actions if needed.

The initial coordination does not seem suitable in such scenarios, as reallocating batteries allows sectors to adapt and to launch more missiles and so to destroy more enemy objects. Stiff allocations calculated at the beginning of the attack are relatively effective in the case of perfect prediction of the direction of attacking objects ($\sigma_\gamma = 0$), but they cannot provide protection even when groups of objects slightly change their route. It is obvious that larger batteries can help to solve this problem, but comparison of the last two variants (Tables 3 and 4) reveals that it is very ineffecient—doubling the number of missiles allows us to reduce the maximum number of hits from 6 to 5 in the last two cases and results in no change in the remaining simulations.

The results obtained for the second attack scenario (Fig. 8(b)) do not differ much from the previous case. This can be treated as an argument for using periodic coordination as its abilities to adapt to various conditions are evident (especially Fig. 8(a) needs such an adaptation, as two groups are present at the same time). The scenario was chosen as less demanding owing to the separation of two stages of the attack in time, but it is also important to note that as the second group arrives after the time necessary for the first group to reach its targets (which are the weapons of the system), some weapon can be damaged and the defence gets weakened. For Fig. 8(a) damage may occur at the end of this scenario and does not deteriorate so much the defence abilities of sectors (e.g. it is not important for the long-range weapon which turned out to be crucial for an effective defence). The results are briefly presented in Tables 5 and 6.

The computations were performed on a PC computer with AMD K6 350 MHz processor running a Linux operating system. To compare the effectiveness of the coordination algorithm, the times of single simulations (i.e. a full course of control) are provided in Table 7. It is important to note that Fig. 8(b) contains more nodes, so the resulting problem is more complex. Although the computation times seem reasonable for off-line simulation purposes, note that such simulations are usually repeated, e.g. the required times were 2 hours 45 minutes and 8 hours 50 minutes for the graphs of Figs. 8(a) and 8(b), respectively, to collect the data presented in this paper. For an on-line decision taking such time is much too long, as the first stage of the simulated attack scenarios (i.e. the time from the detection of enemy objects to

Table 5. Maximum number of hits for Fig. 8(b).

	Periodic coordination					Initial coordination				
	σ_γ					σ_γ				
Variant of batteries	0°	45°	90°	135°	180°	0°	45°	90°	135°	180°
1	3	4	3	5	5	3	6	6	6	6
2	1	3	3	3	3	3	6	6	5	8
3	0	0	0	0	0	2	2	5	4	6

Table 6. Maximum number of missiles launched for Fig. 8(b).

	Periodic coordination					Initial coordination				
	σ_γ					σ_γ				
Variant of batteries	0°	45°	90°	135°	180°	0°	45°	90°	135°	180°
1	14	16	16	16	16	12	12	12	12	12
2	18	18	18	18	18	16	16	16	16	16
3	18	18	18	18	18	16	18	18	18	16

hit by the first salvo) lasts only 32 seconds, but it is important to note that more general methods are much slower. The results presented in (Arabas *et al.*, 1999) indicate that only a simplified linear model can be used in a hierarchical scheme. A more precise model involves solving a lower-level optimization problem with dynamic programming which was (depending on the problem complexity) several times to several hundred times slower than linear programming (a typical result was 30 minutes for dynamic programming and 2 seconds for linear programming). As the lower-level problem must be solved several hundred times during the coordination, it should be solved in less than one second, which is attained in the presented program owing to a further optimization of the proposed algorithm.

Table 7. Times of a single simulation (in seconds).

	Graph (a)		Graph (b)	
	Coordination		Coordination	
	periodic	initial	periodic	initial
Stage 1	39	39	89	89
Stage 2	10	< 1	58	< 1
Stage 3	9	< 1	40	< 1
Stage 4	7	< 1	10	< 1
Stage 5	—	—	9	< 1
Stage 5	—	—	5	< 1
Total	66	40	212	89

7. Conclusions

A hierarchical system for decision support in sectored anti-missile defence systems was proposed together with simulation results. The main part of the work was concentrated on the design of algorithms being efficient enough for practical use. For that purpose, some simplifications were necessary, related mainly to linearization and modelling damage by casualty functions instead of a full stochastic description. As a result, using the direct method of coordination with a simple, but reliable algorithm of finding, the allocation of batteries and tasks was possible. The advantage of this approach is not only a reliability higher than in the case of a centralized system, but also the better modelling of the information flow in a real system.

Although remarkable effectiveness was attained, there are still several possible ways of improving the performance of this algorithm. The simplest of these, but suitable only for large systems and relatively uncomplicated scenarios of the attack, consists in using only an initial coordination—such an approach was presented in (Arabas and Malinowski, 1999). The disadvantage is not only a suboptimal nature of solutions, but also a long time necessary to calculate the decisions for the first stage, which can be a serious obstacle in the case of on-line commands and control. Another possibility is to optimize the software, especially by using a better implementation of the branch-and-bound algorithm applied at the lower level. It is likely that such an implementation can, e.g. by using better heuristics, speed up the algorithm several times. Another way is parallelization of the computing tasks, which is possible due to the construction of the coordinator problem.

Acknowledgments

The work reported in this paper was supported by the State Committee for Scientific Research under grant No. 8 T11A 011 15.

References

Arabas P. and Malinowski K. (1999): *Hierarchical system for anti-missile defence.* — Proc. *Advanced Simulation of Systems Conf., ASIS'99*, Krnov, pp.289–294.

Arabas P., Warchoł M. and Malinowski K. (1999): *Model of sectored air defence system.* — Proc. 13th Europ. *Simul. Multiconf.*, Warsaw, pp.599–604.

Ardema M.D., Heymann M. and Rajan N. (1985): *Combat games.* — J. Optim. Theory Appl., Vol.4, No.44, pp.391–398.

Lenox D. (Ed.) (1998): *AA Missiles Systems.* — Jane's Strategic Weapon Systems, No.27.

Hill J.R. (1988): *Air Defense at Sea.* — London: Ian Allan.

Kimbleton S.R. (1969): *Attrition rates for weapons with Markov-dependent fire.* — Oper. Res., Vol.2, No.17, pp.698–705.

Parkhideh Sh. and Gafarian A.V. (1996): *General solution to many-on-many heterogeneous stochastic combat.* — Naval Res. Logist., Vol.4, No.43, pp.937–953.

Piasecki S. (1968): *Elements of Destruction Theory.* — Warsaw: Military University of Technology (in Polish).

Piasecki S. and Boratyn W. (1968): *Processes of Fight Control.* — Warsaw: Military University of Technology (in Polish).

Przemieniecki J.S. (1994): *Mathematical Methods in Defence Analyses.* — Washington: Amer. Inst. of Aeronautics and Astronautics, Inc.

Warchoł M. and Malinowski K. (1993): *Algorithm for solution of optimization problems under multiple scenarios of uncontrolled outputs.* — Proc. 16th IFIP Conf. *System Modelling and Optimization*, Compiegne, pp.239–242.

Warchoł M. and Malinowski K. (1995): *Optimization of system with nondifferentiable performance index under multiple scenarios of uncontrolled inputs.* — Proc. 17th IFIP Conf. *System Modelling and Optimization*, Prague, pp.566–569.

Zdrodowski B. (1998): *Air Defence Fundamentals.* — Warsaw: Academy of National Defence (in Polish).

Zdrodowski B. (1999): *Anti-Rocket Defence.* — Warsaw: Military University of Technology (in Polish).

Received: 23 August 2000
Revised: 14 December 2000