

Automatization of Decision Processes in Conflict Situations: Modelling, Simulation and Optimization

Zbigniew Tarapata

*Military University of Technology in Warsaw, Faculty of Cybernetics
Poland*

1. Introduction

Military conflict is one of the types of conflict situations. The automation of simulated battlefield is a domain of Computer Generated Forces (CGF) systems or semi-automated forces (SAF or SAFOR) (Henninger et al., 2000; Lee & Fishwick, 1995; Longtin & Megherbi, 1995; Lee, 1996; Mohn, 1994; Petty, 1995). CGF or SAF (SAFOR) is a technique, which provides a simulated opponent using a computer system that generates and controls multiple simulation entities using software and possibly a human operator. In the case of Distributed Interactive Simulation (DIS) systems, the system is intended to provide a simulated battlefield which is used for training military personnel. The advantages of CGF are well-known (Petty, 1995): they lower the cost of a DIS system by reducing the number of standard simulators that must be purchased and maintained; CGF can be programmed, in theory, to behave according to the tactical doctrine of any desired opposing force, and so eliminate the need to train and retrain human operators to behave like the current enemy; CGF can be easier to control by a single person than an opposing force made up of many human operators and it may give the training instructor greater control over the training experience. One of the elements of the CGF systems is module for movement planning and simulation of military objects. In many of existing simulation systems there are different solutions regarding to this subject. In the *JTLS* system (JTLS, 1988) terrain is represented using hexagons with sizes ranging from 1km to 16km. In the *CBS* system (Corps Battle Simulation, 2001) terrain is similarly represented, but vectoral-region approach is additionally applied. In both of these systems there are manual and automatic methods for route planning (e.g. in the *CBS* controller sets intermediate points (coordinates) for route). In the *ModSAF* (*Modular Semi-Automated Forces*) system in module "SAFsim", which simulates the entities, units, and environmental processes the route planning component is located (Longtin & Megherbi, 1995). In the paper (Mohn, 1994) implementation of a *Tactical Mission Planner* for command and control of Computer Generated Forces in *ModSAF* is presented. In the work (Benton et al., 1995) authors describe a combined on-road/off-road planning system that was closely integrated with a geographic information system and a simulation system. Routes can be planned for either single columns or multiple columns. For multiple columns, the planner keeps track of the temporal location of each column and insures they will not occupy the same space at the same time. In the same paper the *Hierarchic Route*

Planner as integrate part of *Predictive Intelligence Military Tactical Analysis System (PIMTAS)* is discussed. In the paper (James et al., 1999) authors presented on-going efforts to develop a prototype for ground operations planning, the *Route Planning Uncertainty Manager (RPLUM)* tool kit. They are applying uncertainty management to terrain analysis and route planning since this activity supports the Commander's scheme of manoeuvre from the highest command level down to the level of each combat vehicle in every subordinate command. They extend the *PIMTAS* route planning software to accommodate results of reasoning about multiple categories of uncertainty. Authors of the paper (Campbell et al., 1995) presented route planning in the *Close Combat Tactical Trainer (CCTT)*. Authors (Kreitzberg et al., 1990) have developed the *Tactical Movement Analyzer (TMA)*. The system uses a combination of digitized maps, satellite images, vehicle type and weather data to compute the traversal time across a grid cell. *TMA* can compute optimum paths that combine both on-road and off-road mobility, and with weather conditions used to modify the grid cost factors. The smallest grid size used is approximately 0.5 km. The author uses the concept of a signal propagating from the starting point and uses the traversal time at each cell in the array to determine the time at which the signal arrives to neighbouring cells. In the paper (Tarapata, 2004a) models and methods of movement planning and simulation in some simulation aided system for operational training on the corps-brigade level (Najgebauer, 2004) is described. A combined on-road/off-road planning system that is closely integrated with a geographic information system and a simulation system is considered. A dual model of the terrain ((1) as a regular network of terrain squares with square size 200mx200m, (2) as a road-railroad network), which is based at the digital map, is presented. Regardless of types of military actions military objects are moved according to some group (arrangement of units). For example, each object being moved in group (e.g. during attack, during redeployment) must keep distances between each other of the group (Tarapata, 2001). Therefore, it is important to recognize (during movement simulation) that objects inside units do not "keep" required distances (group pattern) and determine a new movement schedule. All of the systems presented above have no automatic procedures for synchronization movement of more than one unit. The common solution of this problem is when movement (and simulation, naturally) is stopped and commanders (trainees) make a new decision or the system does not react to such a situation. Therefore, in the paper (Tarapata, 2005) a proposition of a solution to the problem of synchronization movement of many units is shown. Some models of synchronous movement and the idea of module for movement synchronization are presented. In the papers (Antkiewicz et al., 2007; Tarapata, 2007c) the idea and model of command and control process applied for the decision automata on the battalion level for three types of unit tasks: attack, defence and march are presented.

The chapter is organized as follows. Presented in section 2 is the review of methods of environment modelling for simulated battlefield. An example of terrain model being used in the real simulator is described. Moreover, paths planning algorithms, which are being applied in terrain-based simulation, are considered. Sections 3 and 4 contain description of automatization methods of main battlefield processes (attack, defence and march) in simulation system like CGF. In these sections, a decision automata, which is a component of the simulation system for military training is described as an example. Presented in section 5 are some conclusions concerning problems and proposition of their solution in automatization of decision processes in conflict situations.

2. Environment modelling for simulation of conflict situations

2.1 An overview

The terrain database-based model is being used as an integrated part of route CGF systems. Terrain data can be as simple as an array of elevations (which provides only a limited means to estimate mobility) or as complex as an elevation array combined with digital map overlays of slope, soil, vegetation, drainage, obstacles, transportation (roads, etc.) and the quantity of recent weather. For example, in (Benton et al., 1995) authors describe HERMES (Heterogeneous Reasoning and Mediator Environment System) will allow the answering of queries that require the interrogation of multiple databases in order to determine the start and destination parameters for the route planner.

There are a few approaches in which the map (representing a terrain area) is decomposed into a graph. All of them first convert the map into regions of go (open) and no-go (closed). The no-go areas may include obstacles and are represented as polygons. A few methods of map representation is used, for example: visibility diagram, Voronoi diagram, straight-line dual of the Voronoi diagram, edge-dual graph, line-thinned skeleton, regular grid of squares, grid of homogeneous squares coded in a quadtree system, etc. (Benton et al., 1995; Schiavone et al., 1995a; Schiavone et al., 1995b; Tarapata, 2003).

The polygonal representations of the terrain are often created in database generated systems (DBGS) through a combination of automated and manual processes (Schiavone et al., 1995; Schiavone et al., 2000). It is important to say that these processes are computationally complicated, but are conducted before simulation (during preparation process). Typically, an initial polygonal representation is created from the digital terrain elevation data through the use of an automated triangulation algorithm, resulting in what is commonly referred to as a Triangulated Irregular Network (TIN). A commonly used triangulation algorithm is the Delaunay triangulation. Definition of the Delaunay triangulation may be done via its direct relation to the Voronoi diagram of set S with an N number of 2D points: the straight-line dual of the Voronoi diagram is a triangulation of S .

The *Voronoi diagram* is the solution to the following problem: given set S with an N number of points in the plane, for each point p_i in S what is the locus of points (x,y) in the plane that are closer to p_i than to any other point of S ?

The *straight-line dual* is defined as the graph embedded in the plane obtained by adding a straight-line segment between each pair of points of S whose Voronoi polygons share an edge. Fig.1a depicts an irregularly spaced set of points S , its Voronoi diagram, and its straight-line dual (i.e. its Delaunay triangulation).

The *edge-dual graph* is essentially an adjacency list representing the spatial structure of the map. To create this graph, we assign a node to the midpoint of each map edge, which does not bound an obstacle (or the border). Special nodes are assigned to the start and goal points. In each non-obstacle region, we add arcs to connect all nodes at the midpoints of the edges, which bound the same region. The fact that all regions are convex, guarantees that all such arcs cannot intersect obstacles or other regions. An example of the edge-dual graph is presented in Fig.1b.

The *visibility graph*, is a graph, whose nodes are the vertices of terrain polygons and edges join pairs of nodes, for which the corresponding segment lies inside a polygon. An example is shown in Fig.2.

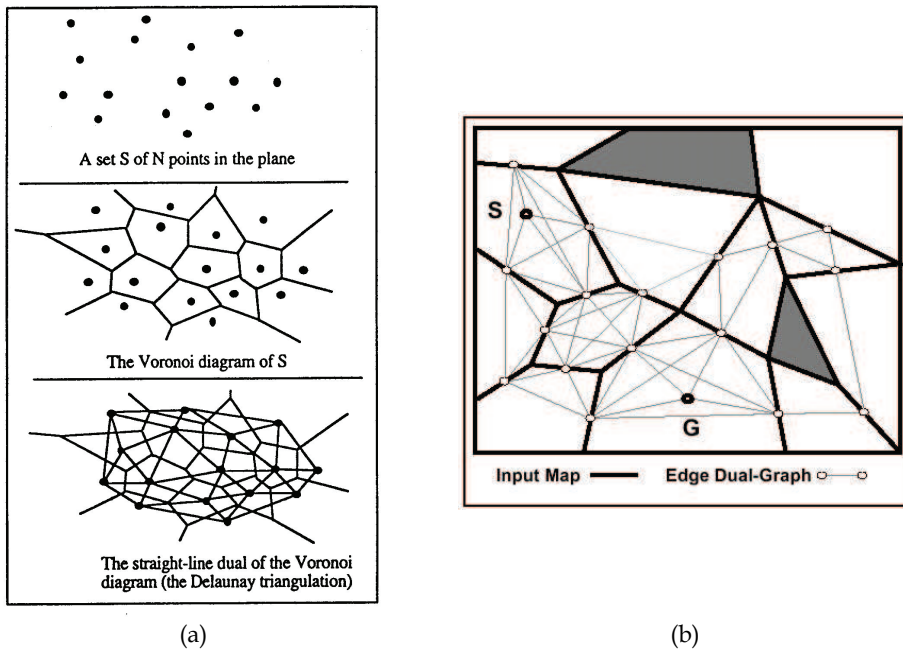


Fig.1. (a) Voronoi diagram and its Delaunay triangulation (Schiavone et al., 1995); (b) Edge-dual graph. Obstacles are represented by filled polygons

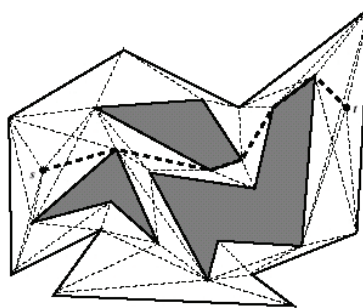


Fig.2. Visibility graph (Mitchell, 1999). The shortest geometric path is marked from source node s to destination t . Obstacles are represented by filled polygons

The *regular grid of squares* (or hexagons, e.g. in JTLS system (JTLS, 1988)) divides terrain space into the squares with the same size and each square is treated as having homogeneity from the point of view of terrain characteristics (Fig.3).

The *grid of homogeneous squares coded in quadtree system* divides terrain space into the squares with heterogeneous size (Fig.4). The size of square results from its homogeneity according to terrain characteristics. An example of this approach was presented in (Tarapata, 2000).

Advantages and disadvantages of terrain representations and their usage for terrain-based movement planning are presented in section 2.3.

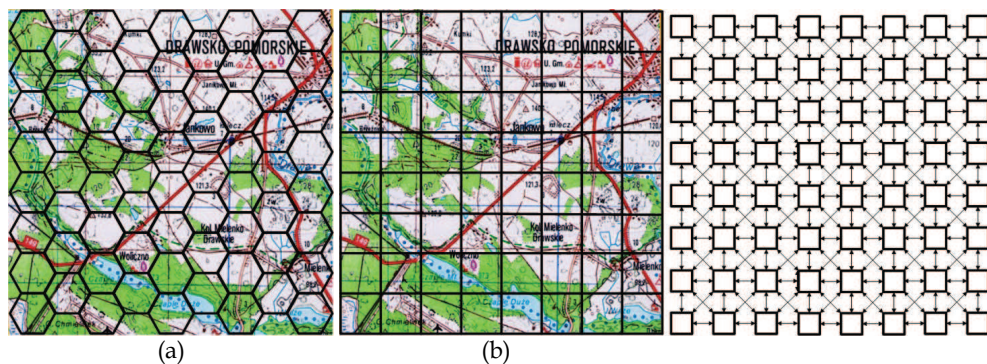


Fig.3. Examples of terrain representation in a simulated battlefield: (a) regular grid of terrain hexagons; (b) regular grid of terrain squares and its graph representation.

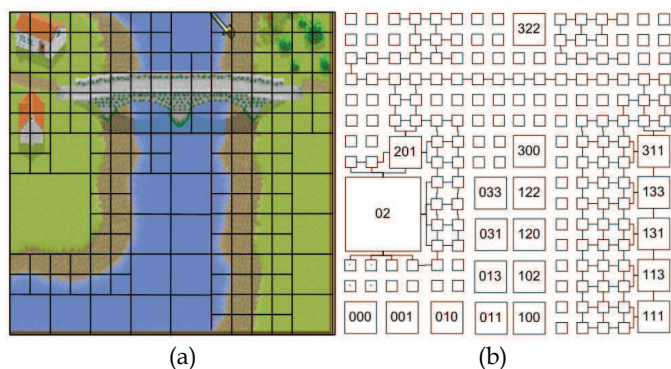


Fig.4. (a) Partitioning of the selected real terrain area into squares of topographical homogeneous areas; (b) Determination of possible links between neighbouring squares and a description of selected vertices in the quadtree system for terrain area presented in (a)

In many existing simulation systems there are different solutions regarding terrain representation. In the *JTLS* system (JTLS, 1988) terrain is represented using hexagons with a size ranging from 1km to 16km. In the *CBS* system (Corps Battle Simulation, 2001) terrain is similarly represented, but an additional vectoral-region approach is applied. In the simulation-based operational training support system “Zlocien” (Najgebauer, 2004) a dual model of the terrain: (1) as regular network of terrain squares with square size 200mx200m, (2) as road-railroad network, which is based on a digital map, is used.

Taking into account multiresolution terrain modelling (Behnke, 2003; Cassandras et al., 2000; Davis et al., 2000; Pai & Reissell, 1994; Tarapata, 2001) the approach is also used for battlefield modelling and simulation. For example, in the paper (Tarapata, 2004b) a decomposition method, and its properties, which decreases computational time for path searching in multiresolution graphs has been presented. The goal of the method is not only computation time reduction but, first of all, using it for multiresolution path planning (to apply similarity in decision processes on different command level and decomposing-merging approach). The method differs from very effective representations of terrain using

quadtree (Kambhampati & Davis, 1986) because of two main reasons: (1) elements of quadtree which represent a terrain have irregular sizes, (2) in majority applications quadtree represents only binary terrain with two types of region: open (passable) and closed (impassable). Hence, this approach is very effective for mobile robots, but it is not adequate, for example, to represent battlefield environment (Tarapata, 2003).

2.2 Terrain model for a battlefield simulation – an example

The terrain (environment) model S_0 , which we use as a battlefield model for further discussions (sections: 3.4 and 4) is based on the digital map in VPF format. The model is twofold: (1) as a regular network Z_1 of terrain squares, (2) as a road-railroad network Z_2 and it is defined as follows (Tarapata, 2004a):

$$S_0(t) = \langle Z_1(t), Z_2(t) \rangle \quad (1)$$

Regular grid of squares Z_1 (see Fig.3) divides terrain space into squares with the same size (200m×200m) and each square is homogeneous from the point of view of terrain characteristics (degree of slowing down velocity, ability to camouflage, degree of visibility, etc.). This square size results from the fact that the nearest level of modelled units in SBOTSS “Zlocien” (Najgebauer, 2004) is a platoon and 200m is approximately the width of the platoon front during attack. The Z_1 model is used to plan off-road (cross-country) movement e.g. during attack planning. In the Z_2 road-railroad network (see Fig.5) we have crossroads as network nodes and section of the roads linking adjacent crossroads as network links (arcs, edges). This model is used to plan fast on-road movement, e.g. during march (redeployment) planning and simulation.

These two models of terrain are integrated. This integration gives possibilities to plan movement inside both models. It is possible, because each square of terrain contains information about fragments of road inside this square. On the other hand each fragment of road contains information on squares of terrain, which they cross. Hence, route for any object (unit) may consist of sections of roads and squares of terrain. It is possible to get off the road (if it is impassable) and start movement off-road (e.g. omit impassable section of road) and next returning to the road. Conversely, we can move off-roads (e.g. during attack), access a section of road (e.g. any bridge to go across the river) and then return back off-road (on the other riverside). The characteristics of both terrain models depend on: time, terrain surface and vegetation, weather, the day and time of year, opponent and own destructions (e.g. destruction of the bridge which is element of road-railroad network) (see Table 1 and Table 2).

The formal definition of the regular network of terrain squares Z_1 is as follows (see Fig.3):

$$Z_1(t) = \langle G_1, \Psi_1(t) \rangle \quad (2)$$

where G_1 defines Berge's graph defining structure of squares network, $G_1 = \langle W_1, \Gamma_1 \rangle$, W_1 - set of graph's nodes (terrain squares); $\Gamma_1 : W_1 \rightarrow 2^{W_1}$ - function describing for each nodes of G set of adjacent nodes (maximal 8 adjacent nodes); $\Psi_1(t) = \{\Psi_{1,0}(\cdot, t), \Psi_{1,1}(\cdot, t), \Psi_{1,2}(\cdot, t), \dots, \Psi_{1,LW_1}(\cdot, t)\}$ - set of functions defined on the graph's nodes (depending on t).

One of the functions of $\Psi_1(t)$ is the function of slowing down velocity $FSDV(n, \dots)$, $n \in W_1$ which describes slowing down velocity (as a real number from [0,1]) inside the n -th square of the terrain,

$$FSDV: W_1 \times T \times K_Veh \times K_Meteo \times K_YearS \times K_DayS \rightarrow [0,1] \quad (3)$$

where: T – set of times, K_Veh – set of vehicle types, $K_Veh = \{Veh_Wheeled, Veh_Wheeled-Caterpillar, Veh_Caterpillar\}$; K_Meteo – set of meteorological conditions, K_YearS – set of the seasons of year, K_DayS – set of the day of the season.

The function $FSDV$ is used to calculate crossing time between two squares of terrain. Other functions (as subset of $\Psi_1(t)$) described on the nodes (squares) of G_1 and essential from the point of view of trafficability and movement are presented in the Table 1.

Description of the function	Definition of the function
Geographical coordinates of node (centre of square)	$FWSP: W_1 \rightarrow R^3$
Ability to camouflage in the square	$FCam: W_1 \times T \rightarrow [0,1]$
Degree of terrain undulation in the square	$FUnd: W_1 \rightarrow [0,1]$
Subset of node's set of Z_2 network, which are located inside the square	$FW1OnW2: W_1 \rightarrow 2^{W_2}$

Table 1. The most important functions described on the terrain square (node of G_1)

Formal definition of the road-railroad network Z_2 is following (see Fig.5):

$$Z_2(t) = \langle G_2, \Psi_2(t), \zeta_2(t) \rangle \quad (4)$$

where G_2 describes Berge's graph defining structure of road-railroad network, $G_2 = \langle W_2, U_2 \rangle$,

W_2 – set of graph's nodes (crossroads); $U_2 \subset W_2 \times W_2$ – set of graph G_2 arcs (sections of roads);

$\Psi_2(t) = \{\Psi_{2,0}(\cdot, t), \Psi_{2,1}(\cdot, t), \dots, \Psi_{2,LW_2}(\cdot, t)\}$ – set of functions defined on the graph's G_2 nodes (depending on t); $\zeta_2(t) = \{\zeta_{2,i}(\cdot, t)\}_{i=1, \overline{UG_2}}$ – set of functions defined on the graph's G_2 arcs (depending on t).

Functions (as subset of $\Psi_2(t)$ and $\zeta_2(t)$) are presented, which are essential from the point of view of trafficability and movement, described on the nodes and arcs of G_2 in the Table 2. One of the most important functions is slowing down velocity function $FSDV2(u, \dots)$, $u \in U_2$ which describes slowing down velocity (as real number from $[0,1]$) on the u -th arc (section of road) of the graph:

$$FSDV2: U_2 \times T \times K_Veh \times K_Meteo \times K_YearS \times K_DayS \rightarrow [0,1] \quad (5)$$

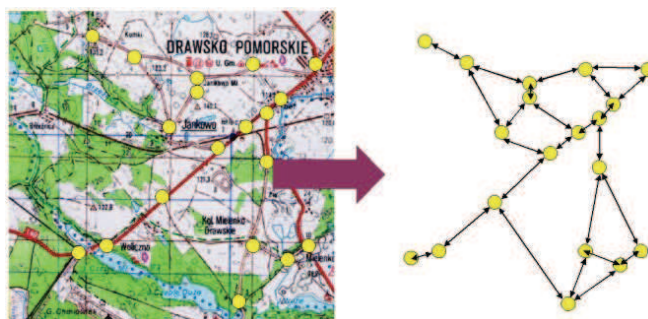


Fig.5. Road-railroad network (left-hand side) and its graph model G_2 (right-hand side)

Description of the function	Definition of the function
Geographical coordinates of node (crossroad)	$FWSP2 : W_2 \rightarrow R^3$
Node Z_1 , which contains node Z_2	$FW2OnW1 : W_2 \rightarrow W_1$
Subset of set of nodes of the Z_1 network, which contains the arc	$FU2OnW1 : U_2 \rightarrow 2^{W_1}$
Degree of terrain undulation on the arc	$FUnd : U_2 \rightarrow [0,1]$
Arc length	$FLen : U_2 \rightarrow R^+$

Table 2. The most important functions described on the crossroads and on part of the roads (G_2)

2.3 Paths planning algorithms in terrain-based simulation

There are four main approaches that are used in a battlefield simulation (CGF systems) for paths planning (Karr et al., 1995): free space analysis, vertex graph analysis, potential fields and grid-based algorithms.

In the *free space approach*, only the space not blocked and occupied by obstacles is represented. For example, representing the centre of movement corridors with Voronoi diagrams (Schiavone et al., 1995) is a free space approach (see Fig.1). The advantage of Voronoi diagrams is that they have efficient representation. Disadvantages of Voronoi diagrams are as follows: they tend to generate unrealistic paths (paths derived from Voronoi diagrams follow the centre of corridors while paths derived from visibility graphs clip the edges of obstacles); the width and trafficability of corridors are typically ignored; distance is generally the only factor considered in choosing the optimal path.

In the *vertex graph approach*, only the endpoints (vertices) of possible path segments are represented (Mitchell, 1999). Advantages of this approach: it is suitable for spaces that have sufficient obstacles to determine the endpoints. Disadvantages are as follows: determining the vertices in "open" terrain is difficult; trafficability over the path segment is not represented; factors other than distance can not be included in evaluating possible routes.

In the *potential field approach*, the goal (destination) is represented as an "attractor", obstacles are represented by "repellers", and the vehicles are pulled toward the goal while being repelled from the obstacles. Disadvantages of this approach: the vehicles can be attracted into box canyons from which they can not escape; some elements of the terrain may simultaneously attract and repel.

In the *regular grid approach*, the grid overlays the terrain, terrain features are abstracted into the grid, and the grid rather than the terrain is analyzed. Advantages are as follows: analysis simplification. Disadvantages: "jagged" paths are produced because movement out of a grid cell is restricted to four (or eight) directions corresponding to the four (or eight) neighbouring cells; granularity (size of the grid cells) determines the accuracy of terrain representation.

Many route planners in the literature are based on the *off-line path planning algorithms*: a path for the object is determined before its movement. The following are exemplary algorithms of this approach: Dijkstra's shortest path algorithm, A* algorithm (Korf, 1999), geometric path planning algorithms (Mitchell, 1999) or its variants (Korf, 1999; Logan, 1997; Logan & Sloman, 1997; Rajput & Karr, 1994; Tarapata, 1999; 2001; 2003; 2004; Undeger et al., 2001). For example, A* has been used in a number of Computer Generated Forces systems as the

basis of their component planning, to plan road routes (Campbell et al., 1995), to avoid moving obstacles (Karr et al., 1995), to avoid static obstacles (Rajput & Karr, 1994) and to plan concealed routes (Longtin & Megherbi, 1995). Moreover, the multicriteria approach to the path determined in CGF systems is often used. Some results of selected multicriteria paths problem and analysis of the possibility to use them in CGF systems are described, e.g. in (Tarapata, 2007a). Very extensive discussion related to geometric shortest path planning algorithms was presented by Mitchell in (Mitchell, 1999) (references consist of 393 papers and handbooks). The geometric shortest path problem is defined as follows: given a collection of obstacles, find an Euclidean shortest obstacle-avoiding path between two given points. Mitchell considers the following problems: geodesic paths in a simple polygon; paths in a polygonal domain (searching the visibility graph, continuous Dijkstra's algorithm); shortest paths in other metrics (L_p metric, link distance, weighted region metric, minimum-time paths, curvature-constrained shortest paths, optimal motion of non-point robots, multiple criteria optimal paths, sailor's problem, maximum concealment path problem, minimum total turn problem, fuel-consuming problem, shortest paths problem in an arrangement); on-line algorithms and navigation without map; shortest paths in higher dimensions.

The basic idea of the *on-line path planning algorithms* (Korf, 1999), in general, is that the object is moved step-by-step from cell to cell using a heuristic method. This approach is borrowed from robots motion planning (Behnke, 2003; Kambhampati & Davis, 1986; LaValle, 2006; Logan & Sloman, 1997; Undeger et al., 2001). The decision about the next move (its direction, speed, etc.) depends on the current location of the object and environment status. Examples of on-line path planning algorithms (Korf, 1999): RTA* (Real-Time A*), LRTA* (Learning RTA*), RTEF (Real-Time Edge Follows), HLRTA*, eFALCONS. For example, the idea of RTEF (real-time edge follow) algorithm (Undeger et al., 2001) is to let the object eliminate closed directions (the directions that cannot reach the target point) in order to decide on which way to go (open directions). For instance, if the object has a chance to realize that moving north and east won't let him reach the goal state, then it will prefer going south or west. RTEF finds out these open and closed directions by decreasing the number of choices the object has. However, the on-line path planning approach has one basic disadvantage: in this approach using a few criterions simultaneously to find an optimal (or acceptable) path is difficult and it is rather impossible to estimate, the moment of reaching the destination in advance. Moreover, it does not guarantee finding optimal solutions and even suboptimal ones may significantly differ from acceptable.

3. Automatization of main battlefield decision processes

3.1 Introduction

In this section the idea and model of command and control process applied for the decision automata for attack and defence on the battalion level are considered. In section 4 we will complete the description of the automata for the third type of unit task – march. As it was written in section 1 these problems are very rarely discussed in the literature; however some ideas we can come across in (Dockery & Woodcock et al., 1993; Hoffman H. & Hoffman M., 2000). The decision automata being presented replaces battalion commanders in the simulator for military trainings and it executes two main processes (Antkiewicz et al., 2003; Antkiewicz et al., 2007): decision planning process and direct combat control. The decision planning process (DPP) contains three stages: the identification of a decision situation, the

generation of decision variants, the variants evaluation and the selection of the best variant, which satisfy the proposed criteria. The decision situation is classified according to the following factors: own task, expected actions of opposite forces, environmental conditions – terrain, weather, the day and season, current state of own and opposite forces in a sense of personnel and weapon systems. For this reason, we can define identification of the decision situation (the first stage of the DPP and the most interesting from the point of view of automatization process) as a multicriteria weighted graph similarity decision problem (MWGSP) (Tarapata, 2007b) and present it in sections 3.3 and 3.4 presenting them through a short overview of structural objects similarity (section 3.2). The remaining two stages of DPP (the variants evaluation and selecting the best variant) are described in detail in (Antkiewicz et al., 2003; Antkiewicz et al., 2007): for each class of decision situations a set of action plan templates for subordinate and support forces are generated. For example the proposed action plan contains (Antkiewicz et al., 2007): forces redeployment, regions of attack or defence, or manoeuvre routes, intensity of fire for different weapon systems, terms of supplying military materiel to combat forces by logistics units. In order to generate and evaluate possible variants the pre-simulation process based on some procedures: forces attrition procedure, slowing down rate of attack procedure, utilization of munitions and petrol procedure is used. In the evaluation process the following criteria: time and degree of task realization, own losses, utilization of munitions and petrol are applied.

3.2 Structural objects similarity – a short overview

Object similarity is an important issue in applications such as e.g. pattern recognition. Given a database of known objects and a pattern, the task is to retrieve one or several objects from the database that are similar to the pattern.

If graphs are used for object representation this problem turns into determining the similarity of graphs, which is generally referred to as graph matching. Standard concepts in graph matching include (Farin et al., 2003; Kriegel & Schonauer, 2003): graph isomorphism, subgraph isomorphism, graph homomorphism, maximum common subgraph, error-tolerant graph matching using graph edit distance (Bunke, 1997), graph's vertices similarity, histograms of the degree sequence of graphs. A large number of applications of graph matching have been described in the literature (Bunke, 2000; Kriegel & Schonauer, 2003; Robinson, 2004). One of the earliest applications was in the field of chemical structure analysis. More recently, graph matching has been applied to case-based reasoning, machine learning planning, semantic networks, conceptual graph, monitoring of computer networks, synonym extraction and web searching (Blondel et al., 2004; Kleinberg, 1999; Kriegel & Schonauer, 2003; Robinson, 2004; Senellart & Blondel, 2003). Numerous applications from the areas of pattern recognition and machine vision have been reported (Bunke, 2000; Champin & Solon, 2003; Melnik et al., 2002). They include recognition of graphical symbols, character recognition, shape analysis, three-dimensional object recognition, image and video indexing and others. It seems that structural similarity is not sufficient for similarity description between various objects. The arc in the graph gives only binary information concerning connection between two nodes. And what about, for example, the connection strength, connection probability or other characteristics? Thus, the weighted graph matching problem is defined, but in the literature it is relatively rarely considered (Almohamad et al., 1993; Champin & Solon, 2003; Tarapata, 2007b; Umeyama, 1988) and it is most often regarded as a special case of graph edit distance, which is a very time-complex measure

(Bunke, 2004; Kriegel & Schonauer, 2003). Therefore, in section 3.3 we will define a multicriteria weighted graph similarity decision problem (MWGSP) and we will show how to use it for pattern recognition (matching) of decision situations (PRDS) in decision automata, which replaces commanders in simulators for military trainings (Antkiewicz et al., 2007).

3.3 Definition of the multicriteria weighted graph similarity problem (MWGSP)

3.3.1 Structural and quantitative similarity measures between weighted graphs

Let us define weighted graph WG as follows:

$$WG = \left\langle G, \{f_i(n)\}_{i \in \{1, \dots, LF\}, n \in N_G}, \{h_j(a)\}_{j \in \{1, \dots, LH\}, a \in A_G} \right\rangle \quad (6)$$

where: G - Berge's graph, $G = \langle N_G, A_G \rangle$, N_G , A_G - sets of graph's nodes and arcs, $A_G \subset \{ \langle n, n' \rangle : n, n' \in N_G \}$, $f_i : N_G \rightarrow R^n$ - the i -th function described on the graph's nodes, $i = 1, \dots, LF$, (LF - number of node's functions); $h_j : A_G \rightarrow R^n$ - the j -th function described on the graph's arcs, $j = 1, \dots, LH$ (LH - number of arc's functions).

Let two weighted graphs G_A and G_B be given. We propose to calculate two types of similarities of the G_A and G_B : structural and non-structural (quantitative). To calculate structural similarity between G_A and G_B it is proposed to use approach defined in (Blondel et al., 2004). Let A and B be the transition matrices of G_A and G_B . We calculate following sequence of matrices:

$$Z_{k+1} = \frac{BZ_k A^T + A^T Z_k B}{\|BZ_k A^T + A^T Z_k B\|_F}, \quad k \geq 0 \quad (7)$$

where $Z_0 = \mathbf{1}$ (matrix with all elements equal 1); x^T - matrix x transposition; $\|x\|_F$ - Frobenius

(Euclidian) norm for matrix x , $\|x\|_F = \sqrt{\sum_{i=1}^{n_B} \sum_{j=1}^{n_A} x_{ij}^2}$, n_B - number of matrix rows (number of

nodes of G_B), n_A - number of matrix columns (number of nodes of G_A). Element z_{ij} of the matrix Z describes similarity score between the i -th node of the G_B and the j -th node of the G_A . The essence of the graph's nodes similarity is the fact that two graphs' nodes are similar if their neighbouring nodes are similar. The greater value of z_{ij} the greater the similarity between the i -th node of the G_B and the j -th node of the G_A . We obtain structural similarity matrix $S(G_A, G_B)$ between nodes of graphs G_A and G_B as follows (Blondel et al., 2004):

$$S(G_A, G_B) = [s_{ij}]_{n_B \times n_A} = \lim_{k \rightarrow +\infty} Z_{2k} \quad (8)$$

Some computation aspects of calculation $S(G_A, G_B)$ have been presented in (Blondel et al., 2004). We can write (7) more explicit by using the matrix-to-vector operator that develops a matrix into a vector by taking its columns one by one. This operator, denoted vec , satisfies the elementary property $vec(C \otimes D) = (D^T \otimes C^T) vec(X)$ in which $\otimes$ denotes the Kronecker product (also denoted tensorial, direct or categorical product). Then, we can write equality (7) as follows:

$$z_{k+1} = \frac{(A \otimes B + A^T \otimes B^T)z_k}{\|(A \otimes B + A^T \otimes B^T)z_k\|_F} \quad (9)$$

Unfortunately, the iteration z_{k+1} does not always converge. Authors of (Melnik et al., 2002) showed that if we change the formula (9) for $z_{k+1} = \frac{(A \otimes B + A^T \otimes B^T)z_k + b}{\|(A \otimes B + A^T \otimes B^T)z_k + b\|_F}$, then the formula (9) converges for $b > 0$. Having matrix $S(G_A, G_B)$, we can formulate and solve an optimal assignment problem (using e.g. Hungarian algorithm) to find the best allocation matrix $X = [x_{ij}]_{n_B \times n_A}$ of nodes from graph describing G_A, G_B :

$$d_S(G_A, G_B) = \sum_{i=1}^{n_B} \sum_{j=1}^{n_A} s_{ij} \cdot x_{ij} \rightarrow \max \quad (10)$$

with constraints:

$$\sum_{i=1}^{n_B} x_{ij} \leq 1, \quad j = \overline{1, n_A} \quad (11)$$

$$\sum_{j=1}^{n_A} x_{ij} \leq 1, \quad i = \overline{1, n_B} \quad (12)$$

$$\forall_{i \in \{1, \dots, n_B\}} \forall_{j \in \{1, \dots, n_A\}} x_{ij} \in \{0, 1\} \quad (13)$$

The $d_S(G_A, G_B)$ describes the value of *structural similarity measure* of G_A and G_B (Fig.6).

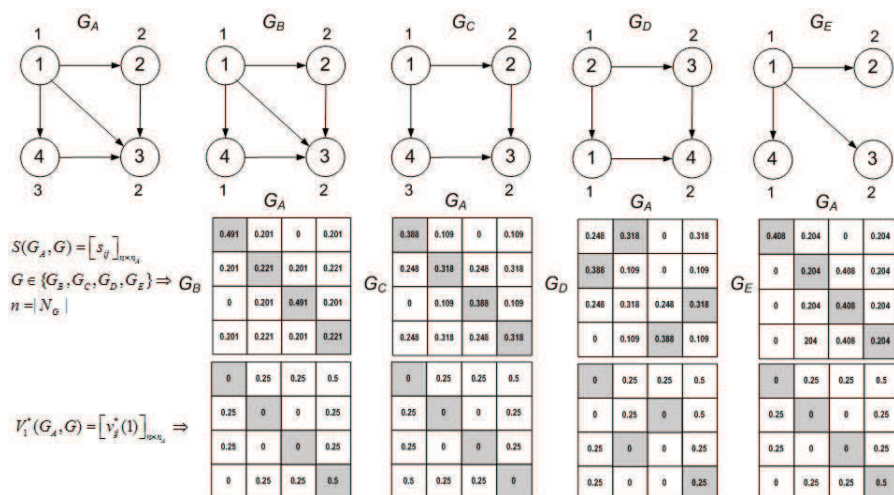


Fig.6. Examples of weighted graphs with a single function described on the nodes (set of functions described on the arcs is empty) and their structural ($S(G_A, G)$) and quantitative ($V_1^*(G_A, G)$) similarity matrices. Filled cells describe ones, which create optimal assignment the nodes of G_A to nodes of G .

To calculate non-structural (quantitative) similarity between G_A and G_B we should consider similarity between values of node's and arc's functions (*nodes and arcs quantitative similarity*). To compute nodes quantitative similarity we propose to create vector $\mathbf{v}(G_A, G_B) = \langle V_1, \dots, V_{LF} \rangle$ of matrices, where $V_k = [v_{ij}(k)]_{n_B \times n_A}$, $k=1, \dots, LF$, describing similarity matrix between nodes of G_A and G_B from the point of view of the k -th node's function ($f_k^A : N_{G_A} \rightarrow R^n$ for G_A and $f_k^B : N_{G_B} \rightarrow R^n$ for G_B) and $v_{ij}(k) = \|f_k^B(i) - f_k^A(j)\|$ describes "distance" between the i -th node of G_B and the j -th node of G_A from the point of view of f_k^B and f_k^A , respectively. We can apply a norm with parameter $p \geq 1$ as distance measure:

$$\|f_k^B(i) - f_k^A(j)\| = \|f_k^B(i) - f_k^A(j)\|_p = \left(\sum_{r=1}^n |f_{k,r}^B(i) - f_{k,r}^A(j)|^p \right)^{1/p} \quad (14)$$

where $f_{k,r}^A(\cdot)$, $f_{k,r}^B(\cdot)$ describe the r -th component of the vector being value of f_k^A and f_k^B , respectively. Next, we compute for each $k=1, \dots, LF$ normalized matrix $V_k^* = [v_{ij}^*(k)]_{n_B \times n_A}$, where $v_{ij}^*(k) = v_{ij}(k) / \|V_k\|_F$. This procedure guarantees that each $v_{ij}^*(k) \in [0, 1]$. Finally, we compute total quantitative similarity between the i -th node of G_B and the j -th node of G_A as follows:

$$\bar{v}_{ij} = \sum_{k=1}^{LF} \lambda_k \cdot v_{ij}^*(k), \quad \sum_{k=1}^{LF} \lambda_k = 1, \quad \forall_{k=1, \dots, LF} \lambda_k \in [0, 1] \quad (15)$$

The $d_{QN}(G_A, G_B)$ nodes quantitative similarity measure of G_A and G_B we compute solving assignment problem (10)-(12) substituting $\bar{v}_{ij}$ for s_{ij} (because of that the smaller value of $\bar{v}_{ij}$ the better) and $d_{QN}(G_A, G_B)$ for $d_S(G_A, G_B)$ in (10). Example of calculations similarity matrices between nodes of some graphs and similarity measures d_S and d_{QN} between graphs are presented in the Fig.6 and in the Table 3. Let us note that the best structural matched graph to G_A is G_B ($d_S(G_A, G_B)=1.423$ is the maximal value among of values of this measure for other graphs) but the best quantitative matched graph to G_A is G_C ($d_{QN}(G_A, G_C)=0$ is minimal value among of values of this measure for other graphs). Question is: which graph is the most similar to G_A : G_B or G_C ? Some method for solving the problem and to answer the question is presented in section 3.3.2: we have to apply multicriteria choice of the best matched graph to G_A . We can obtain arcs quantitative similarity measure $d_{QA}(G_A, G_B)$ by analogy to $d_{QN}(G_A, G_B)$: we build vector $\mathbf{e}(G_A, G_B) = \langle E_1, \dots, E_{LH} \rangle$ of matrices, where $E_k = [e_{ij}(k)]_{m_B \times m_A}$, $k=1, \dots, LH$ (m_A , m_B - number of arcs in G_A and G_B) describing similarity matrix between arcs of G_A and G_B from the point of view of the k -th arc's function ($h_k^A : A_{G_A} \rightarrow R^n$ for G_A and $h_k^B : A_{G_B} \rightarrow R^n$ for G_B),

$$e_{ij}(k) = \|h_k^B(i) - h_k^A(j)\|_p, \text{ next } e_{ij}^*(k) = e_{ij}(k) / \|E_k\|_F \text{ and } \bar{e}_{ij} = \sum_{k=1}^{LH} \mu_k \cdot e_{ij}^*(k), \quad \sum_{k=1}^{LH} \mu_k = 1, \quad \forall_{k=1, \dots, LH} \mu_k \geq 0.$$

Substituting in (10) $\bar{e}_{ij}$ for s_{ij} , $d_{QA}(G_A, G_B)$ for $d_S(G_A, G_B)$ and solving (10)-(12) we obtain $d_{QA}(G_A, G_B)$.

Graph G	$d_S(G_A, G)$	$d_{QN}(G_A, G)$	$0.5d_S(G_A, G) - 0.5d_{QN}(G_A, G)$
G_B	1.423	0.5	0.462
G_C	1.412	0	0.706
G_D	1.412	0.25	0.456
G_E	1.225	0.5	0.362

Table 3. Values of similarity measures between G_A and each of the four graphs from Fig.6

Let us note that it is possible to determine single quantitative similarity measure for G_A and G_B . To this end we use some transformation of graph $G = \langle N, A \rangle$ into temporary graph $G^* = \langle N^*, A^* \rangle$ as follows: $N^* = N \cup A$, $A^* \subset N^* \times N^*$ and

$$\forall_{v \in N, a \in A} \left(\bigvee_{x \in N} (\exists (v, x) = a \Rightarrow (v, a) \in A^*) \vee \right. \\ \left. (\exists_{x \in N} (x, v) = a \Rightarrow (a, v) \in A^*) \right) \quad (16)$$

If G was a weighted graph then in G^* we attribute the arc's and node's functions from G to appropriate nodes of G^* (that is to nodes and arcs from G). Using this procedure for G_A and G_B we obtain G_A^* and G_B^* . Next, for G_A^* and G_B^* we can calculate nodes quantitative similarity measure $d_{QN}(G_A^*, G_B^*)$. Example of constructing G^* from G is presented in the Fig.7.

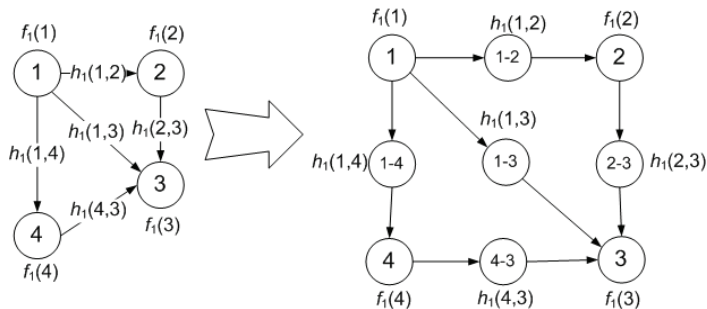


Fig.7. Transformation of G (left-hand side) into G^* (right-hand side)

3.3.2 Formulation of multicriteria weighted graphs similarity problem (MWGSP)

Let us accept $SG = \{G_1, G_2, \dots, G_M\}$ as a set of weighted graphs defining certain objects. Moreover, we have weighted graph P that defines a certain pattern object. The problem is to find such a graph G^o from SG that is the most similar to P . We define this problem as a multicriteria weighted graphs similarity problem (MWGSP), which is a multicriteria optimization problem in the space SG with relation R_D :

$$MWGSP = (SG, F, R_D) \quad (17)$$

where $F: SG \rightarrow R^3$, $F(G) = (d_S(P, G), d_{QN}(P, G), d_{QA}(P, G))$ and

$$R_D = \left\{ (Y, Z) \in SG \times SG : \begin{aligned} & d_S(P, Y) \geq d_S(P, Z) \wedge \\ & d_{QN}(P, Y) \leq d_{QN}(P, Z) \wedge \\ & d_{QA}(P, Y) \leq d_{QA}(P, Z) \end{aligned} \right\} \quad (18)$$

Domination relation R_D (Pareto relation between elements of SG) gives possibilities to compare graphs from SG . Weighted graph Z is more similar to P than Y if structural similarity between P and Y is not smaller than between P and Z and, simultaneously, both quantitative similarities between P and Y are not greater than between P and Z . There are many methods for solving the problem (17) (Eschenauer et al., 1990): weighted sum (scalarization of set of objectives), hierarchical optimization (the idea is to formulate a sequence of scalar optimization problems with respect to the individual objective functions subject to bounds on previously computed optimal values), trade-off method (one objective is selected by the user and the other ones are considered as constraints with respect to individual minima), method of distance functions in L_p -norm ($p \geq 1$) and others. We propose to use scalar function $H(G) : SG \rightarrow R$ as weighted sum of objectives:

$$\begin{aligned} H(G) &= \alpha_1 \cdot d_S(P, G) + \alpha_2 \cdot (-d_{QN}(P, G)) + \alpha_3 \cdot (-d_{QA}(P, G)) \\ \alpha_1, \alpha_2, \alpha_3 &\geq 0, \quad \alpha_1 + \alpha_2 + \alpha_3 = 1 \end{aligned} \quad (19)$$

Taking into account (19) the problem of finding the most matched G^o to pattern P can be formulated as follows: to determine such a $G^o \in SG$, that $H(G^o) = \max_{G \in SG} H(G)$. In the last column of the Table 3 the scalar function $H(G)$ is defined as follows:

$$H(G) = \alpha_1 \cdot d_S(P, G) + \alpha_2 \cdot (-d_{QN}(P, G)) + \alpha_3 \cdot (-d_{QA}(P, G)) \quad (20)$$

where $\alpha_1 = \alpha_2 = 0.5$, $\alpha_3 = 0$, $P = G_A$, $SG = \{G_B, G_C, G_D, G_E\}$. Let us note that the best matched graph to G_A being solution of MWGSP with scalar function $H(G)$ is G_C ($H(G^o = G_C) = 0.706$). In the paper (Tarapata, 2007b) epsilon-similarity of weighted graphs as another view on quantitative similarity between weighted graphs is additionally considered.

3.4 Application of weighted graphs similarity to pattern recognition of decision situations

For the identification of the decision situation described in section 3.1 we define decision situations space as follows:

$$DSS = \left\{ SD : SD = [SD_{ij}]_{i=1, \dots, X}^{j=1, \dots, Y} \right\} \quad (21)$$

where SD denotes net of terrain squares as a model of activities (interest) area $SD_{ij} = (SD_{ij}^k)_{k=1, \dots, 8}$. For the terrain square with the indices (i, j) each of elements denotes: SD_{ij}^1 - the degree of terrain passability, SD_{ij}^2 - the degree of forest covering, SD_{ij}^3 - the degree of water covering, SD_{ij}^4 - the degree of terrain undulating, SD_{ij}^5 - armoured power (potential) of opposite units deployed in the square, SD_{ij}^6 - infantry power (potential) of

opposite units deployed in the square, SD_{ij}^7 - artillery power (potential) of opposite units deployed in the square, SD_{ij}^8 - coordinates of square, X - the width of an activities (interest) area (number of squares), Y - the depth of an activities (interest) area (number of squares) and $SD_{ij}^k \in [0,1]$, $k = 1, \dots, 7$, $SD_{ij}^8 \in R_+^2$. Moreover, we have set $PDSS$ of decision situations patterns written in the database, $PDSS = \{PS : PS \in DSS\}$ and current situation $CS \in DSS$. The problem is: to find the most similar $PS \in PDSS$ to current situation $CS \in DSS$.

In the presented proposition the weighted graphs similarity approach to identification of decision situation is used. It consists of three stages:

1. Building weighted graphs $WGT(CS)$, $WGD(CS)$ and $WGT(PS)$, $WGD(PS)$ representing decision situations: current (CS) and pattern (PS) for topographical conditions (WGT) and units (potential) deploying (WGD);
2. Calculation of similarity measures between pairs: $WGT(CS)$, $WGT(PS)$ and $WGD(CS)$, $WGD(PS)$ for each $PS \in PDSS$;
3. Selecting the most similar PS to CS using calculated similarity measures.

Stage 1

The first stage is to build weighted graphs WGT and WGD as follows:

$$WGT = \left\langle GT = \langle N_{GT}, A_{GT} \rangle, \{f_k^T(n)\}_{\substack{k \in \{1, \dots, 5\} \\ n \in N_{GT}}} \right\rangle, \quad WGD = \left\langle GD = \langle N_{GD}, A_{GD} \rangle, \{f_k^D(n)\}_{\substack{k \in \{1, \dots, 4\} \\ n \in N_{GD}}} \right\rangle$$

where G (GT or GD) - Berge's graphs, $G = \langle N_G, A_G \rangle$, N_G, A_G - sets of graph's nodes and arcs, $A_G \subset \{\langle n, n' \rangle : n, n' \in N_G\}$. Weighted graphs WGT and WGD describe decision situations (current CS and pattern PS). Each node n of GT and GD describes terrain cells $(i,j)=n$ with non-zero values of characteristics defined as components of SD_{ij} from (21) and $\forall_{k \in \{1, \dots, 4\}} f_k^T(n) = SD_{ij}^k$, $f_5^T(n) = SD_{ij}^8$, $\forall_{k \in \{1, \dots, 3\}} f_k^D(n) = SD_{ij}^{4+k}$, $f_4^D(n) = SD_{ij}^8$. Two nodes $x, y \in N_{GD}$ (for $x, y \in N_{GT}$ by analogy) are linked by an arc, when cells represented by x and y are adjacent (more precisely: they are adjacent cells that taking into account the direction of action, see Fig.8). For example, the terrain can be divided into 15 cells (3 rows and 5 columns, left-hand side, see Fig.8). The units are located in some cells (denoted by circles and Xs). Structural representation of deployment of units is defined by the graph GD . Let us note that similar representation can be used for topographical conditions (single graph for one of the topographical information layer: waters, forests, passability or single graph GT for all of this information, see Fig.8, right-hand side).

Stage 2

Having weighted graphs $WGD(CS)$ and $WGD(PS)$ ($WGT(CS)$ and $WGT(PS)$) representing current CS and pattern PS decision situations (for units deploying) we use the procedure described in section 3.3.1 to calculate structural and quantitative similarity measures for both graphs. We obtain for WGD : $d_s(WGD(CS), WGD(PS)) = d_s^D(CS, PS)$, $d_{QN}(WGD(CS), WGD(PS)) = d_{QN}^D(CS, PS)$ and for WGT : $d_s(WGT(CS), WGT(PS)) = d_s^T(CS, PS)$, $d_{QN}(WGT(CS), WGT(PS)) = d_{QN}^T(CS, PS)$.

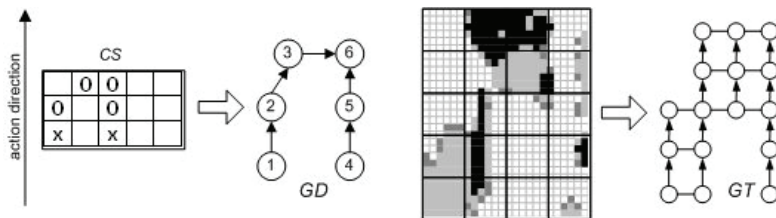


Fig.8. Deployment of units and their structural (graph GD) representation (left-hand side) and terrain covering (growth) and its structural (GT) representation (right-hand side). Circle (O) and sharp (X) describe two types of units

Stage 3

We formulate problem (17), separately for WGT and WGD , where: $SG := PDSS$, $F(G) := F_D(PS)$, $d_s(P, G) := d_s^D(CS, PS)$, $d_{QN}(P, G) := d_{QN}^D(CS, PS)$ for WGD and $F(G) := F_T(PS)$, $d_s(P, G) := d_s^T(CS, PS)$, $d_{QN}(P, G) := d_{QN}^T(CS, PS)$ for WGT . Next, we define scalar functions (19) to solve the problem (17) for WGD and WGT :

$$H_D(\cdot) = \alpha_1 \cdot d_s^D(\cdot, \cdot) + \alpha_2 \cdot (-d_{QN}^D(\cdot, \cdot))$$

and

$$H_T(\cdot) = \gamma_1 \cdot d_s^T(\cdot, \cdot) + \gamma_2 \cdot (-d_{QN}^T(\cdot, \cdot)).$$

Having $H_D(PS)$ and $H_T(PS)$ we can combine these criteria (like in (19)) or set some threshold values and select the most matched pattern situation to the current one.

An example of using the presented approach to find the most matched pattern decision situation to current one is presented in the Fig.9 and in the Table 4. Results of calculations $H_D(PS)$ are presented for each $PS \in PDSS = \{PS_1, \dots, PS_8\}$. Only function $f_4^{D(CS)}(n) = SD_{ij}^8$ ($f_4^{D(PS)}(n)$ for pattern PS) is used from WGD to compute nodes quantitative similarity (see section 3.3.1) because all units have the same type. Thus, vector $v(WGD(CS), WGD(PS))$ of matrices has one component $V_1 = [v_{ij}(1)]_{|N_{GD(PS)}| \times |N_{GD(CS)}|}$. Function $f_4^{D(CS)}(n)$ describes coordinates of node n (left-lower cell has coordinates (1,1)). The norm from (14) has the form

of: $\|f_4^D(i) - f_4^D(j)\|_{p=2} = \left(\sum_{r=1}^2 |f_{4,r}^D(i) - f_{4,r}^D(j)|^2 \right)^{1/2}$ and it describes the geometric distance

between nodes $i \in N_{GD(PS)}$ and $j \in N_{GD(CS)}$. Let us note that for weights $\alpha_1 = 0$, $\alpha_2 = 1$ value in Table 4 (for the row PS_i) describes $d_{QN}^D(CS, PS_i)$ and for $\alpha_1 = 1$, $\alpha_2 = 0$ describes $d_s^D(CS, PS_i)$. The best matched PS to CS is PS_2 (taking into account d_s^D and d_{QN}^D).

The process of optimal selection of weights can be organized as follows: we build a learning set $\{CS_i, PDSS_i\}_{i=1, \dots, LS}$ and for different values of weights experts estimate whether, in their subjective opinion, CS_i is similar to $PS^* \in PDSS_i$ determined from the procedure. Combination of weight values, which are indicated by majority of experts is the optimal combination.

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

