Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

GLIDING STRATEGY DESIGN WITH THE USE OF DISCRETE OPTIMIZATION

Zdzislaw Kowalczuk* Krzysztof E. Olinski*

* *Gdansk University of Technology, Dept. of Decision Systems* (*WETI*),
Narutowicza 11/12, 80-952 Gdansk, Poland,
e-mail: kova@eti.pg.gda.pl, kolin@eti.pg.gda.pl

Abstract: A glider flight strategy design problem can be effectively solved within the framework of optimal control. Optimality relates to glider characteristics and environmental conditions. One is often interested in determining an initial glider altitude position guaranteeing that a required destination point is achievable. Another way of posing such a control problem is linked with a maximal glider range resulting from given initial and environmental conditions. In fact, in practical applications, trying to determine a best flying strategy with the use of optimal approaches, we are usually faced with the problem of a large degree of freedom, which makes the classical analytical and numerical optimization methods ineffective. In this paper we introduce a simple method utilizing a search graph algorithm for the purpose of finding an optimal flight trajectory. We discuss the characteristics of this approach and present results of optimization performed for a glider manufactured by the PZL Swidnik. *Copyright@2008IFAC*

Keywords: Optimal control theory, aerospace applications, optimal control problems.

1. INTRODUCTION

The issue of optimal glider flight strategy is fairly universal when considering all the applications of optimization results for soaring competitions. In such cases our main goal is to minimize a flight time or to maximize a cross-country cruise speed. To achieve the goal a pilot has to choose a flight path using thermals (rising masses of air) when it is necessary to increase its cruise altitude. An important soaring parameter is a glide ratio, which should be chosen according to the tail winds, head winds and horizontal movements of the air. One more task concerns course corrections necessary to minimize the effect of drifts caused by crosswinds. The most familiar flight optimization method is ascribed to MacCready. It is so popular that almost every guidebook on soaring techniques contains a description of this theorem, see e.g. (FAA, 2003). However, the MacCready method is used to perform rather a local optimization of the flight because it answers the questions concerning the problem of selecting a glide ratio to assure a best cruise speed between thermals in the presence of tail/head and lift/sink winds. It is clear that global flight optimization requires a more complex approach.

The nature of the question makes it attractive to be applied as an instance for the usage of optimization algorithms solving problems with perfect or uncertain knowledge about atmospheric conditions. A concise survey of such approaches applied to optimal soaring has been given by Almgren and Tourin (2004). Nevertheless, most of the analytical methods (like Pontryiagn's, Bellman's, Hamilton-Jacobi-Bellman, or other methods based on the calculus of variation) cannot be directly applied to dynamical processes that are described by models containing hard nonlinearities (Lewis, 1992) characterized by lack of derivatives in some points of their functional domains. Another disadvantage of the analytical approach is the complexity of symbolic transformations necessary to be performed. Even though there are a variety of numerical algorithms, most of them have disadvantages lying in either local convergence (Bertsekas, 2000) or even numerical instability (Asher and Petrold, 1997), especially when dealing with the so-called *stiff* ODEs.

In this paper we concentrate on a deterministic problem with a well-defined, time-invariant atmospheric model. Such a restrictive assumption certainly constrains the potential applicability of the presented method. Nevertheless, it appears to be an interesting and useful search study. Our principal goal is to find a glider flight trajectory in a 3D work space that minimizes the total flight time between given initial and terminal points.

Our method is based on the following simple rule. First we construct a 3D mesh or graph image of the work space using a suitable set of nodes. The sought trajectory is represented as a piecewise linear path consisted of adjacent line segments (edges), which connect two 'neighboring' vertices, being nodes of the mesh. The optimal flight trajectory is determined by means of the Dijkstra algorithm (Siena, 1997), which treats nodes of the graph structure as potential vertices. The flow value between selected two nodes is proportional to the flight time between the two corresponding work points. The graph representation has an extra advantage of giving the possibility of defining certain zones forbidden for the considered flight. They can be applied to portray regions of dangerous atmospheric conditions or ground obstacles (like mountains), which should be omitted during the flight.

The paper is organized as follows. Section 2 describes a glider and an environment model. Section 3 is devoted to a formal description of our method. Results of three simulated cases are given in Section 4. The last section presents the paper conclusions.

2. MODEL DESCRIPTION

This section presents a model of the assumed shape of the ground as well as an atmospheric and a glider model. For the sake of clarity, the optimal flight trajectory will be sought within a restricted 3D work space of the size of (30 km × 10 km × 1 km) as depicted in Fig.1.

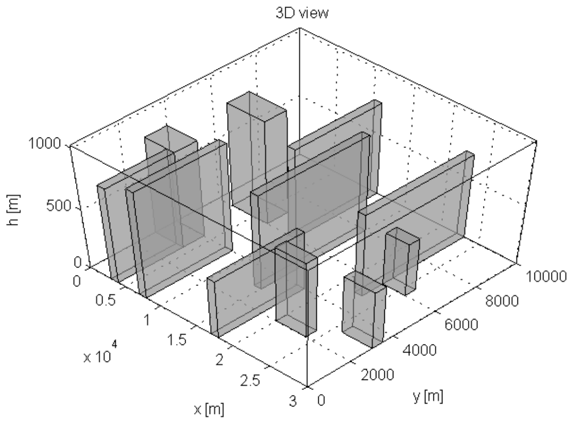

Fig. 1. Work space of the size of (30km × 10km × 1 km) with ground obstacles (gray boxes).

2.1 Ground shape model

As shown in Fig. 1, we assume that each ground obstacle is represented within the ground shape model by a cubic box.

2.2 Atmospheric model

We presuppose a pretty simple atmospheric model with vertically and horizontally moving masses of air. The horizontal winds are modeled by the following functions:

$$v_{wx}(x,y,h) = x/\alpha \cdot h \cdot 10^{-3}$$
$$v_{wy}(x,y,h) = \frac{-\cos(y) - \sin(x)(y - 5 \cdot 10^{3})h}{\beta \cdot 10^{3}} \quad (1)$$

where $v_{wx}(.)$, $v_{wy}(.)$ represent the horizontal speeds of wind along the respective axes, α, β are scaling coefficients, and h denotes the altitude. The arrows shown in Fig. 2 represent the horizontal winds at $h = 1000$ m. The heights of ground shaping boxes are given in square brackets.
The vertical winds are represented by means of the areas of constant speeds of the air mass flow. In Fig. 2 such air streams for the rising air masses are denoted by rectangles filled with ⊙, and for the descending ones – with ⊗. The fixed speeds of the vertical winds (in m/s) are given in round brackets.

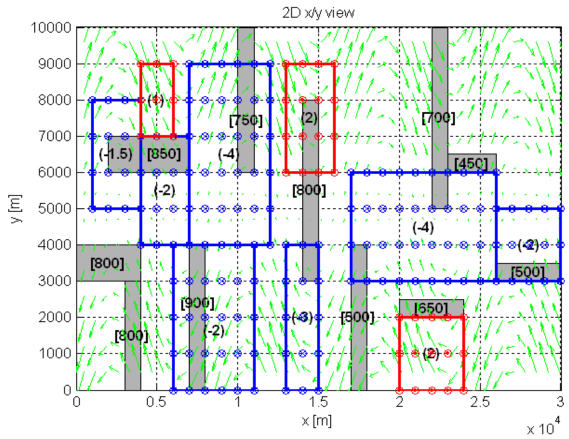

Fig. 2. Wind representation at $h = 1$ km against the ground shape: horizontal winds (arrows) and vertical winds (rectangles of a constant speed given in round brackets).

2.3 Glider model

A major simplification is applied to the model of a glider of the type *PW-5 "Smyk"* (WZL, 1994) whose total mass equals 250 kg. In fact, to select its glide ratio we analyze only on its pole curve shown in Fig. 3. This means that no transient effects associated with maneuvers, and resulting in actual changes of the glide ratio, are taken into account.

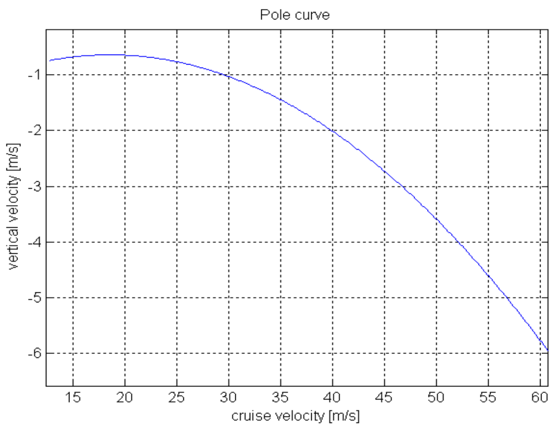

Fig. 3. Pole curve for the glider mass of 250kg.

The glider model allows us to determine the flight time between two points in the space. In this study we recognize two modes of the flight: cruise flight and climbing. Before introducing a flight-time calculation procedure let us define three basic reference frames: (i) an earth-fixed frame, (ii) a heading frame, and (iii) a body frame. With a flat flight, assumed for simplicity, the direction *h* is common for all the frames. The superscripts in the notation used in Fig. 4 symbolize particular frames.

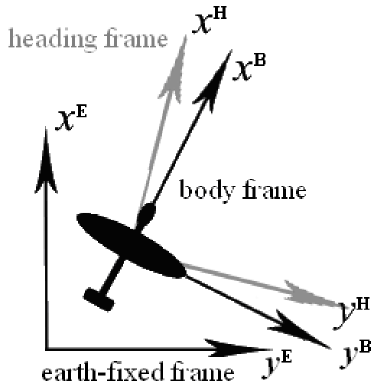

Fig. 4. Earth-fix, heading and body frames.

The proposed flight-time calculation procedure is based on linear approximation of the weather conditions between two consecutive points. This means that in order to determine the time of the flight from a point *a* to a point *b*, first we calculate the mean values of the horizontal and vertical winds:

$$\begin{aligned} \bar{v}_{wx}^{E} &= (v_{wx}^{E}(a) + v_{wx}^{E}(b))/2 \\ \bar{v}_{wy}^{E} &= (v_{wy}^{E}(a) + v_{wy}^{E}(b))/2 \\ \bar{v}_{wh} &= (v_{wh}(a) + v_{wh}(b))/2 \end{aligned} \tag{2}$$

In the case of climbing we assume that $d_{ab}^{h} = h_b - h_a > 0$ and $\bar{v}_{wh} - s_{MIN} > 0$ for a given minimum sinking speed s_{MIN}. The flight time from *a* to *b* is then simply

$$t_{ab} = d_{ab}^{h} / (\bar{v}_{wh} - s_{MIN}) + \tau \tag{3}$$

The constant τ stands for the time interval of changing a cruise flight into climbing, which is necessary for the glider to gain the required attitude and the minimum sinking speed (Almgren and Tourin, 2004).

Flight time determination for the cruise-flight mode is more complex. First, we have to functionally approximate the pole curve. By following Almgren and Tourin (2004) we have:

$$s(v_{gx}^{B}) = v_{gx}^{B} r_0 + \eta (v_{gx}^{B} - v_0)^2, \ \eta = \frac{r_0^2}{4(s_0 - s_{MIN})} \tag{4}$$

where $s(.)$ denotes the glider sink rate (sinking speed) function, v_{gx}^{B} stands for the absolute value of the longitudinal projection of the glider cruise velocity v_g^E relative to the earth-frame, r_0 is the maximum value of the glide ratio being the ratio of s_0 to v_0, which are certain parameters relating to the sinking speed and the cruise speed, respectively. These values can be found in the operation manual of aircrafts (see (WZL, 1994) for the *PW-5* glider, for instance).

Next, we consider the necessary course correction required to take account of the effect of crosswinds.

Let us introduce the notions of path distance and elevation:

$$d_{ab}^{xy} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \tag{5}$$

$$r_{ab} = \frac{h_b - h_a}{d_{ab}^{xy}} = \frac{s(v_{gx}^{B}) + \bar{v}_{wh}}{v_{gx}^{H}} \tag{6}$$

If there are no crosswinds, the components v_{gx}^{B} and v_{gx}^{H} are equal. In the other cases the crosswind component v_{wy}^{H}. should be compensated, resulting in

$$| v_{gx}^{B} | = \sqrt{{v_{gx}^{H}}^2 + {v_{wy}^{H}}^2} \tag{7}$$

Considering only positive values of v_{gx}^{B} in (4), and then in (6), and next sorting w.r.t. v_{gx}^{H}, we obtain the following:

$$\begin{aligned} & {v_{gx}^{H}}^4 \eta^4 + {v_{gx}^{H}}^3 (2 r_{ab} \eta) + \\ & + {v_{gx}^{H}}^2 (2 {v_{wx}^{H}}^2 \eta + {r_{ab}}^2 - 2\eta\rho - \lambda^2) + \\ & + v_{gx}^{H} (2 r_{ab} \eta {v_{wx}^{H}}^2 - 2 r_{ab} \rho) + \bar{v}_{wh}^4 \eta^2 + \rho^2 + \\ & - {\bar{v}_{wy}^{H}}^2 (2\eta\rho + \lambda^2) = 0 \end{aligned} \tag{8}$$

subject to:

$$\mathrm{Im}(v_{gx}^{H}) = 0 \wedge v_{MIN} \le v_{gx}^{H} \le v_{MAX} \tag{9}$$

where $\rho = \eta v_0^2 + \bar{v}_{wh}$ and $\lambda = r_0 - 2\eta v_0$, whereas v_{MIN} and v_{MAX} stand for the minimum and the maximum allowed cruise speed, respectively.

Once the maximal root v_{gx}^{H} of (8) satisfying (9) is found, the flight time can be calculated as

$$t_{ab} = d_{ab}^{xy} / v_{gx}^{H} \tag{10}$$

If there is no solution v_{gx}^{H} satisfying (9) we simply assume that $t_{ab} = \infty$.

3. ALGORITHM DESCRIPTION

While constructing the 3D mesh representation of the work space one has to observe specific process feasibility constraints. In particular, the distance between the mesh nodes in particular directions should be consistent with the glide ratio for a given glider type.

Fig. 5 presents an exemplary set of mesh nodes that will be used in the illustrative example discussed in Section 4. This is a regular structure whose horizontal (in the *x-y* plane) grids of the size 1000 × 1000 m are placed each 30 m along the axis *h*. The applied 'safe' dimensions of the segments were chosen according to the optimal glide ratios resulting from Fig. 3, which are suitable for the crosswinds ±5 m/s vertically and ±30 m/s horizontally.

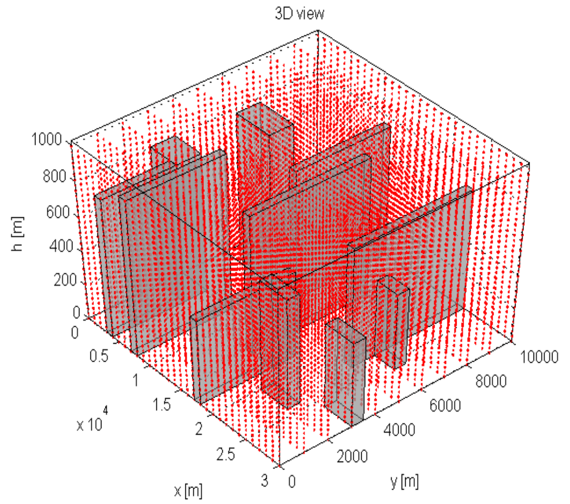

Fig. 5. Exemplary mesh representing a flight workspace.

On the basis of the flight workspace mesh, we can attempt to seek for an optimal flight trajectory. The presented algorithm consists in searching for a piece-wise linear flight trajectory founded on the mesh. Consequently, the vertices of the trajectory found represent a sequence of neighboring nodes of the 3D workspace.

For physical reasons, the vertices of the prospective trajectory (nodes of the mesh), say *a* and *b*, have to satisfy the following neighboring restriction:

$$d_{ab}^{xy} \leq 1000\sqrt{2}\ \text{m} \wedge (h_b < h_a) \quad \text{for cruise flight} \quad (11)$$

$$d_{ab}^{xy} = 0 \wedge h_b - h_a = 30\ \text{m} \quad \text{for climb flight} \quad (12)$$

There are also other feasibility restrictions concerning the existence of possible obstacles. Namely, none of the edges connecting any pair of trajectory vertices can pass through the areas earmarked for obstacles or extremely bad weather conditions.

Definition 1. Optimal flight trajectory

A piece-wise linear trajectory comprised of the line segments (edges) connecting a sequence of the trajectory vertices $e = \{a_0, a_1, \ldots, a_n\}$ *is said to be optimal if it satisfies the following condition:*

$$t_{OPT} = \min_{e \in \Theta} \sum_{i=0}^{n-1} t(a_i, a_{i+1}) \quad (13)$$

where t_{OPT} *is an optimal flight time between* a_0 and a_n *points,* $t(a_i, a_{i+1})$ *denotes the flight time between two consecutive points (vertices), and* Θ *denotes the set of all possible sequences of neighboring nodes (points) having* a_0 *and* a_n *as the initial and the terminal vertex, respectively.*

The arrangement of the workspace points is represented in a graph structure, by associating each mesh node with a graph vertex, and connecting each pair of neighboring vertices by means of an edge, whose flow value is proportional to the flight time between the points portrayed by these vertices, which can be determined by the model prescriptions given in Section 2.

The next step consists in using a graph search algorithm, the purpose of which is to find a cheapest path between two vertices representing the initial and terminal workspace points (and the starting and goal positions of the glider). In our design we applied the Dijkstra algorithm (Siena, 1997).

If the optimal flight trajectory does not exist, we simply consider that such a terminal point is not reachable.

Clearly, there are applications of the presented approach, which may call for some innovations in the presented algorithm. For instance, with slight modifications this algorithm can be utilized to determine the maximum range of the glider for a given initial position. In such a case with no terminal node being defined the graph search procedure is performed until the whole graph is explored. Moreover, for the sake of efficiency certain simplifications concerning the method of cost calculation can be introduced. Nevertheless, such deliberations are not considered here due to the limited size of this presentation.

4. EXAMPLES

In this section three examples are presented. The first one concerns seeking for the optimal flight trajectory based on the atmospheric and glider modeling introduced in Section 2. The other two examples illustrate the idea of using the presented method to determine the area of reachability for the glider starting at a given initial position.

4.1 *Example 1*

Let us consider the problem of finding an optimal flight trajectory for the glider *PW-5* originated at the initial position $a_0 = \{0,0,270\}$ and landing at the required terminal position $a_n = \{30000,10000,0\}$. The optimal solution obtained is characterized in Figs. 6, 7 and 8.

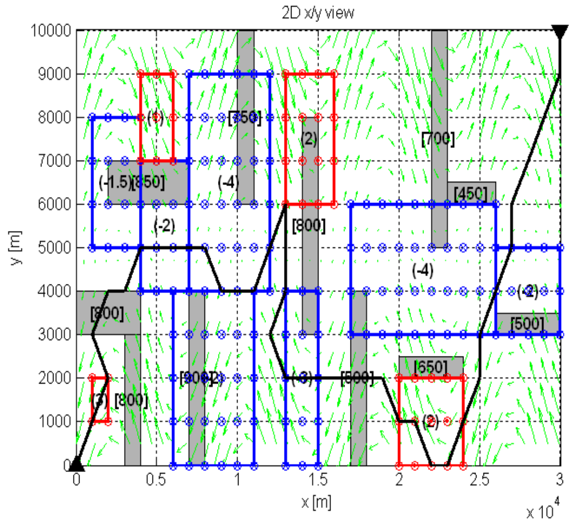

Fig. 6. Optimal flight trajectory (black solid line) in 2D for initial (Δ) and terminal points (∇).

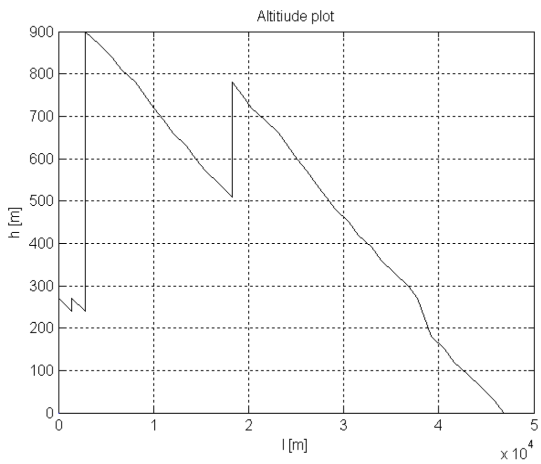

Fig. 7. Plot of the glider altitude along the route.

Note that the glider starts in the area surrounded by the ground obstacles of the height of 800 m. Therefore, the nearest thermal is we utilized to increase the glider altitude.

4.2 *Example 2*

Let us analyze the issue of determining the maximum range of the glider starting from the initial position at $a_0 = \{0,0,1000\}$. This time we assume that there are no thermals (see Fig. 9). The results are presented in Fig. 10.

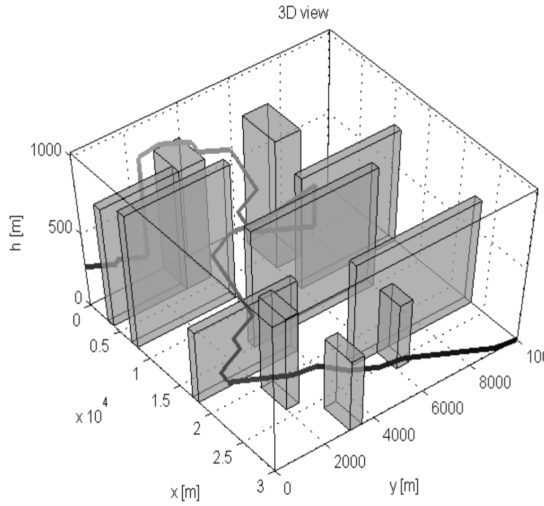

Fig. 8. Optimal flight trajectory (shaded solid line) in 3D.

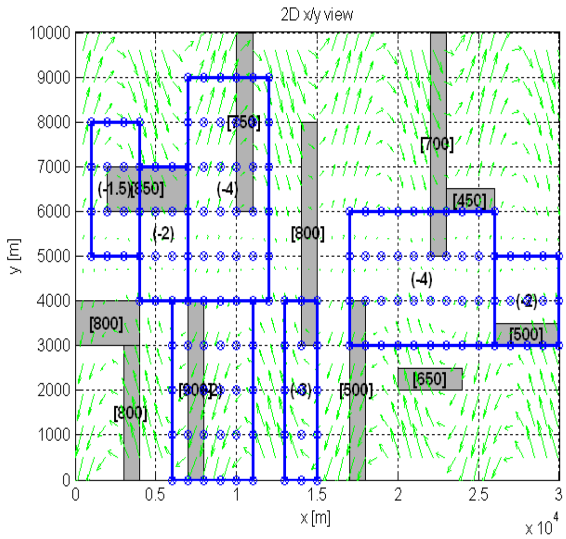

Fig. 9. Atmospheric conditions for Example 2.

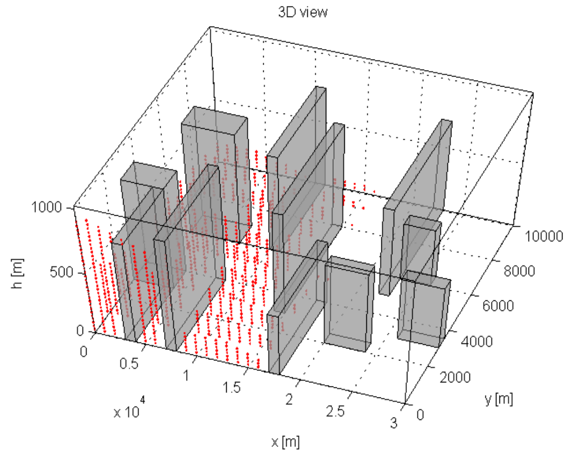

Fig. 10. Maximum glider range characterized by means of a set of reachable nodes of the workspace mesh of Example 1.

4.3 *Example 3*

Let us reassess the problem of Example 2. This time the assumed atmospheric conditions are presented in Fig. 11, whereas and the obtained solution is portrayed by Fig. 12.

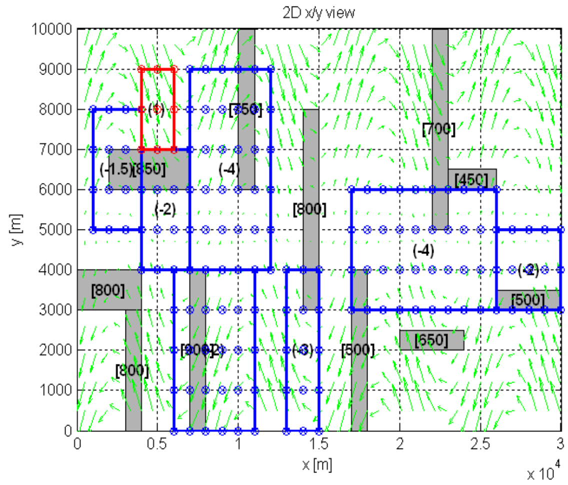

Fig. 11. Atmospheric conditions in Example 3.

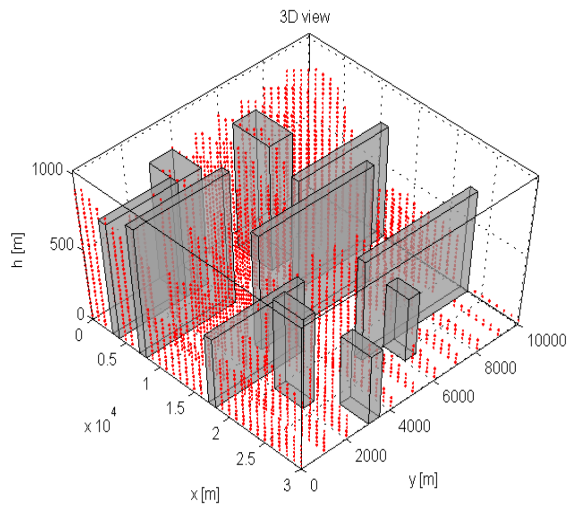

Fig. 12. Maximum glider range in Example 3 as a set of reachable nodes of the workspace.

5. CONCLUSIONS

This paper presents a research framework and an algorithm for optimal flight trajectory planning. However, the primary goal of this paper was not to present a complex solution for a particular problem but to outline the idea of this method. In fact, we are concerned about a general problem of utilizing the autonomous dynamics of processes to design a safe and optimal control strategy. Such a kind of study is reported in our recent presentations (Kowalczuk *et al.*, 2007; Kowalczuk and Olinski, 2007).

Since, generally, in the real world, there are no exact atmospheric maps, we have assumed a relatively straightforward atmospheric model. Even if we were able to construct an approximated weather condition map, it would be valid practically only for some limited time interval. To make the atmospheric model more reliable it would also be necessary to take into consideration some additional phenomena, like reflected winds near the ground obstacles or other shapes of thermals imitating the vertical winds.

Due to the limited capacity of the paper, we have omitted the discussion about the efficiency of the presented method, which principally depends on the applied search graph algorithm. The Dijkstra algorithm imposes the complexity of the presented approach (which is thus in our case of square order). Some clues concerning the possibilities of improving the search graph process can be found, for instance, in Bertsekas, (1995) and Fredman and Petzold (1987).

REFERENCES

Almgren, R. and A. Tourin (2004). *Optimal Soaring with Hamilton – Jacobi - Bellman Equations. http://www.cims.nyu.edu/~almgren/papers/optsoar.pdf.*

Asher, U.M. and L.R. Petzold (1997). *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*. Society of Industrial and Applied Mathematics, Philadelphia (PA).

Bertsekas, D.P. (1995). *Dynamic Programming and Optimal Control*. Athena Scientific, Nashua (NH).

Bertsekas, D.P. (2000). *Nonlinear Programming*. Athena Scientific, Nashua (NH).

FAA (2003). *Glider Flying Handbook*. Flight Standards Service, Federal Aviation Administration, U.S. Department of Transportation.

Friedman, M.L. and R.E. Petzold (1987). Fibbonaci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3), 596-615.

Kowalczuk, Z., K. Rudzinska-Kormanska, and K.E. Olinski (2007). Designing nonlinear control systems by state-space flow graph optimization. In: *Proc. of the 11th IFAC Symposium on Large Scale Systems*, (CD-ROM). Gdansk (Poland).

Kowalczuk, Z. and K.E. Olinski (2007). Sub-optimal fault-tolerant control with the use of discrete optimization. In: *Fault Diagnosis and Fault Tolerant Control* (J. Korbicz, K. Patan, M. Kowal (Eds.)), pp. 165-172. Academic Publishing House EXIT, Warsaw.

Siena, S. S. (1997). *The Algorithm Design Manual*. Springer-Verlag, New York.

WSK (1994). *Sail Plane Flight Manual*. WSK “PZL Swidnik”, Swidnik (Poland).