Algorithm of routes optimization for mobile robots

Vladislav Solovtsov, Dzmitry Adzinets

***Abstract* —** The article presents a method of route optimization on resources which should be spent for route passage. The proposed algorithm is based on the idea that rise require more resources than the descent for passage. For flat locality, the result of the algorithm coincides with the result obtained from Google Maps. For hilly and mountainous terrain, the algorithm selects routes that are differ from Google Maps with a lower content of the number of lifts, which accordingly makes routes less energy consuming.

***Keywords* —** energy functional, navigation solution, optimal route.

I. Introduction

Nowadays mobile robots can be used for carrying out explosion works and works in dangerous areas, carrying out works during liquidation of consequences of worst-case situations. Also, robots are widely used in delivery services. Before sending robots to work on the ground, the routes they will follow should be optimized. Existing navigation solutions offer the following route search optimizations: the fastest route and the shortest route [1], [2]. In this article we will propose an algorithm of route optimization on resources which should be spent for route passage.

The target audience of this algorithm is pedestrians, cyclists and drivers.

II. Task statement

It is required to find a route between two points (the route can also include a number of waypoints), the passage of which requires the least amount of resources.

We will proceed from the fact that more resources are spent on a rise than on the descent, then *the least cost route* is the route with the least quantity of lifts, as the rise takes more petrol for drivers as well physical effort for pedestrians and cyclists.

Primary data. Suppose the route includes 4 points, where point #1 is the starting point, point #4 is the end point, points #2 and #3 are waypoints of the route (Fig. 1). There are 3 routes (can be more or less) between each current and next points, in the figure they are marked with green, red and yellow colors.

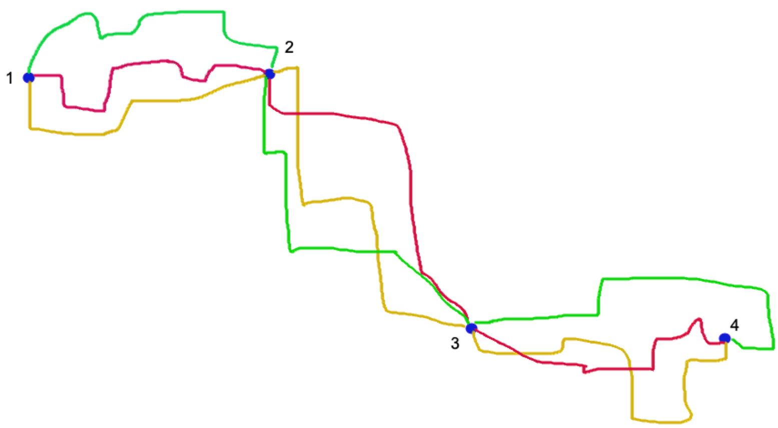

Fig. 1 Route and waypoints, top view
(the figure does not carry any information about the elevation on the ground)

V Solovtsov, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus (e-mail: solovtsov.vladislav@gmail.com).
D. Adzinets, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus (e-mail: adzinets@bsuir.by).

Routes between points shown in the figure are obtained using navigation services such as Google Maps/Here Maps/Sygic. In this paper, the service Google Maps will be used as an electronic map data provider. Google Maps Directions API will be used to get routes [3], to do this, HTTP GET request should be sent to the following address:

https://maps.googleapis.com/maps/api/directions/json?parameters

where parameters are query parameters, among them:

— origin, required parameter, departure address;

— destination, required parameter, destination address;

— key, required parameter, API Authorization Key, obtained in Google Developers Console;

— mode, travel mode for route building, one of the DRIVING, WALKING, BICYCLING;

— alternatives, a flag, specifies that the Directions service may provide more than one route alternative in the response.

The value of the alternatives flag must be set to true, because then the least cost route will be selected from the returned main and alternative routes.

Elevation value of any location can be obtained with Google Maps Elevation API [4]. To do this, HTTP GET request should be sent to the following address:

https://maps.googleapis.com/maps/api/elevation/json?parameters

where parameters are query parameters, among them:

— key, required parameter, API Authorization Key, obtained in Google Developers Console;

— locations, the location(s) on the earth (latitude and longitude) from which to return elevation data.

Thus for any point of any route it is possible to get the elevation value. For example, for the green route section between points 1-2 of Figure 1, the chart of the elevation differences may look like in Fig. 2:

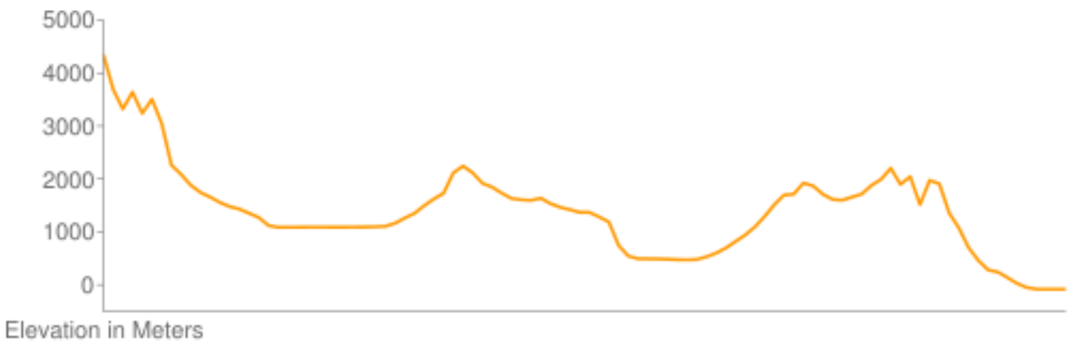

Fig. 2 Chart of the elevation differences for the green route section 1-2

III. DESCRIPTION

The main idea is to assign a weight to each route between each adjacent two points. For this, the route is divided into many sections, for each section the length L, the elevation of the starting point Hstart and the elevation of the end point of the section Hend are known.

Let's introduce the concept of *energy functional*:

$$E = f(a), \tag{1}$$

where α — slope angle of the route. The angle can be calculated by the following formula:

$$\alpha = sin\frac{H_{end} - H_{start}}{L} \quad (2)$$

Let's introduce the concept of *cost functional*:

$$b = f(E, L), \quad (3)$$

where E — previously introduced energy functional, L — section length.

The weight value for the entire route will be calculated as the sum of the cost functional values for each section:

$$B = \sum_{i=1}^{n} b_i \quad (4)$$

Weights are calculated for each route. Thus, each route will correspond to its total weight, proportional to the cost of its passage:

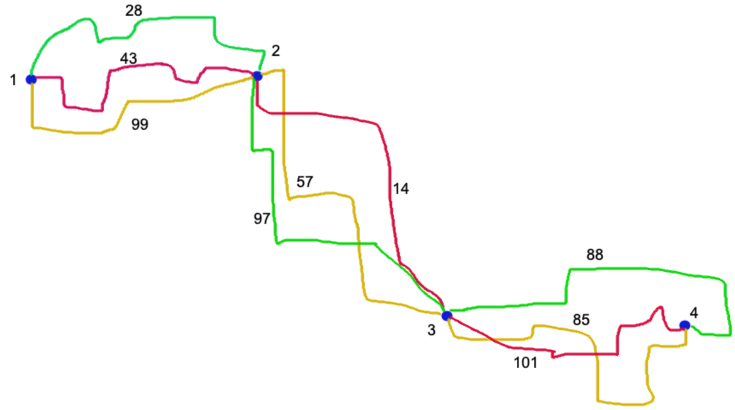

Fig. 3 Routes with calculated weights

After the weights are calculated, we can move from the map (Fig. 3) to the graph where the vertices are the points of the route, and the routes between the points are the edges of the graph. The weight of the route is equal to the edge weight (Fig. 4).

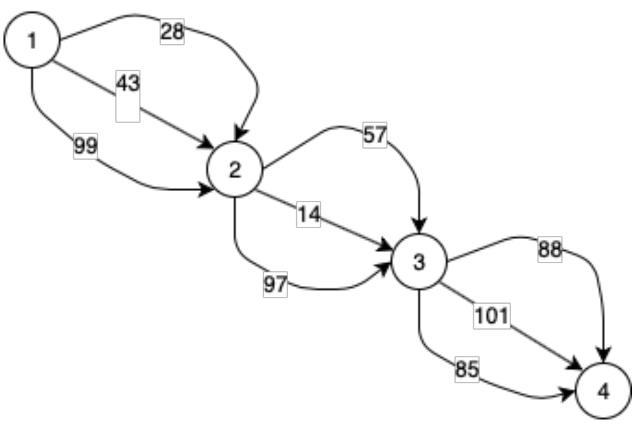

Fig. 4 Route graph

Here are examples of calculations of energy functional and cost functional (Fig. 5, 6):

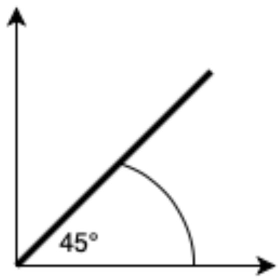

Fig. 5 Route section — rise — has a slope α = 45

For this section the energy functional will be calculated according to the following formula:

$$E = 1 + \frac{a}{100} = 1.45 \tag{5}$$

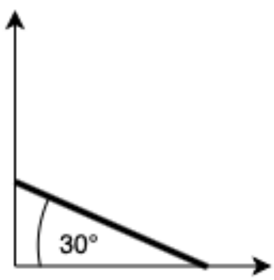

Fig. 6 Route section — declivity — has a slope α = 30°

For this section the energy functional will be calculated according to the following formula:

$$E = 1 - \frac{a}{100} = 0.70 \tag{6}$$

Thus for the rises the functional energy will be calculated by the formula:

$$E = 1 + \frac{a}{100} \tag{7}$$

for descents:

$$E = 1 - \frac{a}{100} \tag{8}$$

In turn, the cost functional will be calculated as:

$$b = f(E, L) = E \cdot L \tag{9}$$

IV. Test Results

An example of the algorithm work is the route searching in Lisbon, Portugal, the beginning of the route — R. Pinheiro Chagas 1050, the end of the route — R. Damasceno Monteiro 1170 (Fig. 7).

Alternative routes and corresponding weights calculated according to the algorithm are presented in table 1.

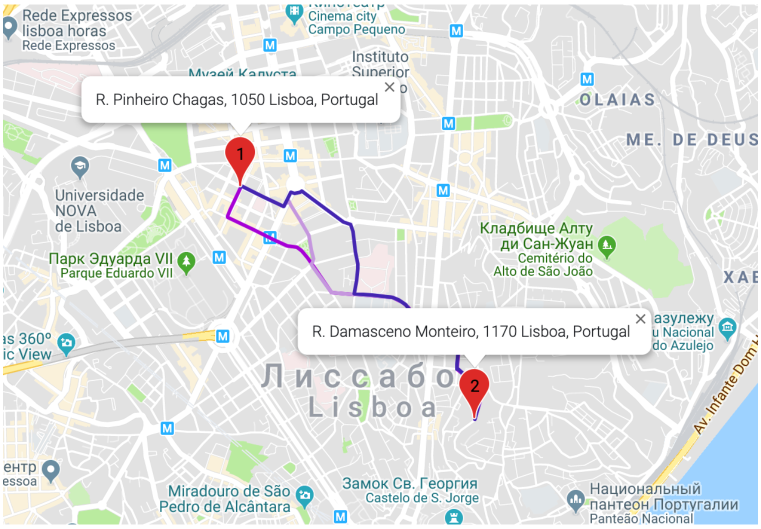

Fig. 7 Routes between waypoints, Lisbon, Portugal

TABLE I
Routes weights, Lisbon, Portugal

#	Start	End	Color	Weight
1	R. Pinheiro Chagas 1050	R. Damasceno Monteiro 1170		2603.28827
2	R. Pinheiro Chagas 1050	R. Damasceno Monteiro 1170		2478.29080
3	R. Pinheiro Chagas 1050	R. Damasceno Monteiro 1170		2492.28877

Thus the algorithm has selected the route with the lowest weight value, namely the route #2 from table 1 (Fig. 8).

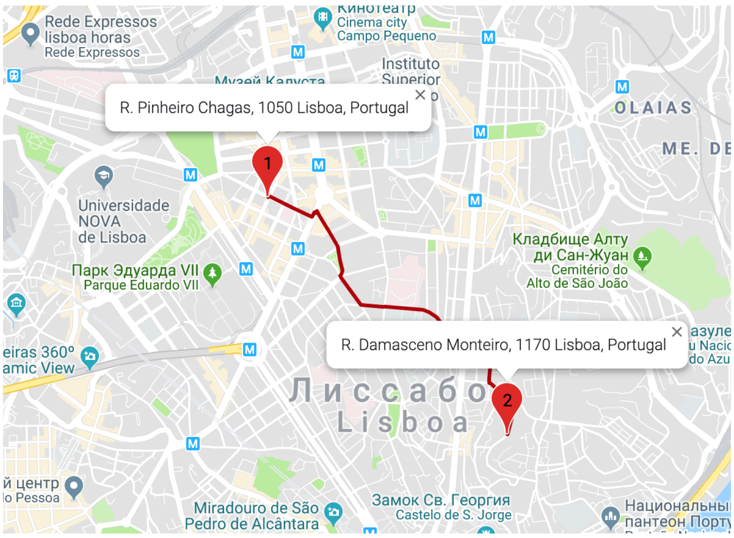

Fig. 8 Route chosen by the algorithm, Lisbon, Portugal

The Route offered by Google Maps navigation service corresponds to route #3 from table 1 and is shown in Fig. 9.

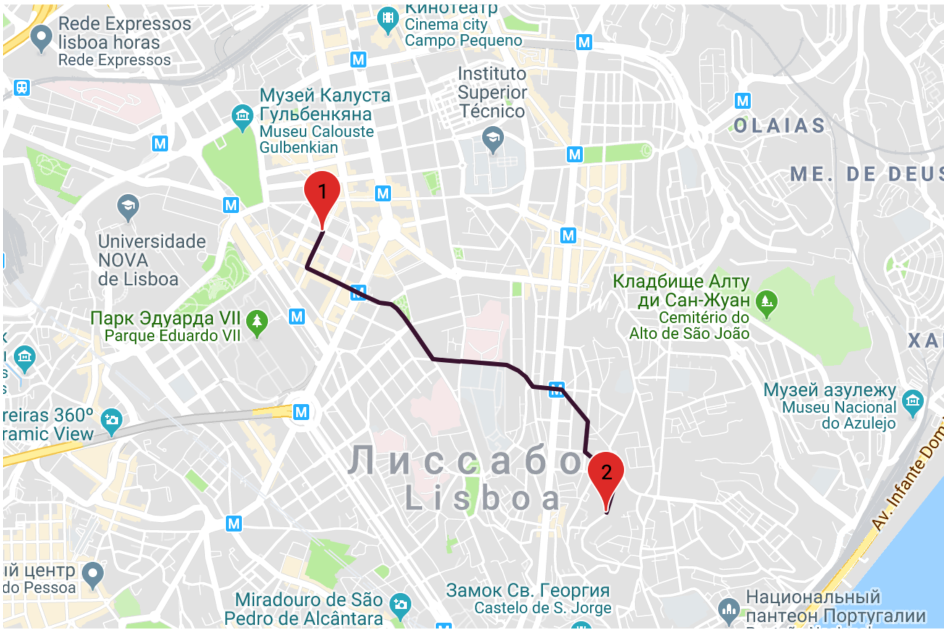

Fig. 9 Route offered by Google Maps Service

Example for a route that includes waypoints is route searching in Tokyo, Japan. The Point of departure is 2-chōme−11−4, 105-0012 Tōkyō-to, Minato City, Shibadaimon, the destination point is 2-chōme-4-12 Kyōbashi, Chuo City, Tōkyō-to 104-0031, and the waypoint is 2-chōme-2 Kasumigaseki, Chiyoda City, Tōkyō-to 100-0013 (Fig. 10).

Fig. 10 Routes between waypoints, Tokyo, Japan

Alternative routes and corresponding weights calculated according to the algorithm are presented in table 2.

Based on the data in table 2, the search route will consist of two parts, route #1 and route #5, as these routes have the smallest weights for their sections of the path. Searching route is shown in Fig. 11.

The route offered by Google Maps navigation service corresponds to route #3 and route #4 from table 2 and is presented in Fig. 12.

TABLE II
Routes weights, Tokyo, Japan

#	Start	End	Color	Weight
1	2-chōme–11–4, Shibadaimon, Minato City, Tōkyō-to 105-0012	2-chōme-2 Kasumigaseki, Chiyoda City, Tōkyō-to 100-0013		5025.43739
2	2-chōme–11–4, Shibadaimon, Minato City, Tōkyō-to 105-0012	2-chōme-2 Kasumigaseki, Chiyoda City, Tōkyō-to 100-0013		5071.41868
3	2-chōme–11–4, Shibadaimon, Minato City, Tōkyō-to 105-0012	2-chōme-2 Kasumigaseki, Chiyoda City, Tōkyō-to 100-0013		5042.46492
4	2-chōme-2 Kasumigaseki, Chiyoda City, Tōkyō-to 100-0013	2-chōme-4-12 Kyōbashi, Chuo City, Tōkyō-to 104-0031		2306.16193
5	2-chōme-2 Kasumigaseki, Chiyoda City, Tōkyō-to 100-0013	2-chōme-4-12 Kyōbashi, Chuo City, Tōkyō-to 104-0031		2235.16115
6	2-chōme-2 Kasumigaseki, Chiyoda City, Tōkyō-to 100-0013	2-chōme-4-12 Kyōbashi, Chuo City, Tōkyō-to 104-0031		2280.21866

Fig. 11 The route chosen by algorithm

Fig. 12 The route proposed by Google Maps service

V. CONCLUSION

By analyzing finished research, it can be concluded that the result of the algorithm is better than the similar solution from Google Maps for hilly and mountainous terrain. For flat locality, the result of the algorithm coincides with the results obtained from Google Maps. For hilly and mountainous terrain, the described algorithm selects routes that are differ from Google Maps with a lower content of the number of lifts, which accordingly makes routes less energy consuming.

REFERENCES

[1]. Tarasyan V., Polushkin A. (2017) OPTIMIZATION OF THE PATH IN THE INHOMOGENEOUS ENVIRONMENT // Fundamental research. Retrieved from http://www.fundamental-research.ru/ru/article/view?id=41828

[2]. Improving Operations with Route Optimization – Towards Data Science (2018). Retrieved from https://towardsdatascience.com/improving-operations-with-route-optimization-4b8a3701ca39

[3]. Developer Guide | Directions API | Google Developers (2019). Retrieved from https://developers.google.com/maps/documentation/directions/intro

[4]. Developer Guide | Elevation API | Google Developers (2019). Retrieved from https://developers.google.com/maps/documentation/elevation/intro