Electrical, Control and Communication Engineering
ISSN 2255-9159 (online)
ISSN 2255-9140 (print)
2021, vol. 17, no. 1, pp. 59–66
https://doi.org/10.2478/ecce-2021-0007
https://content.sciendo.com

A Hybrid Routing Approach Using Two Searching Layers

Gonca Ozmen Koca (*Associate Professor, Mechatronics Engineering Department, Technology Faculty, Firat University, Elazığ, Turkey*),
Seda Yetkin* (*Lecturer, Vocational School of Technical Sciences, Electronics and Automation Department, Bitlis Eren University, Bitlis, Turkey*)

***Abstract* – This paper considers SUB_GOALs by using basic A* algorithm and Subgoal Graphs in a hybrid approach to execute optimal route. SUB_GOALs identified with pre-searching from basic A* at break points and Subgoal Graphs at corners of obstacles are added to SUB_TABLE to expedite the final searching in the hybrid approach. Map to work on is divided to subregions with decision-making process by using line-of-sight to avoid redundant searching. In the final searching layer, all feasible SUB_GOALs gained from decision-making process in the same subregion are connected to find final solutions of routes. Solutions achieved in the divided subregions are evaluated and combined to discover the final optimal route. The proposed hybrid approach is applied to three different scenarios in various dimensions of maps. In these three scenarios, the shortest route without hitting obstacles is calculated as 46.67, 57.76 and 124.7 units, respectively, and compared with other search algorithms. Simulation results of route planning are demonstrated to exhibit the effectiveness of the proposed hybrid approach.**

***Keywords* – Motion control; Optimization; Path planning; Shortest path problem.**

I. Introduction

In mobile robot research, route planning is one of the key factors for implementation of autonomous mobility of robots. Various solutions can be suggested by different characteristics of planning as performance improvements, different escape plans according to the risks, collision avoidance especially for mobile obstacles. Most of the studies dedicated to the planning of the optimum route indicate obtaining the effective route from starting point S to final point F avoiding obstacles or collisions in complicated environment. Route planning algorithms focus on different approaches as grid-based searching, interval-based searching, geometric algorithms, sample-based algorithms, and remarkable points algorithms. These approaches can be used to solve various structures of different problems with advantages and disadvantages.

In recent years, grid-based search algorithms have been used to find the optimum route in many application areas (such as maps, games). Photographers in the Indonesian city of Bandar Lampung have used one of the grid-based searching algorithms, the A* algorithm, to find the optimum route to the photo points [1]. In another study in Indonesia, the A star algorithm has been used to find the shortest path distance between the Ministry of Pharmacy Polytechnic and Medan Health Polytechnic [2]. A* Algorithm has been used in a game involving a guard and a thief to choose the shortest route that can be taken by the guard to catch the thief [3].

Grid-based searching divides the operation field into grids and the robot is moved from the grid where it is located to neighbour grids according to the searching algorithm (such as A*, Dijkstra, etc.) in order to create a route from point S to point F [4]. Interval-based searching algorithms originated from a similar approach of grid-based searching algorithms but the interval searching algorithm is operated with configuration space which uses each coordinate as a degree of freedom of the object instead of grids [5]. Sample-based algorithms use sample configurations in the configuration space. A*, D* and etc. algorithms keep in a list remarkable points to find the shortest route from S to F. A* Algorithm is an admissible heuristic shortest route finding algorithm which uses the function $f(n) = g(n) + h(n)$ to calculate the distance. $g(n)$ is the cost of coming from starting point S to current point R, $h(n)$ is also estimated cost to reach final point F from current point R and as will be noted, the reason why $f(n)$ is heuristic is that it has a heuristic function h(n) which is based on the prediction of the cost from R to F [6]. D* is also Dynamic A* Algorithm which is one of the improved version of it. In this algorithm, the cost parameters change during the operation of the problem as the name of algorithm implies with Dynamic A* [8]. More A* based algorithms are developed and used for different applications with the improvements in the A* Algorithm to obtain good enough route planning for mobile navigation of autonomous robots [9].

On the other hand, various intelligent optimization methods as Fuzzy controller, Genetic Algorithm, Ant Colony Algorithm, Taboo Search Algorithm, Bee Colony Algorithm and etc. are used to ensure optimum route for different robots [7], [10]. Liu et al. [11] proposed a bat algorithm (IDBA) developed to solve the multipoint shortest path planning problem. Masehian et al. [12] also proposed robot route planning with Particle Swarm

*Corresponding author.
E-mail: sedayetkin23@gmail.com

Optimization based algorithm for the shortest and smooth route. Elkhateeb et al. [13], with the Bee Colony Algorithm, provided the control of 2 free-degree robotic arm manipulators.

II. Literature Review

In the shortest route planning approaches, the studies for finding the shortest route with less computationally intensive algorithms have gained speed by using basic methods or hierarchical structures, including approximations of simple methods. Uras et al. [14] developed a method to create Subgoal Graps (SGs) by using neighbour grids. Subgoals are defined as the corners of obstacles which supply the shortest route in the field without obstacles by using the route from current point R to the other corner point closer to final point F. Uras et al. [15] proposed a subgoal graph algorithm for different levels to generate optimal route planning in grid based areas. They tried to achieve generalization of partitioning process in order to abbreviate the graph. Nussbaum et al. [16] suggested a new algorithm for Moving Target Search with SGs to find a solution to dynamic robotics applications. This algorithm is established for segregation of the field and SGs are implemented to increase the speed of searches. Xu et al. [17] investigated finding the way by using the subgoal graph algorithm to reduce a large area to a certain extent in large area scans.

Zuo et al. [18] proposed a two-level structured hierarchical algorithm based on basic A* and least-squares policy iteration to create a smooth route with robot kinematics. The simulation results are also obtained for the moving obstacles. In the next work, they will ensure these performances in real robotic application. Peng et al. [19] developed a new solution to store data in OPEN and CLOSED tables (OPEN_TABLE, CLOSED_TABLE) in order to shorten the operation time of A* algorithm. The operation efficiency was tried to be improved by using a new storing approach and more than 40 % gain was obtained in this study. Duchon et al. [20] suggested some modifications of A* algorithm to provide route optimality and less operation time. There are some assessments related to modified A* algorithms for various scenarios of optimal route planning.

III. Motivation

In this study, some significant notions are represented as below.

- A new method is proposed in order to obtain optimal route considering SUB_GOALs by using basic A* algorithm and SGs in a hybrid approach.
- The operation area is divided into subregions with decision-making process to prevent unnecessary searching.
- SUB_GOALs determined with pre-searching from basic A* at break points and SGs at corners of obstacles are added to SUB_TABLE to speed up the final searching in the hybrid approach.
- In the divided subregions by blocking unnecessary searching with decision-making process and by using line-of-sight (LOS), all feasible SUB_GOALs from SUB_TABLE in the same subregion are connected to find solutions of routes.
- Solutions are evaluated and combined with the decision-making process to find the final optimal route.

The paper is organised as follows: In Section 1, the content of the subject is introduced. The literature review about related works is presented in Section 2. Motivation of the study is specified in Section 3. All algorithms used in this study are given in Section 4. The proposed method is represented in Section 5. Results for different scenarios are illustrated in Section 6 and finally conclusion is presented in Section 7.

IV. Objects

In this study, a hybrid algorithm is developed with basic approaches of two basic algorithms called A* algorithm and SGs. A guidance for the final search is created by using determined SUB_GOALs from basic algorithms. Thus, the shortest route is guaranteed with final search. The map is also divided into subregions to make searching as a very simple procedure. Algorithms used in the hybrid method are introduced as below.

A. Algorithm_1

A* algorithm is the most preferred option for route finding in various applications, since it can be used in a wide range of contexts with its flexibility [21]. In the basic terminology of A* algorithm, the function $f(n)$ is used for calculating distance as mentioned before and n defines each point of the operation area divided into grids. The first term of $f(n)$ is given in Eq. (1):

$$g(n) = 10\sqrt{(x_n - x_s)^2 + (y_n - y_s)^2} \tag{1}$$

and the heuristic function is presented in Eq. (2) according to Manhattan function [18]:

$$h(n) = 10(|\chi_n - \chi_F| + |y_n - y_F|). \tag{2}$$

Here, descriptions of the terms in Eqs. (1) and (2) can be given with Eq. (3):

$$\left\{(\chi_k, y_k) \rightarrow \begin{Bmatrix} \|term\|_{k=s} \\ \|term\|_{k=F} \end{Bmatrix}\right\}. \tag{3}$$

Here χ_k is the horizontal distance and y_k is the vertical distance.

Pseudocode of basic A* algorithm process indicated by Algorithm_1 is represented in Table I.

SUB_GOALs obtained from basic A* algorithm are determined as break points and saved in SUB_TABLE to generate final searching.

B. Algorithm_2

SGs use graphs between SUB_GOALs defined as corners of obstacles. Direct access of the SUB_GOALs to each other in any cardinal or diagonal direction is achieved by adding edges that do not intersect the obstacle. SUB_GOALs are selected that are closest to S and F and connecting SUB_GOALs with S and

F is achieved to find the shortest route. The pseudocode showing the operation of the SGs is given in Table II.

Let us define the location of an obstacle as OBS(i, j) and find the boundary of them with i_min∧i_max of OBS in vertical direction, j_min∧j_max of OBS in horizontal direction. SUB_GOALs are also located in {MAP(i_max+1,j−1)∧MAP(i_max+1,j+1)}∧{MAP(i_min−1,j −1)∧MAP(i_min−1,j+1)}(i,j∈OBS) for vertical direction and {MAP(i−1,j_max+1)∧MAP(i+1,j_max+1)}∧{MAP(i−1,j_min −1)∧MAP(i+1,j_min−1)}(i,j∈OBS) for horizontal direction.

V. The Proposed Hybrid Method

Hybrid method is combined from two searching layers by using basic A* and SGs algorithms. In order to determine the final route as mentioned in Fig. 1, SUB_GOALs are assigned to create SUB_TABLE at the pre-searching layer.

A* algorithm is utilised to obtain break points to be used in SUB_GOAL mission and it also shows the orientation from starting SUB_GOAL to final SUB_GOAL of each divided subregion.

TABLE I

Pseudocode of Basic A* Algorithm

	Algorithm_1
1/	*create OPEN_TABLE & put S on the OPEN_TABLE*
2/	*create CLOSE_TABLE*
3/	*while (OPEN_TABLE is not empty matrix) {*
/1	*find the neighbor_Ri with min(f) on the OPEN_TABLE*
/2	*remove neighbor_R_i from OPEN_TABLE*
/3	*assign neighbor_R_i (i=1,2,…,8) of R & define R as their parent*
//1	*for (each neighbor_R_i) {*
//2	*if (neighbor_R_i is F) stop searching {*
//3	*g(neighbor_R_i)=R*g+distance between (neighbor_R_i & R)*
//4	*h(neighbor_R_i)=distance between(F & neighbor_R_i)*
//5	*f(neighbor_R_i)=g(neighbor_R_i)+h(neighbor_R_i)*
///1	*If (there is any point (P) has lower f than neighbor_R_i on the OPEN_TABLE){*
///2	*jump neighbor_R_i*
///3	*If (there is any point has lower f than neighbor_R_i on the CLOSED_TABLE){*
///4	*jump neighbor_R_i*
///5	*else add P to the OPEN_TABLE}} end*
//6	*}end*
/4	*add R on the CLOSED_TABLE*
4/	*}end*

TABLE II

Pseudocode of Basic SGs

	Algorithm_2
1/	*create MAP & assign 2 to Map matrix*
2/	*in Map, assign −1 for obstacles, 1 for S, 0 for F*
3/	*If(MAP(i, j)=−1){OBS matrix}*
4/	*for OBS size{*
/1	*If (OBS(i)+1=OBS(i) & OBS(j)=OBS(j)){find horizontal ongoing obstacles(H_OBS)*
/2	*If (OBS(j)+1=OBS(j) & OBS(i)=OBS(i)){find vertical ongoing obstacles(V_OBS)*
/3	*else {find single obstacles (S_OBS)}*
/4	*end}*
5/	*find start and final point of H_OBS & V_OBS*
6/	*find SUB_GOALs {start H_OBS(i−1,j+1) & start H_OBS(i−1,j−1) ||start V_OBS(i−1,j−1) & start V_OBS(i+1,j−1) ||final H_OBS(i+1,j+1) & start V_OBS(i+1,j−1) ||final V_OBS(i−1,j+1) & start V_OBS(i+1,j+1) ||S_OBS(i−1,j+1) & S_OBS(i−1,j−1) & S_OBS(i+1,j+1) & S_OBS(i+1,j−1)*
7/	*select the point into SUB_GOALs which is nearest to S & distances between the selected point to other SUB_GOALs are calculated.*
8/	*path=min(distance)*

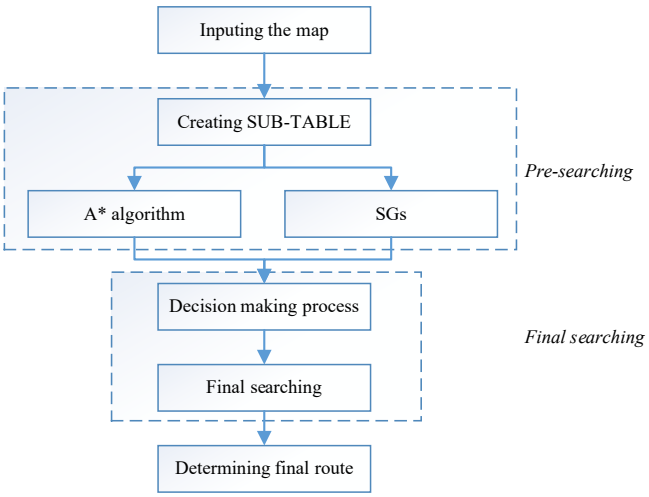

Fig. 1. The flowchart of the proposed hybrid method.

Definition 1: Creating joint SUB_GOAL to obtain subregions.

$$\begin{aligned}&\forall point(i,j)\in[Map]\_mxn,\\&(i=1,...,m;j=1,...,n)\\&(j\to \text{distance}_x; i\to \text{distance}_y)\\&If\,(\text{parents(SUB\_TABLE(p)},\\&\{p=1,...,size(\text{SUB\_TABLE})\})\text{is A}^*\\&\quad\&\,(\text{parents(SUB\_TABLE(p))}\\&\{p=1,...,size(\text{SUB\_TABLE})\})\text{is SGs})\\&\quad\text{then}(p\,\text{is joint}\text{SUB}_{\text{GOAL}})\end{aligned}\tag{4}$$

Let us find the other new joint SUB_GOAL which has the same ($i \vee j$) with joint SUB_GOAL. Then let us choose joint SUB_GOAL=new joint SUB_GOAL. Let us find the number of (joint SUB_GOAL) $\to$t which is also the number of divided subregion. In the light of these descriptions, the Map_1 divided into sub-regions for the first scenario is given in Fig. 2.

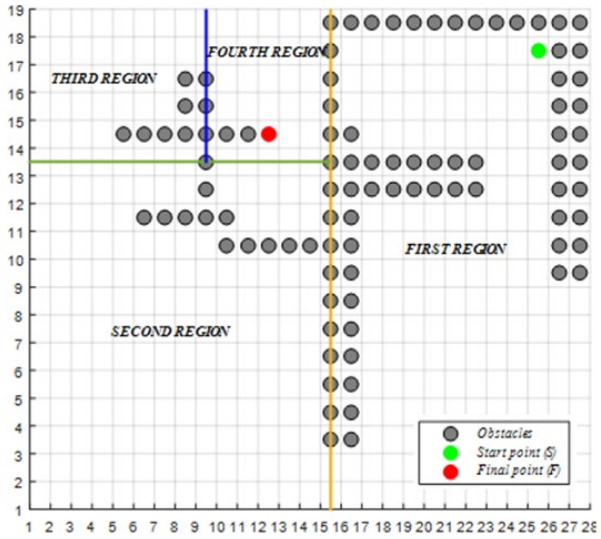

Fig. 2. Division into four regions of Map_1 for the first scenario.

Definition 2: Determining S and F of each subregion.
Let us assign starting and final point of each subregion as:

$$\begin{aligned}&\{\text{S(subregion}(k))=\text{S},\text{F(subregion}(k))\\&\qquad=\text{jointSUB\_GOAL(k)}\},\{k=1\}\\&\left\{\begin{matrix}\text{S(subregion}(k+1)=\text{F(subregion}(k),\\ \text{F(subregion}(k+1)=\text{jointSUB\_GOAL}(k+1)\end{matrix}\right\},\\&\qquad\{1<k<\text{t}\}\end{aligned}\tag{5}$$

If one uses all SUB_GOAL combination to find the shortest path, it may take long time. In this approach, LOS defined with the pseudocode given in Table III is used to see direct h-reachable SUB_GOALs from one to other as shown in Fig. 3 for the first region of the map with the first scenario

.

TABLE III
PSEUDOCODE OF LOS

Algorithm_3	
1/	*creat zeros(Map)*
2/	*in Map, assign -1 for obstacles, 1 for S, 2 for SUB_GOALs*
3/	*if (there are obstacles near S in the Map){*
/1	*determine the corner points of the obstacles & construct the OBS corner*
/2	*draw STRAIGHT LINES between the OBS corner and S with y=mx+n*
4/	*if (SUB_GOALS in Map are between STRAIGHT LINES){*
/1	*if the slopes of these SUB_GOALS and S are same)) {*
/2	*select the nearest point of S*
/3	*selected point =visible point } end*
5/	*visible point*
6/	*else invisible point } end*

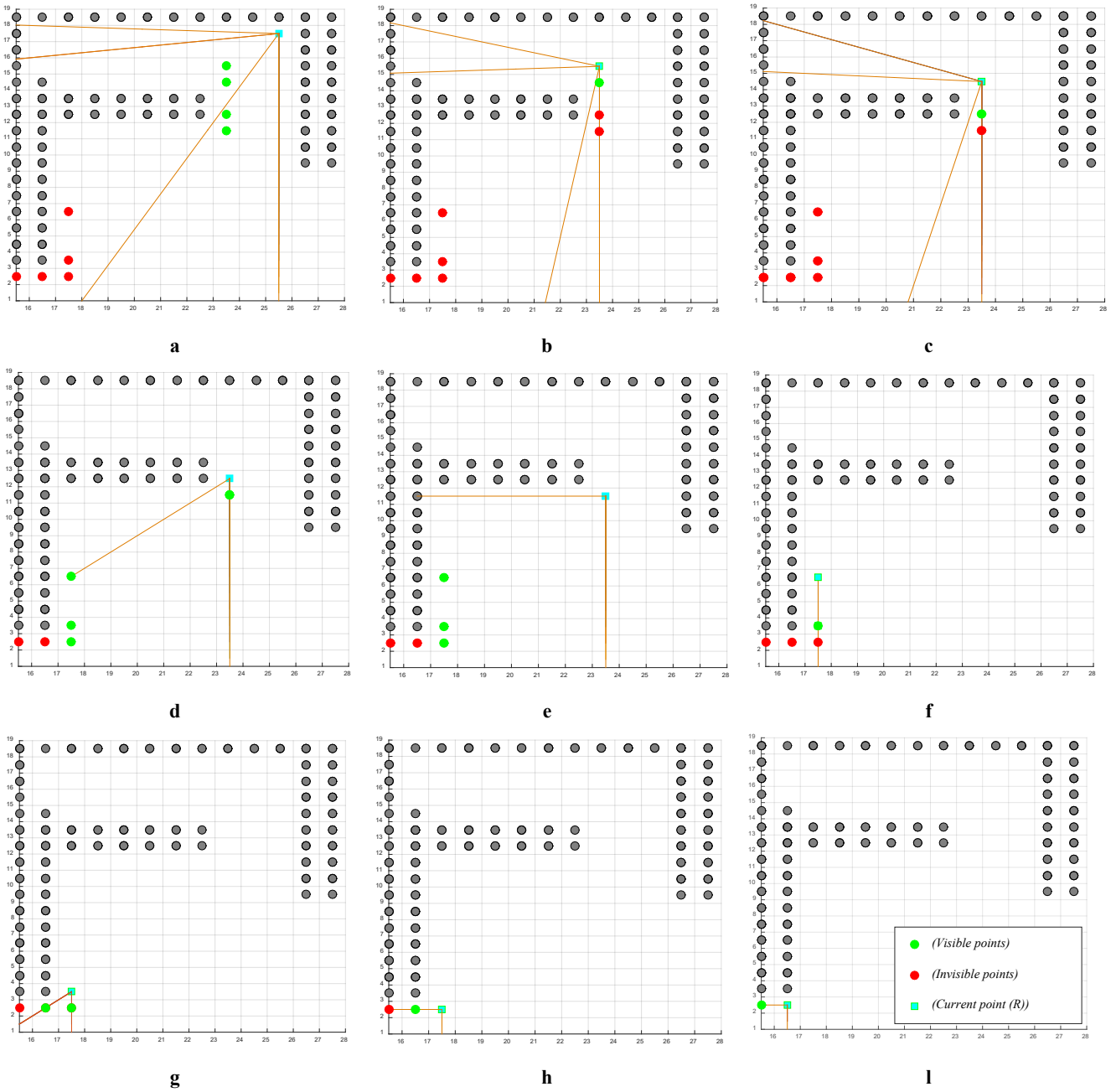

Fig. 3. Step-by-step operation of LOS procedure in the first region of the Map_1 for the first scenario.

Decision-making process is used to determine suitable visible point instead of searching all visible points from the current point. In this way, redundant combinations of current point and visible points are eliminated and only suitable combination is obtained by using the decision-making process. Here, visible, invisible and current points indicate SUB_GOALs from SUB_TABLE.

Definition 3: Orientation of each sub region.

Slope of each subregion gives an idea about the orientation of route at each subregion. Let us consider the first scenario as an example. In this map, t is 4 and the direction of slope can be described with Eq. (6).

$$\begin{aligned}
&\forall SUB_GOAL(i,j) \in subregion(k=1) \rightarrow \\
&\qquad \{-x,-y\},\{i \rightarrow i_{\min} \wedge j \rightarrow j_{\min}\} \\
&\forall SUB_GOAL(i,j) \in subregion(k=2) \rightarrow \\
&\qquad \{-x,+y\},\{i \rightarrow i_{\max} \wedge j \rightarrow j_{\min}\} \\
&\forall SUB_GOAL(i,j) \in subregion(k=3) \rightarrow \\
&\qquad \{+x,+y\},\{i \rightarrow i_{\max} \wedge j \rightarrow j_{\max}\} \\
&\forall SUB_GOAL(i,j) \in subregion(k=3) \rightarrow \\
&\qquad \{+x,-y\},\{i \rightarrow i_{\min} \wedge j \rightarrow j_{\max}\}
\end{aligned} \tag{6}$$

Let us find joint SUB_GOAL (1) for the first region which has the location of ($x = 15 \wedge y = 2$). S(sub region(1))=location of ($x = 25 \wedge y = 17$) and direction of slope$\rightarrow\{-x, -y\}$. Thus, the decision-making process finds the location of ($x = 23 \wedge y = 11$) into all points visible from S as the suitable visible point illustrated in Fig. 3a and this suitable SUB_GOAL is saved to generate the final route.

Remark:

If all visible SUB_GOALs have the same ($i \vee j$) then it is necessary to choose the one that has $[\![(\min(j) \vee \min(i)]\!]$ in the orientation of the first region, which has the direction of slope as $\{-x, -y\}$, which means $\{i \rightarrow i\_\min \wedge j \rightarrow j\_\min\}$. Visible SUB_GOALs from S seen in Fig. 3a are described by using Eq. (7).

$$\begin{aligned} visibleSUB\_GOAL(1) &= \text{location of}(x = 23 \wedge y = 15) \\ visibleSUB\_GOAL(2) &= \text{location of}(x = 23 \wedge y = 14) \\ visibleSUB\_GOAL(3) &= \text{location of}(x = 23 \wedge y = 12) \\ visibleSUB\_GOAL(4) &= \text{location of}(x = 23 \wedge y = 11) \end{aligned} \tag{7}$$

Here, visible SUB_GOAL(4) is a suitable point to acquire the final route.

VI. Results

To examine the effectiveness of the proposed hybrid route planning algorithm, simulations in the MATLAB environment have been carried out for three maps which have different dimensions and three different scenarios have also been applied. Map dimensions are: $\forall$ Map $\in N\hat{}(+(m \times n)$ (units)), $\{m = 19, n = 28; m = 24, n = 24\}$ for Map_1 and Map_2 given in Figs. 4 and 5, respectively.

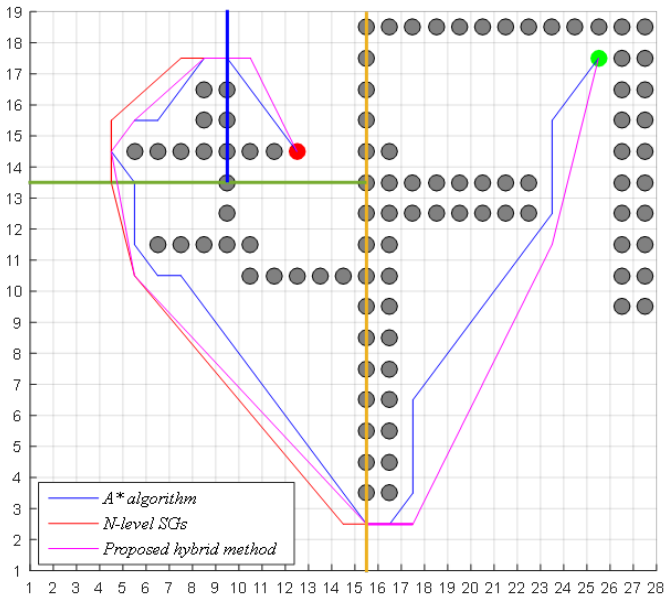

Fig. 4. Route planning in Map_1.

Starting point (S) and final point (F) are also defined as {S = (25,17), F = (12,14)}, {S = (23,5), F = (12,9)} for Map_1 and Map_2 with different scenarios. Routes for all scenarios are obtained by using A* algorithm, *N*-level SGs and the proposed hybrid method given in Figs. 4–6.

In Fig. 6, route planning of Map_3 for a different scenario is illustrated. Map_3 dimensions are:

$$\forall\{x, y\} \in \text{Map} \rightarrow (y = 1, \ldots, n; y = 1, \ldots, m) \rightarrow (m = 51, n = 51)$$

S and F are also defined as $\left\{\begin{matrix}\text{S}=(47,33)\\ \text{F}=(5,48)\end{matrix}\right\}$ for Map_3.

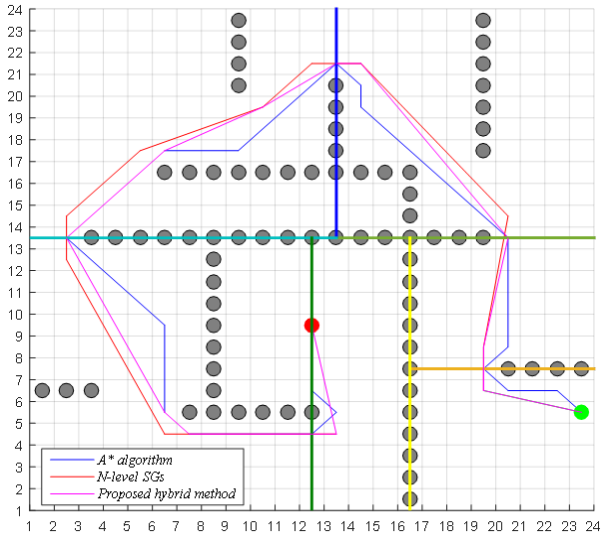

Fig. 5. Route planning in Map_2.

In the proposed method, simple SGs is implied for pre-searching in order to procure SUB_GOALs in an easy way. Nevertheless, *N*-level SGs is applied to show effectiveness of the proposed method in a fair manner. Each map is divided into the number of *t* sub-region by using the decision-making process, which benefits from Eq. (6).

There are 4, 7 and 9 subregions from S to F for three maps given in Figs. 4–6, respectively. Definitions about the first scenario have been specified with Eqs. (6) and (7) in the previous section. In the second and third scenarios (Figs. 5 and 6, the direction of subregions can be defined by using Eqs. (8) and (9), respectively.

$$\begin{aligned} &subregion(1) \wedge subregion(3) \wedge subregion(7) \rightarrow \\ &\qquad \{-x, +y\}, \{i \rightarrow i_{\max} \wedge j \rightarrow j_{\min}\} \\ &subregion(2) \wedge subregion(6) \rightarrow \\ &\qquad \{+x, +y\}, \{i \rightarrow i_{\max} \wedge j \rightarrow j_{\max}\} \\ &subregion(4) \rightarrow \{-x, -y\}, \{i \rightarrow i_{\min} \wedge j \rightarrow j_{\min}\} \\ &subregion(5) \rightarrow \{+x, -y\}, \{i \rightarrow i_{\min} \wedge j \rightarrow j_{\max}\} \end{aligned} \tag{8}$$

$$\begin{aligned} &subregion(1) \wedge subregion(3) \wedge subregion(7) \rightarrow \\ &\qquad \{-x, +y\}, \{i \rightarrow i_{\max} \wedge j \rightarrow j_{\min}\} \\ &subregion(2) \wedge subregion(6) \rightarrow \\ &\qquad \{+x, +y\}, \{i \rightarrow i_{\max} \wedge j \rightarrow j_{\max}\} \\ &subregion(4) \rightarrow \{-x, -y\}, \{i \rightarrow i_{\min} \wedge j \rightarrow j_{\min}\} \\ &subregion(5) \rightarrow \{+x, -y\}, \{i \rightarrow i_{\min} \wedge j \rightarrow j_{\max}\} \end{aligned} \tag{9}$$

Lengths of routes for three scenarios of maps with different dimensions are obtained as presented in Table IV. There are three methods: simple A* algorithm, SGs and the proposed hybrid method as shown in Figs. 4–6.

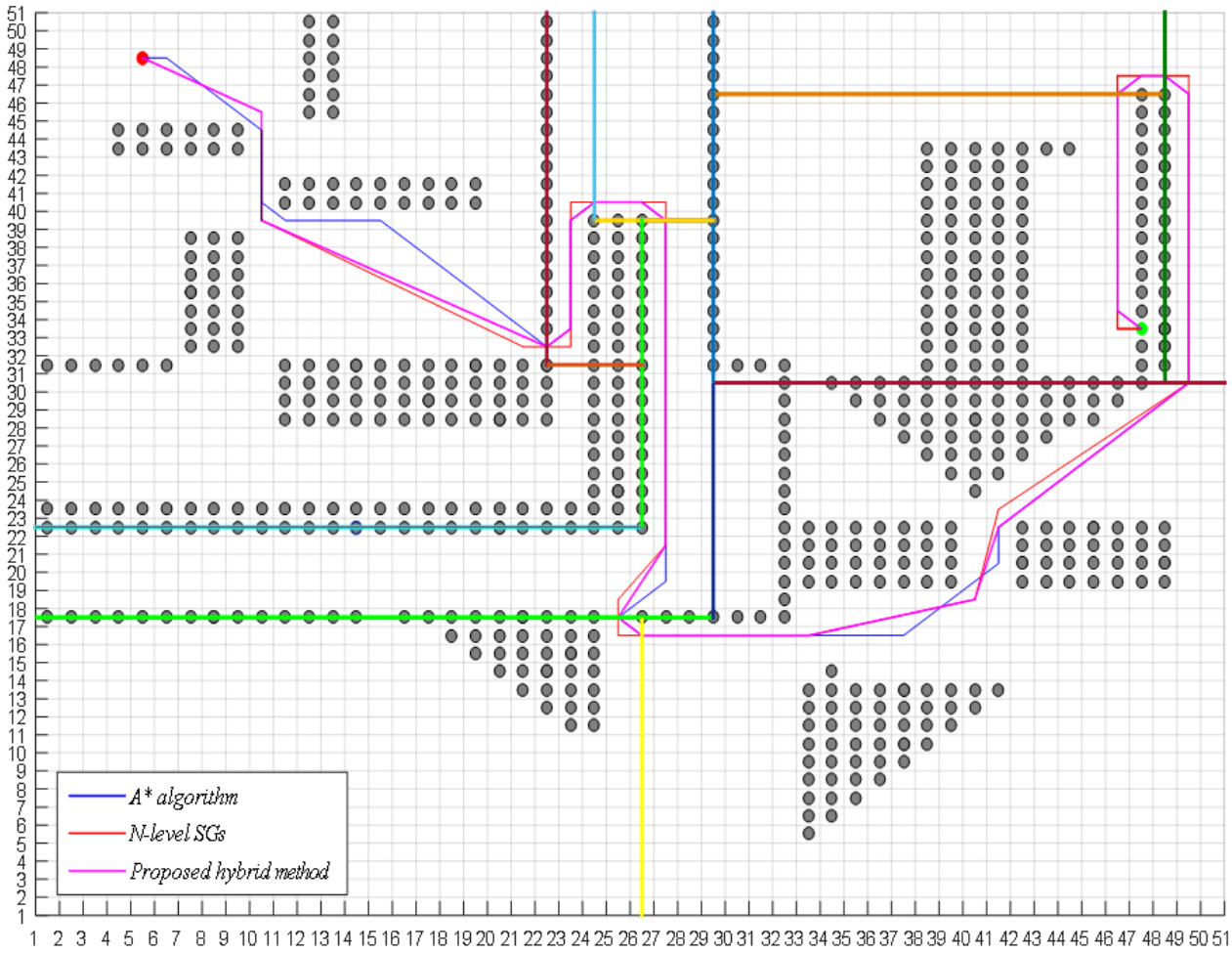

Fig. 6. Route planning for Map_3 with the different scenario.

TABLE IV

LENGTHS OF ROUTES FOR THREE SCENARIOS

Algorithm	Length of routes (units)		
	Map_1 with the first scenario	*Map_2* with the second scenario	*Map_3* with the third scenario
A* algorithm	47.36	58.36	125.67
N-level SGs	49.55	58.92	127.66
Proposed hybrid method	46.67	57.76	124.70

After the decision-making process, the obtained parts of routes for each subregion are combined to carry out final routes for each map. Lengths are obtained for final routes by using $g(n)$ function as mentioned in Eq. (1). The shortest lengths of routes are obtained by using the proposed hybrid method as 46.67 for Map_1, 57.76 for Map_2 and 124.7 for Map_3 and, therefore, the proposed method may be preferred for an operation route that is long and complicated such as video games.

VII. CONCLUDING REMARKS

In this study, a hierarchical structure, including hybrid approximations of simple methods: A* algorithm and SGs, has been proposed by using SUB_GOALs. Two layers have been included to organise pre-searching and final searching. The pre-searching layer consists of determining SUB_GOALs in order to save and apply at the final searching layer. Each map is divided into subregions by using the approach mentioned in Definition 1. The orientation of the route in each subregion is determined benefitting from the joint SUB_GOAL which is also needed to obtain division of subregions. Locations of visible SUB_GOALs obtained with LOS approach are effectual to discover joint SUB_GOALs. After, finding visible SUB_GOALs in the light of Definition 3, parts of the final route can be determined by combining visible SUB_GOALs which guarantee the shortest route. Three maps with different dimensions and scenarios are applied to show effectiveness of the proposed hybrid method. The shortest routes for three maps are obtained with the proposed method as compared with

A* algorithm and SGs. The future work is to improve the hybrid method by adding curves when a part of route is near to any obstacle to ensure the route planning for dynamic mobile robots.

REFERENCES

[1] Y. Fernando, M. A. Mustaqov and A. D. Megawaty, "Penerapan Algoritma A-Star Pada Aplikasi Pencarian Lokasi Fotografi Di Bandar Lampung Berbasis Android," *Jurnnal Teknoinfo*, vol. 14, no. 1, 2020, pp. 27–34. https://doi.org/10.33365/jti.v14i1.509.

[2] H. Yusnita, "Kinerja Dynamic Programming, Algoritma A Star dan Dijkstra Dalam Menentukan Rute Terpendek," Universitas Sumatera Utara, Master thesis, 2020. http://repositori.usu.ac.id/handle/123456789/28221

[3] R. F. Oktanugraha and S. R. Nudin, "Implementasi Algoritma A* (A Star) Dalam Penentuan Rute Terpendek Yang Dapat Dilalui Non Player Character Pada Game Good Thief," *Journal of Informatics and Computer Science*, vol. 2, no. 1, 2020, pp. 74–85.

[4] A. R. Soltani, H. Tawfik, J. Y. Goulermas and T. Fernando, "Path planning in construction sites: performance evaluation of the Dijkstra, A* and GA search algorithms," *Adv. Engineering Informatics*, vol. 16, no. 4, pp. 291–303, 2002. https://doi.org/10.1016/S1474-0346(03)00018-1.

[5] N. Delanoue, L. Jaulin and B. Cottenceau, "Counting the Number of Connected Components of a Set and its Application to Robotics," in J. Dongarra, K. Madsen, J. Waśniewski (eds) *Applied Parallel Computing. State of the Art in Scientific Computing*. PARA 2004. Lecture Notes in Computer Science, vol. 3732, pp. 93–101, 2006. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11558958_11.

[6] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: the case for A*," *International Journal of Geographical Information Science*, vol. 23. no. 4, pp. 531–543, 2009. https://doi.org/10.1080/13658810801949850.

[7] K. Kimura and A. Lipeles, "Fuzzy controller component," U. S. Patent 14,860,040, December 14, 1996.

[8] A. Stentz, "Optimal and efficient path planning for partially-known environments," in M. H. Hebert, C. Thorpe, A. Stentz (eds) *Intelligent Unmanned Ground Vehicles*. The Springer International Series in Engineering and Computer Science (Robotics: Vision, Manipulation and Sensors), vol. 388. Springer, 1997. https://doi.org/10.1007/978-1-4615-6325-9_11.

[9] S. Koenig and M. Likhachev, "Fast Replanning for Navigation in Unknown Terrain," *Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005. https://doi.org/10.1109/TRO.2004.838026.

[10] S. Yetkin, "Application of two-dimensional path planning algorithms for the robot fish," Fırat Üniversitesi, Master thesis, 2016. http://hdl.handle.net/11508/17991.

[11] L. Liu, S. Luo, F. Guo and S. Tan, "Multi-point shortest path planning based on an improved discrete bat algorithm," *Applied Soft Computing*, vol. 95, 2020, 106498. https://doi.org/10.1016/j.asoc.2020.106498.

[12] E. Masehian and D. Sedighizadeh, "Multi-objective PSO-and NPSO-based algorithms for robot path planning," *Advances in Electrical and Computer Engineering*, vol. 10, no. 4, pp. 69–76, 2010. https://doi.org/10.4316/aece.2010.04011.

[13] N. A. Elkhateeb and R. I. Badr, "Novel PID Tracking Controller for 2DOF Robotic Manipulator System Based on Artificial Bee Colony Algorithm," *Electrical, Control and Communication Engineering*, vol. 13, no. 1, 2017, pp. 55–62. https://doi.org/10.1515/ecce-2017-0008.

[14] T. Uras, S. Koenig and C. Hernandez, "Subgoal graphs for optimal pathfinding in eight-neighbor grids," *ICAPS*, pp. 224–232, 2013.

[15] T. Uras and S. Koenig, "Identifying hierarchies for fast optimal search," *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 878–884, 2014.

[16] D. Nussbaum and A. Yörükçü, "Moving target search with subgoal graphs," *Twenty-Fift International Conference on Automated Planning and Scheduling*, Jerusalem, Israel, pp. 179–187, 2015.

[17] K. Xu, Y. Hu, Y. Zeng, Q. Yin and M. Yang, "Improving the Scalability of the Magnitude-Based Deceptive Path-Planning Using Subgoal Graphs," *Entropy*, vol. 22, no. 2, 2020. https://doi.org/10.3390/e22020162.

[18] L. Zuo, Q. Guo, X. Xu and H. Fu, "A hierarchical path planning approach based on A* and least-squares policy iteration for mobile robots," *Neurocomputing*, vol. 170, pp. 257–266, 2015. https://doi.org/10.1016/j.neucom.2014.09.092.

[19] J. Peng, Y. Huang and G. Luo, "Robot path planning based on improved A* algorithm," *Cybernetics and Information Technologies*, vol 15, no. 2, pp. 171–180, 2015. https://doi.org/10.1515/cait-2015-0036.

[20] F. Duchon, et. al. "Path planning with modified A star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014. https://doi.org/10.1016/j.proeng.2014.12.098

[21] I. S. AlShawi, L. Yan, W. Pan and B. Luo, "Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm," *IEEE Sensors Journal*, vol. 12, no. 10, pp. 3010–3018, 2012. https://doi.org/10.1109/JSEN.2012.2207950

Gonca Ozmen Koca was born in Elazig in 1980. She received her Master's degree from Firat University, Institute of Science, Department of Electronics and Computer Education in 2005. She obtained her PhD degree from the Electric and Electronic Engineering Department of Firat University. She is currently an Associate Professor at the Control Division of Mechatronics Engineering Department, Technology Faculty, Firat University. Her research areas cover control systems, path planning, biomimetic systems, intelligent control techniques, optimization methods.

E-mail: gonca.ozmen@gmail.com
ORCID iD: https://orcid.org/0000-0003-1750-8479

Seda Yetkin was born in Elazig in 1992. She graduated from Fırat University, Faculty of Engineering, Mechatronics Engineering and Mechanical Engineering in 2014. She received her Master's degree from Firat University (2016), Faculty of Technology, Department of Mechatronics Engineering. Now she is a PhD student at the same department.
She is currently working as a Lecturer at Bitlis Eren University, Vocational School of Technical Sciences, Department of Biomedical Device Technologies. Her research areas cover robotics, control systems.

E-mail: sedayetkin23@gmail.com
ORCID iD: https://orcid.org/0000-0001-9685-1376