Learning traversability map of different robotic platforms for unstructured terrains path planning

1st Paolo Arena, Carmelo Fabrizio Blanco, Alessia Li Noce, Salvatore Taffara
Dipartimento di Ingegneria, Elettrica, Elettronica e Informatica,
Università degli Studi di Catania
Viale Andrea Doria 6, 95125 Catania, Italy
paolo.arena@unict.it, fab.blanco@gmail.it, a.linoce@gmail.it, s.taffara@tiscali.it

2nd Luca Patanè
Dipartimento di Ingegneria,
Università degli Studi di Messina
Contrada di Dio, 98166 Messina, Italy
lpatane@unime.it

***Abstract*—This paper aims to propose an innovative method to obtain the traversability maps of unstructured environments and the best path between two points on the basis of the specific characteristics of the robots that has to perform a given mission. Taken in consideration a robot team that have to traverse an assigned terrain, the peculiar capabilities of each robot are underlined in a dynamic simulation environment and then embedded into a neural network finally used as a robot model for the generation of the traversability maps. On the basis of the obtained results, the best robot within the team (wheeled, legged, hybrid) can be selected. The proposed strategy, together with the obtained simulation results, are presented, carefully analyzed and then compared.**

***Index Terms*—path planning, traversability map, neural network, dynamical simulation.**

I. Introduction

Path planning in Robotics is an interesting topic in all the tasks in which the achievement of different points in an area is performed using roving robots. A strategy to select the best among all the possible paths is important to guarantee a safe behavior for the mobile robot performing the assigned task. One of the applications of the described method regards the topic of the positioning of sensors for landslides monitoring [1] in which the prediction of optimal routes in highly unstructured terrains plays a fundamental role. Some novel approaches involve motor-skill learning and bio-inspired methods to make the robots able to improve their climbing capabilities in front of novel terrain conditions [2]. Other approaches consider multiple robots available with different characteristics and the optimal configuration can be selected depending on the task. This concept can be extended considering a target area where different robotic structures are deployed and each one can accomplish part of the whole task depending on its particular capabilities. Moreover, when needed, some of them can self-assemble to create a new robot to solve an otherwise impossible task. This mechanism has biological fundamentals as described by Dorigo and colleagues [3] that introduced the biological swarm intelligence dealing with self-organizing systems in which a complex behavior can be possible only though the cooperation among multiple members of the team driven by a collective intelligence, as happens in ant colonies, in bird swarms or in schools of fish.

Dynamic simulators and neural networks can be used together to predict which areas of an environment are traversable using mobile ground robots, in order to plan feasible paths. An approach based on Convolutional Neural Network was adopted in [4] to predict whether a robot will be able to traverse a patch taking in input an image representing the height map of a terrain patch. However, this approach is not feasible when longer paths are to be planned. Images locally acquired by the ground robot cannot be useful in this case. Instead, if the images, covering a suitably large area, are acquired from drones, it is asked to derive traversability maps useful for planning the whole path of the robot from a starting to a target position, which is impossible to be derived with on board visual sensors, with the additional drawback to be stuck in local minima. In literature most of the works concerning the topic of optimal route planning are dealt with an on-line approach, (i.e. taking into consideration real sensors and real robots). Mobile robots, using GPS and sensors, can move in an area of which they know the characteristics a priori or in an area that needs to be explored for the first time. The traversability information can be achieved in real time using a path detection method that mixes together 3D mapping and visual classification trying to learn, in real time, the actual road characteristics [5]. In this case the robot does not know the geometry of the terrain, but the information are acquired in run time. The terrain analysis can be developed also using sensors through which it is possible to separate traversable regions for autonomous driving using cameras and LIDARs. In this case two maps can be distinguished: one is the recognition map and the other is the terrain modelling map; these have to be built independently and then fused [7]. Using a computer vision unit it is possible to extract the colour, geometric, and texture parameters of a local area of the environment. A single image is than converted into a binary image from which it is possible to individuate the smoothness of the area representing the capability of the robot to travel that area [8].

The innovative contribution of this paper is related to the possibility to obtain the traversability maps and the optimal path between two or more points with an a priori analysis of the available terrain map. The idea is to limit the employment of real robots that are instead simulated in a dynamic environment to acquire information about their moving

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

capabilities. The acquired data are finally embedded into a neural network that memorizes and interpolates them. This strategy allows us to obtain specific models for each robot taken into account. These models are built on the basis of the real terrain morphology used as the input information and then provide as output the traversability of the selected area. In [9], to face with the traversability task, all places of the environment are classified into occupied by an obstacle, free, or not mapped, using an Occupancy-Elevation Grid where each cell stores a probabilistic estimation of occupancy and other terrain characteristics. In our work instead, to ensure that the optimal path matches the physicality of the robot, this latter is left to explore the acquired environment and some of its characteristics are detected, such as the success or failure in reaching a given point. Therefore a black box model of the robot can be learned to identify the areas in which it fails to go and those in which it moves fluently and, from these information, the optimal paths. This strategy allows to extract a robot traversability model that can be extended to other terrains, exploiting the generalization capabilities of neural structures. The traversability map can so be considered as an affordance map, i.e. an implicit model that can represent the robot capabilities in front of different types of terrain patches. The topic related to the choice of the best path among all the possible paths is really interesting in view to create an autonomous navigation system. The common path planning algorithms, by using the global information of the terrain, can find the optimal paths but it is really difficult to build up the global distribution of the terrain information. In other words, it is very complex to receive the terrain distribution in advance in some special applications [10]. Another method for the estimation of the best path is the usage of algorithms for the generation of points-cloud thanks to which it is possible to obtain terrain information such as roughness, slopes and breakline [11]. In this work the selection of the best paths is realized thanks to the exploitation of a model that will represent the robot's capabilities in dealing with a given type of terrain. Moreover the paper aims at experimentally show how simple and traditional neural network structures, like Multilayer Perceptron, are able to solve complex environment traversability problems still showing good performance in front of simple and shallow architectures. So the present work aims to answer the following question: given one or more terrains and a series robots, which paths within the environment can them better cope with in terms of their current capabilities or improving them using coupling mechanisms between robots? This paper is organized as follow: section II shows the neural network approach applied to the robotic exploration and the generation of the robotic models different for each platform. Section III deals with the topic related to the mobile robots used and the analysis of the models in terms of predictive reliability. In section IV the algorithm through which is possible to obtain the best path in a map is presented and different test are shown. Finally conclusions are reported in Section VI.

II. A NEURAL NETWORK APPROACH FOR TERRESTRIAL INSPECTION

In order to obtain the neural model of the robot's capabilities, this latter is placed in an unknown environment, divided into tiles of fixed dimensions, that it will have to explore. The selected robot is randomly placed in different areas from which some morphological characteristics, that will be analyzed in the following paragraph, are recorded during its movements. Also the information related to the success, coded as bit 0, and failure, coded as bit 1, in passing from a tile to another along one of the possible directions are recorded. These information are given as input and target respectively to a neural network that is trained to classify as traversable or not the passage between two adjacent tiles. After completing the learning phase, a first task which was assigned to the network, to test its generalization capabilities, was to perform a testing phase using the same map as in the learning campaign, but requesting the network to predict traversability between tiles arranged in the diagonal directions: north-west, north-east, south-west and south-east. The task was also useful for the best arranging of the inputs to the networks, as it will be detailed in the following. The successful results obtained in this phase gave the possibility to generalize the application of the neural structure also to different terrains. A further advantage of using a neural network consists in training the network with data obtained from a dynamic model of the robot and then performing a fine tuning using information drawn from the real behavior of the robot. Typically these real data are more expensive and then used only to refine the already trained model.

To address the difference in scales among features, the so called z-score normalization is employed for each feature descriptor across all the samples. Namely the mean from each value is subtracted and divided by the standard deviation of that particular descriptor. In doing so, the original data were turned into a standard scale for each feature [12]. It was verified that this dataset normalization increases the neural network performance.

In Fig. 1 a flow chart representing the complete algorithm is shown. It starts with the selection of a representative map and the extraction of its morphological characteristics. The successive step consists in letting a set of simulated robots move through the previously selected map and using the data recorded by the dynamic simulator, for each robot, to train a neural models (NN) specific for each robot, embedding their motion capabilities. The next step consists in the selection of a target map, i.e., the map where each neural network, representing a robot model, will be run to provide the traversability among specific points in the target map. This will provide a dataset for generating eight traversability maps used as input of an optimal path algorithm that will provide an optimal route among those specific point set, solving the assigned motion task.

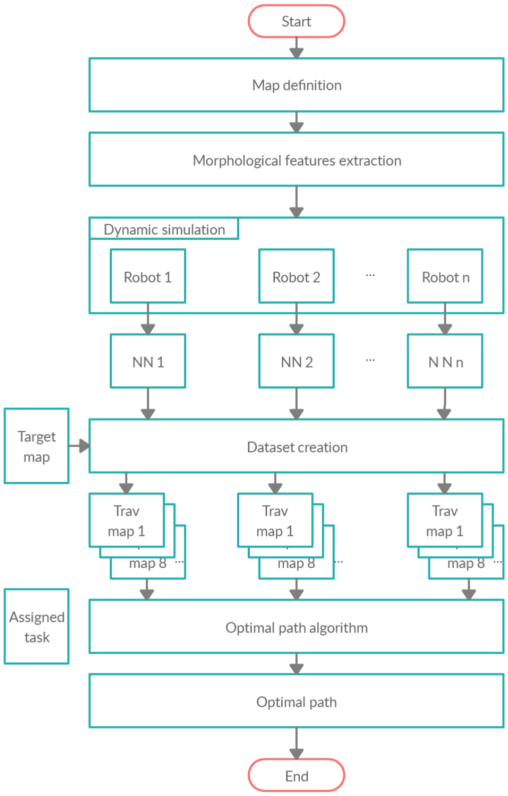
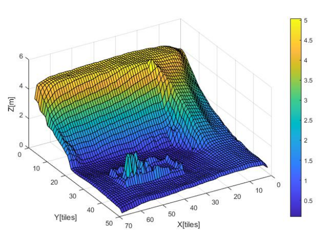


Fig. 1. Flow chart of the implemented algorithm.

A. Feature extraction from the terrain

In order to obtain the inputs of the neural network we consider the height map of the terrain reported in Fig. 2(a) obtained using drones [6]. The image is in the GeoTIFF standard which allows the embedding of georeferencing information. A sampled version of the terrain was adopted (Fig. 2(b)) so that it is possible to take into account the geometry of a set of tiles and to consider the following characteristics: the average height difference, the average slope, the average roughness and the maximum height between two consecutive tiles [11]. The choice to use the average values for each input is adopted to reduce the number of inputs in the Neural Network. The modularity of the algorithm gives the possibility to set the tile physical dimensions based on the physical size of the robot. In our case each tile is equivalent to 0.65 meters.

(a)



(b)

Fig. 2. Terrain used as testbed: (a) Aerial image and (b) 3D reconstruction.

The height differences are calculated considering the differences between the quotes of a tile, given by the mean of the quote of each corner, and the quotes of its near tiles. In this way eight maps of different heights, one for each direction, are generated. The slope is of a tile, with respect to the tangential movement of the robot, was also used. Different methods can be adopted to calculate the roughness, for example in [11] the standard deviation method is taken into account, fixing a threshold. The least-squares plane fitting [13] is the method used in our paper: this gives the possibility to obtain a plane starting from a point cloud that represents the map quotes. The maximum quote in the set of roughness is selected as the maximum height that the robot has to face with while moving.

III. Methods

In this section the software tools and the robotic structures used to test the whole algorithm are described. Moreover the results of the developed neural models are analyzed and compared presenting their outcome in terms of traversability maps.

A. Framework

The tools employed to develop the algorithm are Matlab and Vrep. The control system used to drive the robots is implemented in Matlab. Here, the environmental characteristics extracted from a terrain are used as inputs for training a neural model. The outputs of this latter are useful to generate the traversability maps used in the path planning algorithm. Vrep is a dynamic simulation environment [14] in which the robots and terrains are imported to acquire the data needed to obtain the traversability maps .

B. The robotic structures

To evaluate the proposed method, different robotic structures are considered: wheeled, legged and hybrids.

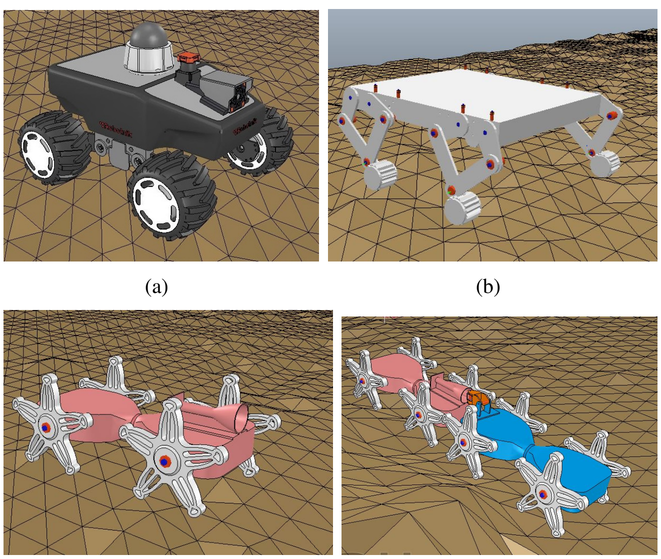
The Robotnik Summit-XL (Fig. 3(a)) is a four wheeled platform with skid-steering kinematics. Each wheel integrates a brushless motor with gearbox and encoder. It has two possible kinematic configurations and the omnidirectional configuration mounts mecanum wheels on an independent suspension system. The mecanum wheels can be easily replaced by conventional wheels, thus allowing easy switch from the indoor omnidirectional configuration to the versatile skid-steering configuration, both indoors and outdoors [15].

The 5BSPL robot (Fig. 3(b)) is a quadruped-like robot adopting a particular kinematic structure called “5-bar symmetric planar linkage” that is composed by a set of 5 links connected in a closed chain by 5 revolute joints, of which two are actuated by brushless DC electric motors (BLDC) and two are passive, made up by ball bearings. This robot is inspired by the Minitaur quadruped configuration [16].

The Asguard robot (Fig. 3(c)) is another hybrid prototype with a very interesting structure composed by four rotating wheels modeled as a five-pointed star. Each point of the star serves as a leg during locomotion and is controlled using bio-inspired central pattern generators (CPGs). This robot is highly agile and fast on flat ground and, at the same time, should be able to deal with very rough terrains, e.g. rubble, gravel, and even stairs [17].

All the considered robots have comparable dimensions (about 60 cm) and the same moving speed (about 10 centimeters/second).

To improve the robot locomotion capabilities, following a bio-inspired strategy, it is possible to reproduce coupling mechanisms. Just think of snails having a reproductive organ or octopuses that to attract females perform a courtship ritual creating a mechanical connection with the partners [18]. To test the improvements that a coupling mechanism can give to the robotic performances two Asgard robots (Fig. 3(d)), a male and a female exemplaries, are connected through a latching mechanism formed by a harpoon installed in the male robot that remains locked inside a slot in the female one. Details about the performance of the assembling mechanism are outside the focus of the paper. Here an already assembled structure will be analyzed. What is expected is that the coupled structure is more stable and capable to overcome higher obstacles than the single system.



(a) (b) (c) (d)

Fig. 3. Different robotic structures taken into consideration for the simulation: (a) Robotnik Summit-XL, (b) 5BSPL, (c) Asguard, (d) Cooperative Asguard.

C. Neural model of the robot locomotion skills

In this paragraph two learning strategies are evaluated for the elaboration of the traversability maps. The two structures are the well-known Multilayer Perceptron (MLP) and the Decision Tree Learning (DTL) that are used to correctly classify if the transition between tiles is feasible or not. In fact, it will be experimentally shown how using just a shallow network architecture and a decision tree structure, instead of the more recent, high performing but more complex deep networks, good results can be obtained, which open the way to very fast learning procedures and still good generalization capabilities. For the goal to be pursued the dataset, composed of about 2000 patterns, obtained from the robot simulation, is divided in learning (80%) and test data (20%).

Decision tree learning is one of the predictive modeling approaches used in statistics, data mining and machine learning whose goal is to build a decision tree that is consistent with a given data set and this typically means to chose the smallest decision trees [19]. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the tree leaves). The Multilayer Perceptron is the well known neural network model composed by an input layer that in our case is composed by four inputs (the average height difference, the average slope, the average roughness and the maximum height between two consecutive tiles), one hidden layer and an output layer. The purpose is to map the set of inputs data to a set of target data using a supervised learning technique called backpropagation.

The convergence times of both algorithms are in the order of seconds on a desktop PC with the following characteristics: quad-core processor 1.4 GHz, 4GB Ram, without using any GPU acceleration on board. This can give an idea of the possible speed up, using dedicated hardware.

The parameters used during the test to set the two structures are reported in tables I for the MLP and in II for the TREE.

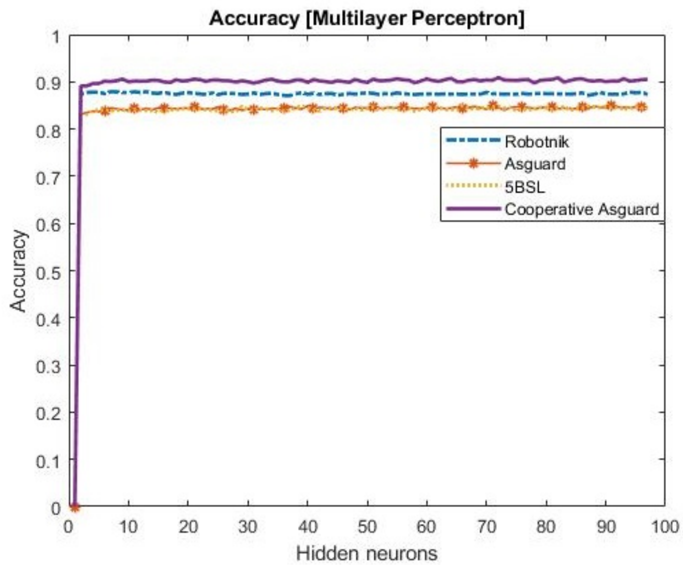
TABLE I
MULTILAYER PERCEPTRON SETTINGS.

MLP	
Attivation function (for hidden and output neuron layers)	Sigmoid tangent
Curve fitting algorithm	Levenberg-Marquardt
Performance function	Mean Squared Error
Hidden neurons	Range [1 100]

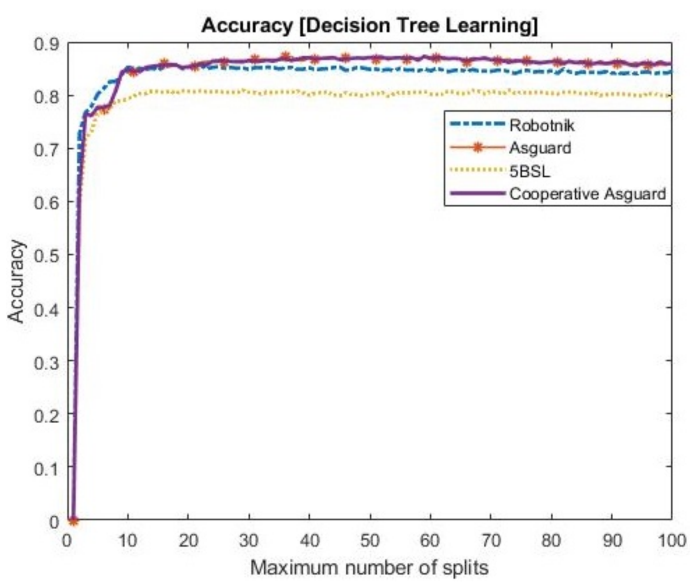
TABLE II
DECISION TREE LEARNING SETTINGS.

TREE	
CrossVal	on
Max number of splits	Range [1 100]

In order to evaluate the algorithm's ability to predict the correct traversability information the accuracy index is used. Considering the MLP case, in (Fig. 4(a)) the accuracy values when the number of hidden neurons changes from 1 to 100 are reported. To have a reliable statistical value, for each hidden neuron, the accuracy value is given by the average of one hundred learned models, each time with a different randomization between learning and test patterns. Considering the DTL method the accuracy is related to the maximum number of splits of the tree, using the average values as seen in the MLP case (Fig. 4(b)). Overall the accuracy values for each structure are really satisfactory being just below the 90 percentage; in particular the best accuracy values are obtained using the MLP, even if the difference with the DTL method is not so marked. This data analysis certifies suitability for the subsequent steps in which are generated the traversability maps because it proves that the network is capable of adequately predicting its outputs.



(a)



(b)

Fig. 4. Accuracy results obtained with the MLP (a) and DTL methods (b).

In tables III and IV the data related to the best accuracy (the degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard, so the closeness of the measurements to a specific value), sensitivity (the proportion of actual positives that are correctly identified as such), specificity (the proportion of actual negatives that are correctly identified as such) and the relative standard deviations for each robot adopting the two previously discussed methods are reported for the best network configuration.

TABLE III
MLP STATISTICAL PERFORMANCE

	Multilayer Perceptron			
	Accuracy	Sensitivity	Specificity	Hidden neurons
Robotnik	87.93 ±2.11%	86.34±2.84%	83.09±3.03%	10
5BSPL	84.97±2.51%	91.45±4.10%	78±3.57%	69
Asguard	84.90±1.99%	94.35±2.19%	78.92±2.79%	43
Coop. Asguard	90.89±2.01%	95.11±1.77%	84.55±2.96%	75

TABLE IV
DLT STATISTICAL PERFORMANCE

	Decision Tree Learning			
	Accuracy	Sensitivity	Specificity	Max splits
Robotnik	85.39±2.07%	89.52±3.11%	82.18±3.55%	17
5BSPL	80.30±2.19%	84.14±3.21%	76.30±3.55%	72
Asguard	86.77±1.73%	91.44±1.97%	79.73±3.34%	36
Coop. Asguard	86.70±1.79%	90.81±2.03%	80.32±3.55%	58

D. *Traversability maps*

The outcome of the learned models are the traversability maps with respect to the four cardinal directions (north, east, south and west) and the four diagonals among them. Analyzing the accuracy performance related to the previously seen methods, it was decided to use the MLP architecture due to the better accuracy values obtained compared with the other one.

The next step consisted in further processing the neural network outputs, forming the traversability map's elements. These were scaled to serve as inputs for an optimal path algorithm. The traversability matrix structure was built as follows: each element represents a physical point in the map and its value is related to the possibility for the robot to reach that point coming from the opposite direction. In particular the maximum value of the tiles, related to an unreachable position, is fixed to 0.1, whereas the minimum value is set to 10^5. For example, considering the north matrix and an element of value equal to $0.1(10^5)$, this means that specific point the cell is referring to is reachable from the south cell adjacent to the selected one. This structure is repeated for each of the eight maps all values. Using this strategy it is possible, applying appropriate algorithms, to select the best path from a point to another. The traversability map values are obtained from the average of one hundred iterations of the algorithm, each time changing the dataset learning and test splitting. As example, in Fig. 5, the eight traversability maps related to the Asguard robot are shown. The colors are related to the numeric value assigned to tiles; the areas in which the color is yellow are those in which the robot can not pass, the others tending to the blue are those in which it can travel in a safer way. The comparison with Fig. 2 reveals that the traversability maps represent the mirror of the robot capabilities in reaching specific points in the real map.

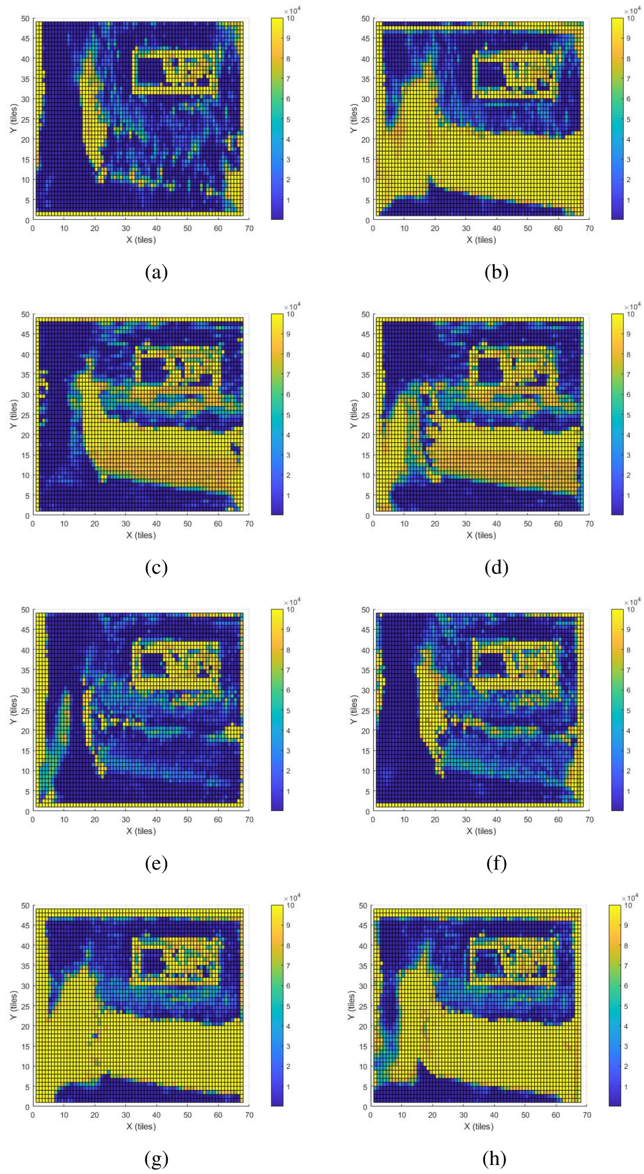


Fig. 5. Traversability maps obtained for the Asguard robot in the eight direction of motions: (a) North, (b) South, (c) East, (d) West, (e) North-East, (f) North-West, (g) South-East, (h) South-West.

In Fig. 6 the results of the north traversability maps for the four previously mentioned robots are reported. Comparing, for example, the maps related to Cooperative Asguard with the others it is possible to see that this robot can cover a higher portion of the map to prove the fact that, thanks to the coupled configuration the whole structure assumes a higher capability to overcome the obstacles. Therefore this robotic structure manages to find pass through gaps along the path which are more difficult for other robots to be faced with. The prize to be paid is the need of a more complex and expensive solution that has to be adopted if simpler structures are not adequate to fulfill the assigned task.

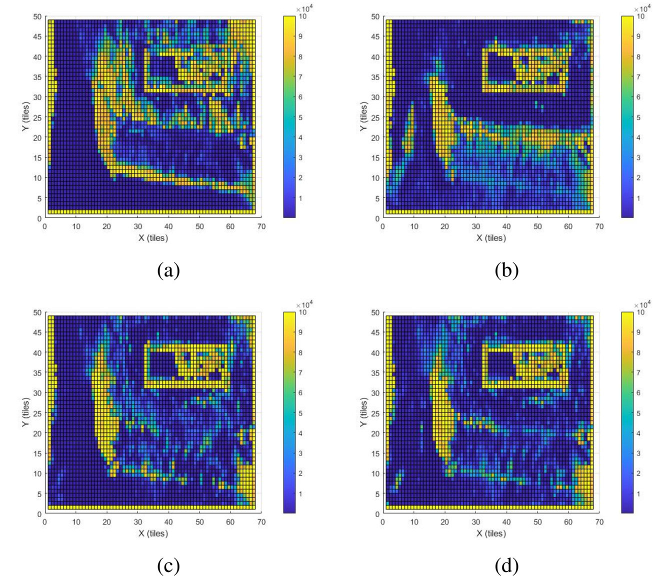


Fig. 6. Traversability maps obtained for the four robots considering only the north direction: (a) Robotnik, (b) 5BSPL, (c) Asguard, (d) Cooperative Asguard.

IV. Optimal path estimation

The shortest path problem can be now taken into account using as input the traversability maps previously generated. We adopted the Dijkstra's algorithm that is a shortest path solver also used in the optimization of routes taken by robots [20] and Automatic Guided Vehicle (AGV) [21] According to the literature, it can be considered the most classical and mature algorithm among all the others used for the optimization of paths in graphs [22]. From the height map of the environment the oriented and weighted graph is obtained in which each node is connected to its neighborhoods so that it is possible to compute the shortest path from a source node to the target one [23]. The weights for the eight directions are given by the relative traversability maps: high (low) weights along a given direction reflect a less (more) convenient path for traversability.

In Fig. 7 three optimal paths obtained using the Asguard's traversability maps are shown. From this view it is possible to see that the most relevant obstacles, like the central fenced structure and the cliff on the left, are areas avoided as results from the path calculation.

In order to test the different robotic models the same task, in terms of initial and final positions, has been assigned and the different paths followed by each robot are depicted in Fig. 8. As expected each robot follows its own route depending on its traversability capabilities; the better the robot ability to find passages along the way, the less tortuous the path taken. Table V reports the number of tiles that each robot has to cross to arrive to the end of the assigned path. It is possible to notice that the mechanical solution to couple two Asguard robots instead of using a single one is a winning strategy in terms of traversability capacity for the assigned task.

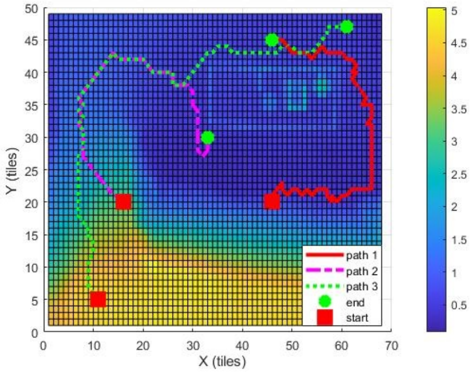
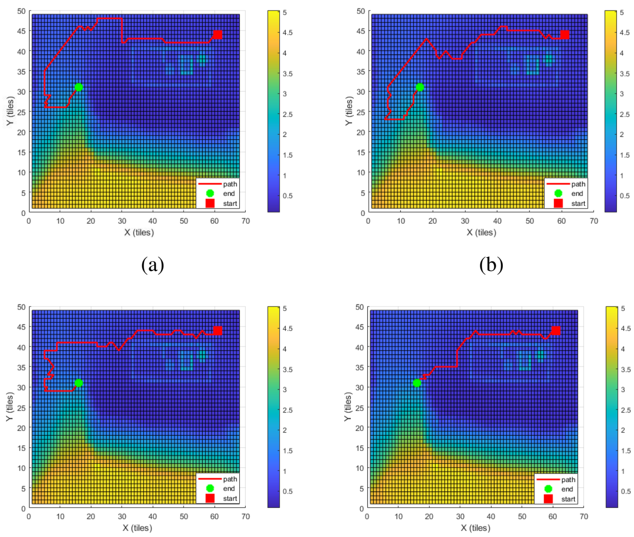


Fig. 7. Examples of optimal paths, for the Asguard robot, changing the start and target position.



(a) (b)

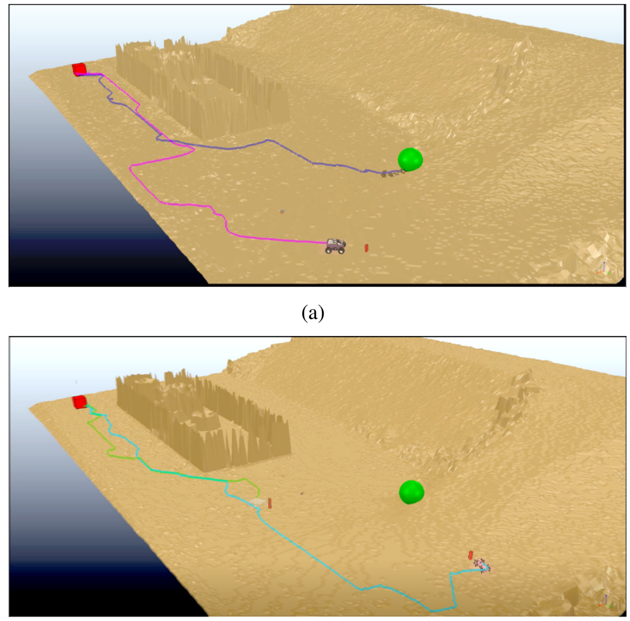
(c) (d)

Fig. 8. Optimal paths performed by the robots with the same start and target position: (a) Robotnik, (b) 5BSPL, (c) Asguard, (d) Cooperative Asguard.

TABLE V
NUMBER OF TILES COVERED BY EACH ROBOTS FOR THE SIMULATION IN FIG. 8.

	Crossed tiles
Robotnik	91
5BSPL	82
Asguard	99
Coop. Asguard	52

In Fig. 9 the results obtained in a typical run of the dynamic simulation environment when all robots are requested to reach a given target, are depicted. During the simulation, when the Cooperative Asguard reaches the target position (green sphere), the other three robots are still moving to complete their routes.



(a)

(b)

Fig. 9. Trajectories followed by the considered robots to reach the target (green sphere) starting from the same initial point (red cuboid): (a) Robotnik and Cooperative Asguard, (b) 5BSPL and Asguard. When the Cooperative Asguard reaches the target, the other robots are still far away from it.

V. CONCLUSION

The need to have an algorithm capable, in a reasonable short time, to foresee the best path that a particular type of robot can follow in a given terrain is certainly an ambitious goal, in particular when performing field tests is both complex and economically expensive. One of the most interesting topics in which such a method finds application is the landslide monitoring in which it is necessary to know in advance the best paths that a robot can tackle to prevent damages during the path or to avoid to trigger the subsidence of the ground itself. In this work we investigated the possibility to use a neural network to learn the capabilities in moving on complex terrain morphologies. A set of high level features have been collected from height maps and used to learn neural structures able to classify if a specific area of the map is traversable for a selected robot. The learning patterns were obtained using a dynamic simulation environment where the robot capabilities can be easily tested and evaluated. Two different solutions based on a multilayer perceptron and a binary tree structure were compared showing their statistical performances in terms of classification accuracy. The MLP gave, from the beginning, interesting generalization capabilities, since after learning to model traversability in the cardinal directions of the map, provided the same performance when tested in the diagonal directions. The traversability maps, that can be generated adopting these structures, can be used to evaluate the optimal path between a start and target position. Finally, depending on the assigned task, the best performing robotic structure can be selected to be used, trying to emphasize the peculiarities of each structure in handling the different terrain configurations. In particular, the reported simulation results show that the capabilities of a single robot can be improved using assembly mechanisms, inspired by the animal world, through which a new more versatile structure can be obtained.

ACKNOWLEDGEMENTS

This research was funded by MIUR project CLARA - Cloud platform for LAndslide Risk Assessment grant number SNC_00451.

REFERENCES

[1] Patanè, L, "Bio-Inspired Robotic Solutions for Landslide Monitoring", Energies, April 2019.

[2] E. Arena, P. Arena, R. Strauss, L. Patané (2017). "Motor-skill learning in an insect inspired neuro-computational control system". Frontiers in Neurorobotics, vol. 11, ISSN: 1662-5218, DOI: 10.3389/fnbot.2017.00012

[3] M. Dorigo, M. Birattari X. Li, M. L. Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle, "Swarm Intelligence", 10th International Conference, ANTS 2016 Brussels, Belgium, September 7-9 2016.

[4] R. O. Chavez-Garcia , J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning Ground Traversability From Simulations", IEEE Robotics and Automation letters Vol. 3, NO. 3, July 2018.

[5] H. Roncancio, M. Becker, A. Broggi and S. Cattani, "Traversability Analysis Using Terrain Mapping and Online-trained Terrain Type Classifier", 2014 IEEE Intelligent Vehicles Symposium (IV), Dearborn, Michigan, USA, June 8-11 2014.

[6] D. C. Guastella, L. Cantelli, C. D. Melita and G. Muscato, "A Global Path Planning Strategy for a UGV from Aerial Elevation Maps for Disaster Response", 9th International Conference on Agents and Artificial Intelligence, January 2017.

[7] J. Sock, K. Kwak, J. Min, and Y. Park, "Probabilistic Traversability Map Building for Autonomous Navigation", 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), Kintex, Gyeonggi-do, Korea October 22-25 2014.

[8] A. Andrakhanov, and A. Stuchkov, "Traversability estimation system for mobile robot in heterogeneous environment with different underlying surface characteristics", CSIT 2017, Lviv, Ukraine, 05-08 September 2017.

[9] A. Souza, and L. M. G. Gonçalves, "Path Planning Based on Traversability Evaluation from Occupancy-Elevation Grid Maps", 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), November 2018.

[10] Y. Guo, A. Song, J. Bao, and H. Zhang, "Optimal Path Planning in Field Based on Traversability Prediction for Mobile Robot". International Conference on Electric Information and Control Engineering, Wuhan, China, 15-17 April 2011.

[11] D. Mongus, and S. Jurič, "Generation of traversability maps based on 3D", 2019 Conference on Next Generation Computing Applications (NextComp), Mauritius, 19-21 September 2019.

[12] H. Bisgin, T. Bera, H. Ding, H. G. Semey, L. Wu, Z. Liu, A. E. Barnes, D. A. Langley, M. Pava-Ripoll, H. J. Vyas, and W Tong, J. Xu, "Comparing SVM and ANN based Machine Learning Methods for Species Identification of Food Contaminating Beetles", Nature. Available online: https://www.nature.com/scientificreports (accessed on 30 December 2019)

[13] C. Ye, "Navigating a Mobile Robot by a Traversability Field Histogram", IEEE Transactions on systems, man, and cybernetics-part B: Cybernetics, vol. 37, no. 2, April 2007.

[14] E. Rohmer, S. P. N. Signgh, and M. Freese, "V-REP: a Versatile and Scalable Robot Simulation Framework," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 06 January 2013.

[15] Robotnik. Available online: https://www.robotnik.eu/mobile-robots/summit-xl/ (accessed on 25 December 2020).

[16] G. Kenneally, A. De and D. E. Koditschek, "Design Principles for a Family of Direct-Drive Legged Robots", IEEE Robotics and Automation Letters, vol. 1, pp. 900-907, July 2016

[17] M. Eich, F. Grimminger, S. Bosse, D. Spenneberg, and F. Kirchner, "Asguard: A Hybrid - Wheel Security and SAR-Robot Using Bio-Inspired Locomotion for Rough Terrain". Proceedings ROBIO 2008, 774-779.

[18] National Geographic. Available online: https://www.nationalgeographic.com/news/2015/03/150310-snails-reproduction- sex-animals-science-evolution/ (accessed on 10 January 2020).

[19] F. S. Hillier, and G. J. Lieberman, Introduction to stochastic models in operations research. England: McGraw-Hill College, 1990.

[20] S. A. Fadzli, S. I. Abdulkadir, M. Makhtar, and A. A. Jamal, "Robotic Indoor Path Planning using Dijkstra's Algorithm with Multi-Layer Dictionaries", 2015 2nd International Conference on Information Science and Security (ICISS), Seoul, South Korea, 14-16 December 2015.

[21] D. Li, and K. Niu, "Dijkstra's algorithm in AGV", 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, China, 9-11 June 2014.

[22] D. Fan, and P. Shi, "Improvement of Dijkstra's Algorithm and Its Application in Route Planning", Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, China, 10-12 August 2010.

[23] M. G. C. Resende, and P. Pardalos, Handbook of Optimization in Telecommunications. United States of America: Springer, 2006.