MOBILE ROBOTS NAVIGATION, MAPPING & LOCALIZATION: PART I

Lee Gim Hee*
CML
DSO National Laboratories
20 Science Park Drive
Singapore 118230
Tel: (+65) 6796 8427
Email: lgimhee@dso.org.sg

Marcelo H. Ang Jr.
Department of Mechanical Engineering
National University of Singapore
9 Engineering Drive 1
Singapore 117576
Tel: (+65) 6516 2555
Fax: (+65) 6779 1459
Email: mpeangh@nus.edu.sg

INTRODUCTION

The development of autonomous mobile robots is continuously gaining importance particularly in the military for surveillance as well as in industry for inspection and material handling tasks. Another emerging market with enormous potential is mobile robots for entertainment.

A fundamental requirement for autonomous mobile robots in most of its applications is the ability to navigate from a point of origin to a given goal. The mobile robot must be able to generate a collision-free path that connects the point of origin and the given goal. Some of the key algorithms for mobile robot navigation will be discussed in this article.

BACKGROUND

Many algorithms were developed over the years for the autonomous navigation of mobile robots. These algorithms are generally classified into three different categories: *global path planners*, *local navigation methods* and *hybrid methods*, depending on the type of environment that the mobile robot operates within and the robot's knowledge of the environment.

In this article, some of the key algorithms for navigation of a mobile robot are reviewed. Advantages and disadvantages of these algorithms shall be discussed. The algorithms that are reviewed include the *navigation function*, *roadmaps*, *vector field histogram*, *artificial potential field*, *hybrid navigation* and the *integrated algorithm*. Note that all the navigation algorithms that are discussed in this article assume that the robot is operating in a planar environment.

GLOBAL PATH PLANNERS

Global path planning algorithms refer to a group of navigation algorithms that plans an optimal path from a point of origin to a given goal in a known environment. This group of algorithms requires the environment to be free from dynamic and unforeseen obstacles. In this section, two key global path planning algorithms: *navigation functions* and *roadmaps* will be discussed.

Navigation Functions

The most widely used global path planning algorithm is perhaps the navigation function computed from the "wave-front expansion" (J.-C Latombe, 1991; Howie Choset et al, 2005) algorithm due to its practicality, ease in implementation and robustness. The navigation function N is the *Manhattan distance* to the goal from the free space in the environment. The algorithm requires information of the environment provided to the robot to be represented as an array of grid cells.

The navigation function assigns a numeric value N to each cell with the goal cell having the lowest value, and the other unoccupied cells having progressively higher values such that a steepest descent from any cell provides the path to the goal. The value of the unoccupied cell increases with the distance from the goal. Each grid cell is either free or occupied space denoted by gC_{free} and $gC_{occupied}$. First, the value of N is set to '0' at the goal cell gC_{goal}. Next, the value of N is set to '1' for every 1-Neighbor (see Figure 1 for the definition of 1-Neighbors) of gC_{goal} which is in gC_{free}. It is assumed that the distance between two 1-Neighbors is normalized to 1. In general, the value of each gC_{free} cell is set to N+1 (e.g., '2') for every unprocessed gC_{free} 1-

Neighbor of the grid cell with value *N* (e.g., ‘1’). This is repeated until all the grid cells are processed.

2	3	4
1	(x, y)	5
8	7	6

Fig. 1. The shaded cells are the 1-Neighbor of the cell (x, y) and the number shows the priority of the neighboring cells.

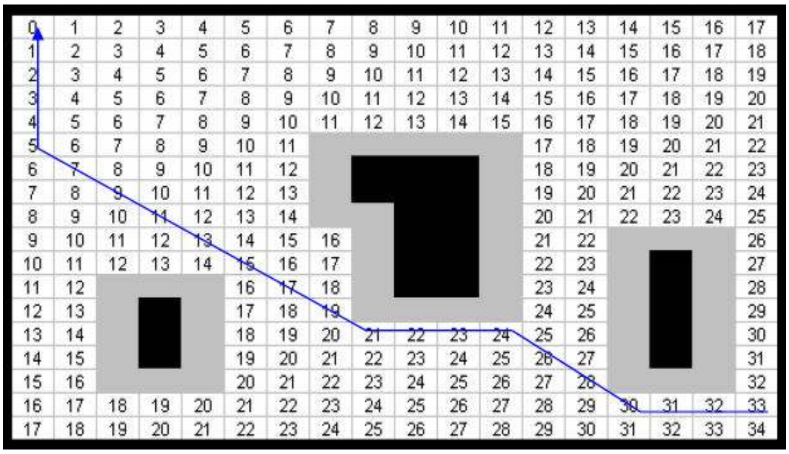

Fig. 2. Path generated by the navigation function.

Finally, a path to the goal is generated by following the steepest descent of the *N* values. To prevent the path from grazing the obstacles, the grid cells which are less than a safety distance α from the obstacles are omitted in the computation of the navigation function. Figure 2 shows a path generated by the navigation function. The black cells are the obstacles and the grey cells are the unsafe regions.

Roadmaps

A *roadmap* is a network of one-dimensional curves that captures the connectivity of free space in the environment (J.-C Latombe, 1991; Danner et al, 2000; Foskey et al, 2001; Isto P., 2002; T. Siméon et al, 2004; Xiaobing Zou et al, 2004; Howie Choset et al, 2005; Bhattacharya et al, 2007). Once a roadmap has been constructed, it is used as a set of standardized paths. Path planning is thus reduced to connecting the initial and goal positions to points in the roadmap. Various methods based on this general idea have been proposed. They include the *visibility graph* (Danner et al, 2000; Isto P., 2002; T. Siméon et al, 2004), *Voronoi diagram* (Foskey et al, 2001; Xiaobing Zou et al, 2004; Bhattacharya et al, 2007), *freeway net* and *silhouette* (J.-C Latombe, 1991; Howie Choset et al, 2005).

The *visibility graph* is the simplest form of *roadmap*. This algorithm assumes that the environment is made up of only polygonal obstacles. The nodes of a *visibility graph* include the point of origin, goal and all the vertices of the obstacles in the environment. The graph edges are straight line segments that connect any two nodes within the line-of-sight of each other. Finally, the shortest path from the start to goal can be obtained from the *visibility graph*.

Advantages and Disadvantages

The advantage of the *navigation functions*, *roadmaps* and other global path planning algorithms is that a continuous collision-free path can always be found by analyzing the connectivity of the free space. However, these algorithms require the environment to be known and static. Any changes in the environment could invalidate the generated path. Hence, the navigation functions and other global path planning algorithms are usually not suitable for navigation in an initially unknown environment and those with dynamic and unforeseen obstacles.

LOCAL NAVIGATION METHODS

In contrast to the global path planners, local navigation methods do not require a known map of the environment to be provided to the robot. Instead, local navigation methods rely on current and local information from sensors to give a mobile robot online navigation capability. In this section, two of the key algorithms for local navigation: *artificial potential field* and *vector field histogram* will be evaluated.

Artificial Potential Field

The *artificial potential field* (O.Khatib, 1986) method, first introduced by Khatib, is perhaps the best known algorithm for local navigation of mobile robots due to its simplicity and effectiveness. The robot is represented as a particle in the configuration space q moving under the influence of an artificial potential produced by the goal configuration q_{goal} and the scalar distance to the obstacles. Typically the goal generates an attractive potential such as

$$U_g(q) = \frac{1}{2} K_g \left(q - q_g\right)^T \left(q - q_g\right) \tag{1}$$

which pulls the robot towards the goal, and each obstacle i produces repulsive potential such as

$$U_{i,o} = \begin{cases} \frac{1}{2} K_o \left(\frac{1}{d_i} - \frac{1}{d_o} \right)^2 & \text{if } d_i < d_0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

which pushes the robot away from the obstacle. In cases where there is more than one obstacle, the total repulsive force is computed by the sum of all the repulsive forces produced by the obstacles. K_g and K_o are the respective gains of the attractive and repulsive potential. d_i is the scalar distance between the robot and obstacle i. The repulsive potential will only have effect on the robot when its moves to a distance which is lesser than d_0. This implies that d_0 is the minimum safe distance from the obstacle that the robot tries to maintain.

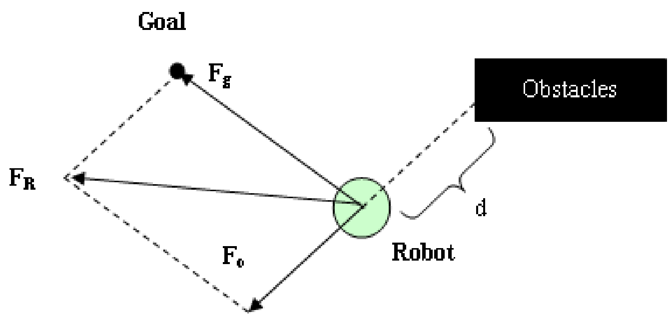

Fig. 3. Robot's motion influenced by potential field.

The negated gradient of the potential field gives the artificial force acting on the robot.

$$F(q) = -\nabla U(q) \tag{3}$$

Figure 3 shows the attractive force

$$F_g(q) = -K_g(q - q_g) \tag{4}$$

that is generated from the goal and the repulsive force

$$F_{i,o}(q) = \begin{cases} K_o\left(\frac{1}{d_i} - \frac{1}{d_o}\right)\frac{1}{d_i^2} & if\ d_i < d_o \\ 0 & otherwise \end{cases} \tag{5}$$

that is generated from an obstacle i. F_R is the resultant of all the repulsive forces and attractive force. Note that n denotes the total number of obstacles which is lesser than a distance d_o from the robot. At every position, the direction of this force is considered as the most promising direction of motion for the robot.

$$F_R(q) = F_g(q) + \sum_{i=1}^{n} F_{i,o}(q) \tag{6}$$

Vector Field Histogram

The *vector field histogram* method (Y.Koren et al, 1991; J.Borenstien, 1991; Zhang Huiliang et al, 2003) requires the environment to be represented as a tessellation of grid cells. Each grid cell holds a numerical value that ranges from 0 – 15. This value represents whether the environment represented by the grid cell is occupied or not. 0 indicates absolute certainty that the cell is not occupied and 15 indicates absolute certainty that the cell is occupied. A two stage data reduction process is carried out recursively to compute the desired motion of the robot at every instance of time.

In the first stage, the values of every grid cells that are in the vicinity of the robot's momentary location are reduced to a one-dimensional *polar histogram*. Each bin from the *polar histogram* corresponds to a direction as seen from the current location of the robot and it contains a value that represents the total sum of the grid cell values along that direction. The values from the *polar histogram* are also known as the *polar obstacle density* and they represent the presence of obstacles in the respective directions.

In the second stage, the robot selects the bin with a low *polar obstacle density* and direction closest to the goal. The robot moves in the direction represented by the chosen bin because this direction is free from obstacles and it will bring the robot closer to the goal.

Advantages and Disadvantages

The advantage of the *artificial potential field*, *vector field histogram* and other local navigation methods is that they do not include an initial processing step aimed at capturing the connectivity of the free space in a concise representation. Hence a prior knowledge of the environment is not needed. At any instant in time, the path is determined based on the immediate surrounding of the robot. This allows the robot to be able to avoid any dynamic obstacles in the robot's vicinity.

The major drawback of the local navigation methods is that they are basically steepest descent optimization methods. This renders the mobile robot to be susceptible to *local minima* (Y.Koren et al, 1991; J.-O. Kim et al, 1992; Liu Chengqing et al, 2000; Liu Chengqing, 2002; Min Gyu Park et al, 2004). A *local minimum* in the *potential field* method occurs when the attractive and the repulsive forces cancel out each other. The robot will be immobilized when it falls into a *local minimum*, and loses the capability to reach its goal. Many methods have been proposed to solve the *local minima* problem (J.-O. Kim et al, 1992; Liu Chengqing et al, 2000; Liu Chengqing, 2002; Min Gyu Park et al, 2004). For example, Liu Chengqing (Liu Chengqing et al, 2000; Liu Chengqing, 2002) has proposed the virtual obstacle method where the robot detects the *local minima* and fills the area with artificial obstacles. Consequently, the method closes all concave obstacles and thus avoiding *local minima* failures. Another method was proposed by Jin-Oh Kim (J.-O. Kim et al, 1992) to solve the *local minima* problem. This method uses *local minima free* harmonic functions based on fluid dynamics to build the artificial potentials for obstacle avoidances.

HYBRID METHODS

Another group of algorithms suggest a combination of the local navigation and global path planning methods. These algorithms aim to combine the advantages from both the local and global methods, and to also eliminate some of their weaknesses. In this section, two key hybrid methods algorithms: *hybrid navigation* and *integrated algorithm* will be reviewed.

Hybrid Navigation

Figure 4 shows an illustration of the *hybrid navigation algorithm* (Lim Chee Wang, 2002; Lim Chee Wang et al, 2002). This algorithm combines the *navigation function* with the *potential field* method. It aims to eliminate local minima failures and at the same time does online collision avoidance with dynamic obstacles.

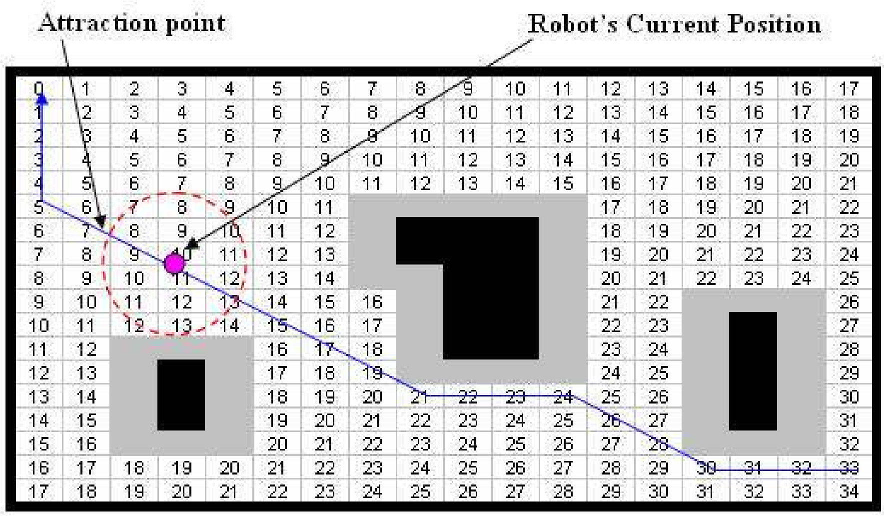

Fig. 4. Illustration of the hybrid navigation algorithm.

The robot first computes the path joining its current position to the goal using the *navigation function*. The robot then places a circle with an empirical radius centered at its current position. The cell that corresponds to the intersection of the circle with the *navigation function* path is known as the attraction point. The attraction point is the cell with the lowest *N* value if there is more than one intersection.

The robot advances towards the attraction point using the *potential field* method and the circle moves along with the robot which will cause the attraction point to change.

As a result, the robot is always chasing after a dynamic attraction point which will progress towards the goal along the *local minima* free *navigation function* path. The radius of the circle is made larger to intersect the *navigation function* path in cases where no intersections are found. The radius of the circle is reduced to smaller than the distance between the robot and its goal when the distance between the robot and its goal becomes smaller than the radius of the circle. This is to make sure that the N value of the next intersection will be smaller than the current N value.

Integrated Algorithm

In recent years, the *integrated algorithm* (Lee Gim Hee, 2005; Lee Gim Hee et al, 2007) has been proposed to give a mobile robot the ability to plan *local minima* free paths and does online collision avoidance to dynamic obstacles in an unknown environment. The algorithm modifies the *frontier-based exploration* method (Brian Yamauchi, 1997), which was originally used for map building, into a path planning algorithm in an unknown environment. The modified *frontier-based exploration* method is then combined with the *hybrid navigation* algorithm into a single framework.

Figure 5 shows an overview of the *integrated algorithm*. The robot first builds a local map (see Part II of this article for details on map building) of its surrounding. It then decides whether the goal is reachable based on the acquired local map. The goal is reachable if it is in free space, and is not reachable if it is in the unknown space. Note that an unknown region is a part of the map which has not been explored during the map building process. The robot will advance towards the goal using the *hybrid navigation* algorithm if it is reachable or advance towards the sub-goal and build another local map at the sub-goal if the goal is not reachable. This map will be added to the previous local maps to form a larger map of the environment. The process goes on until the robot finds the goal within a free space.

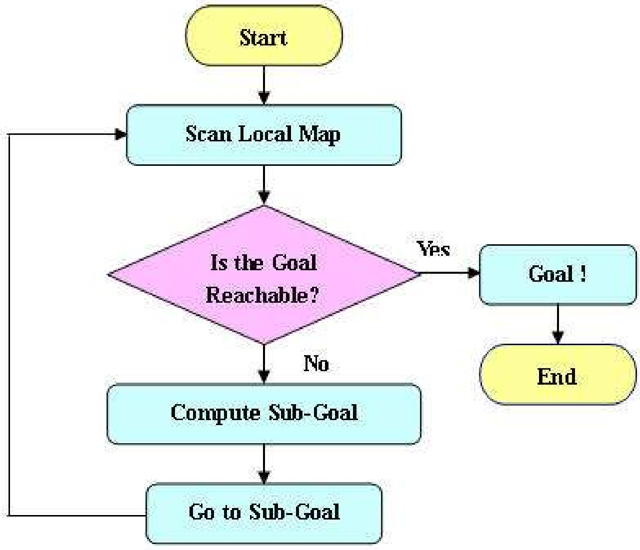

Fig. 5. The integrated algorithm.

The sub-goal is computed in three steps. First, compute the path that joins the robot's current position and the goal using the *navigation function*. The unknown cells are taken to be free space in the computation of the *navigation function*. Second, all the frontiers in the map are computed. The boundary of free space and unknown region is known as the frontier. A frontier is made up of a group of adjacent frontier cells. The frontier cell is defined as any gC_{free} cell on the map with at least two unknown cells $gC_{unknown}$ as its immediate neighbor. The total number of frontier cells that make up a frontier must be larger than the size of the robot to make that frontier valid. Third, the frontier that intersects the navigation function path will be selected and its centroid chosen as the sub-goal.

Advantages and Disadvantages

The *hybrid navigation* algorithm has the advantage of eliminating local minima failures and at the same time doing online collision avoidance with dynamic obstacles. However, it requires the environment to be fully known for the search of a navigation function path to the goal. The algorithm will fail in a fully unknown environment. It also does not possess any capability to re-plan the *navigation function* path during an operation. Therefore any major changes to the environment could cause failure in the algorithm.

The integrated algorithm has the advantages of planning local minima free paths, does online collision avoidance to dynamic obstacles in totally unknown environments. In addition, the algorithm gives the mobile robot a higher level of autonomy since it does not depend on humans to provide a map of the environment. However, the advantages come with the trade off of tedious implementations. This is because the integrated algorithm requires both the *hybrid navigation* algorithm and a good mapping algorithm to be implemented at the same time.

CONCLUSION

Mobile robot navigation involves more than planning a path from a point of origin to a given goal. A mobile robot must be able to follow the planned path closely and avoid any dynamic or unforeseen obstacles during its journey to the goal. Some of the key algorithms that give a mobile robot navigation capability were discussed in this article. These algorithms include the *navigation function*, *roadmaps*, *artificial potential field*, *vector field histogram, hybrid navigation* and the *integrated algorithm*.

FUTURE TRENDS

The assumption that on-board sensors have perfect sensing capability is generally made by researchers researching on mobile robot navigation. In reality, these sensors are corrupted with noise and this usually causes adverse effects on the performance of the navigation algorithms. The greatest challenge for a robust implementation of the navigation algorithms is therefore to minimize the adverse effects caused by the sensor uncertainty.

REFERENCE

Bhattacharya, Priyadarshi, Gavrilova, & Marina L. (2007). Voronoi diagram in optimal path planning. 4th International Symposium on Voronoi Diagrams in Science and Engineering. pp. 38 - 47.

Brian Yamauchi. (1997). A frontier-based approach for autonomous exploration. Proceedings of the IEEE International Symposium on the Computational Intelligence in Robotics and Automation. pp. 146-151.

Danner, T. Kavraki, & L.E. (2000). Randomized planning for short inspection paths. IEEE International Conference on Robotics and Automation**.** pp**.** 971-976.

Foskey, M. Garber, M. Lin, & M.C. Manocha, D. (2001). A Voronoi-based hybrid motion planner. Proceedings of International Conference on Intelligent Robots and Systems. pp. 55-60.

Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Bugrad, Lydia E. Kavraki, & Sebastian Thrun. (2005). Principles of robot motion: Theory, algorithms, and implementations. MIT Press.

Isto P. (2002). Constructing probabilistic roadmaps with powerful local planning and path optimization. IEEE/RSJ International Conference on Intelligent Robots and System. Vol. 3, pp. 2323 - 2328.

J.Borenstien, & Y.Koren. (1991). The vector field histogram – Fast obstacle avoidance for mobile robots. IEEE Journal of Robotics and Automation. Vol 7, No 3, pp. 278-288.

J.-C Latombe. (1991). Robot motion planning. Kluwer Academic Publishers.

J.-O. Kim, & P. K. Khosla. (1992). Real-time obstacle avoidance using harmonic potential functions. Proceedings of IEEE Transactions on Robotics and Automation. Vol.8, pp. 338—349.

Lee Gim Hee, “Navigation of a mobile robot in an unknown environment “, Thesis for Bachelor of Engineering, National University of Singapore 2005.

Lee Gim Hee, Lim Chee Wang, & Marcelo H. Ang Jr. (2007). An integrated algorithm for autonomous navigation of a mobile robot in an unknown environment. Third Humanoid, Nanotechnology, Information Technology, Communication and Control Environment and Management (HNICEM) International Conference.

Lim Chee Wang. (2002). Motion planning for mobile robots. Thesis for Master of Engineering, National University of Singapore.

Lim Chee Wang, Lim Ser Yong, & Marcelo H. Ang Jr. (2002). Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment. Proceedings of the IEEE International Symposium on Intelligent Control. pp. 821-826.

Liu Chengqing. (2002). Sensor based local path planning for mobile robots. Master's Thesis, National University of Singapore.

Liu Chengqing, Marcelo H. Ang Jr, H. Krishnan, & Lim Ser Yong (2000). Virtual obstacle concept for local minima recovery in potential field based navigation. Proceedings of the IEEE Conference on Robotics and Automation. Vol. 2, pp. 983-988.

Min Gyu Park , & Min Cheol Lee. (2004). Real-time path planning in unknown environment and a virtual hill concept to escape local minima. 30th Annual Conference of IEEE Industrial Electronics Society. Vol. 3, pp. 2223 - 2228.

O.Khatib. (1986). Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotic Research. Vol. 5, No. 1, pp.90-98.

T. Siméon, J.-P. Laumond, & C. Nissoux. (2004). Visibility-based probabilistic roadmaps for motion planning. Journal of Advanced Robotics, Brill Academic Publishers. pp. 477-493.

Xiaobing Zou, & Zixing Cai. (2004). Incremental environment modeling method based on approximate Voronoi diagram. Fifth World Congress on Intelligent Control and Automation. Vol.5, pp. 4618 - 4622.

Y.Koren, & J.Borenstien. (1991) Potential field methods and their inherent limitations for mobile robot navigation. Proceedings of the IEEE Conference on Robotics and Automation. pp.1398-1404.

Zhang Huiliang, & Huang Shell Ying. (2003). Dynamic map for obstacle avoidance. Proceedings of IEEE Intelligent Transportation Systems. Vol.2, pp. 1152 - 1157.

TERMS AND DEFINITIONS

Global path planner: A group of navigation algorithms for planning an optimal path that connects a point of origin to a given goal in a known environment.

Manhattan distance: The distance between two points measured along axes at right angles. For example, given two points p_1 and p_2 in a two-dimensional plane at (x_1, y_1) and (x_2, y_2) respectively, the Manhattan distance between p_1 and p_2 is given by $|x_1 - x_2| + |y_1 - y_2|$.

Graph Node: Graph Node is also known as graph vertex. It is a point on which the graph is defined and maybe connected by graph edges.

Graph Edge: Graph edge is usually drawn as a straight line in a graph to connect the nodes. It is used to represent connectivity between two or more nodes and may carry additional information such as the Euclidean distance between the nodes.

Local navigation methods: A group of navigation algorithms that do not require a known map of the environment to be provided to the robot. Instead, local navigation methods rely on current and local information from sensors to give a mobile robot online navigation capability.

Hybrid methods: A group of navigation methods that combine the global path planning and local navigation algorithms. The objective is to combine the advantages eliminate the inherent weaknesses of both groups of algorithms.

Local minima: It is also known as relative minima. Local minimum refers to a minimum within some neighborhood and it may not be a global minimum.