An Improved PBIL Algorithm for Path Planning Problem of Mobile Robots

Qingbin Zhang [1,2], Manjun Cai[3], Fajun Zhou[1], and Hairong Nie[1]

[1] Center of Green & Intelligent Engineering,
Shijiazhuang Institute of Railway Technology, Shijiazhuang 050061, China
[2] Key Laboratory of Advanced Design and Intelligent Computing (Dalian University),
Ministry of Education, Dalian 116622, China
[3] Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
zqbin2002@sina.com

Abstract. The path planning problem of mobile robots is a NP-Hard problem often solved by evolutionary approaches such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO). However, the algorithm's performance is often influenced heavily by the determination of the operators and the choice of related parameters. In this paper, a permutation code PBIL is proposed to solve the path planning problem. First, a free space model of the mobile robot is constructed by the MAKLINK graph; second, a sub-optimal path is generated by the Dijkstra algorithm; then global optimal path is constructed by the permutation code PBIL based on the sub-optimal path. Simulation results show that the PBIL can get satisfied solutions more simply and efficiently with fewer operators and parameters.

Keywords: Path planning problem, MAKLINK graph, Permutation code PBIL.

1 Introduction

The path planning problem of autonomous mobile robots is to find a collision-free path from a starting point to a goal in a given environment according to distance, time or cost [1]. Research results show that the classic approaches, such as the potential field method [2], visibility graph methods [3], and grid methods [4] tend to get local optimal solution which may be far inferior to the global optimal solution, so it is more appropriate to adopt heuristic based evolutionary approaches to solve the problems [5]. Among those approaches, Genetic Algorithm (GA) [6] and Ant Colony Optimization (ACO) [7] have been widely employed to find the globally optimal path.

Sun Shudong and Lin Mao [8] presented a strategy for path planning of multi-mobile robots using GA. MAKLINK graph was used to model the workspace of robots. Sequence numbers was used to mark the cross points on the MAKLINK graph and also as an encoding method for the chromosomes of GA. Simulation results show that the method is effective in path planning of multi-mobile robots. Nagib and Gharieb[9] constructed fixed-length path using GA consisting of binary strings in a grid model. The proposed algorithm could find a quick solution for environments with few

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 85–92, 2013.

obstacles. Al-Taharwa et al.[10] adopted GA with an integer string encoding method to solve the mobile robot path planning problem in a grid environment model. Experiments show that the proposed approach is effective and efficient in handling different types of tasks in static environments.

Tan Guanzheng [11] proposed the ACS to find the globally optimal path of mobile robots based on MAKLINK graph model and the initial sub-optimal collision-free path constructed by Dijkstra algorithm. Results show that ACS performed better than that of GA. Jing Zhou et al. [12] proposed two algorithms based on ACO framework, they applied visibility graph as the roadmap and construction graph. Experimental results illustrated that the proposed algorithms are effective and a flexible balance between the optimum and the number of iterations could also be achieved.

However, the research has also shown that the performance of GA and ACS for the path planning problems of mobile robots depends heavily on determination of the operators as well as choice of related parameters, which needs prior knowledge about the problem at hand, otherwise extensive experiments have to been done to choose an effective configuration of the operators and the parameters [13].

The PBIL is one of the simple optimization algorithm belongs to the Estimation of Distribution Algorithms (EDAs) paradigm [14, 15]. PBIL applies a probability vector to represent the population, with each element referring to the probability of obtaining a given value in a specific gene position. The probability vector is updated towards the selected solutions by Hebbian rule. PBIL has been proved to be more successful on many benchmark functions and has been used in some real-world problems [16].

The initial PBIL uses a binary code string to represent the variables. In this paper, a permutation code PBIL to solve the path planning problem is proposed. Experimental results compared with that of the ACS are reported.

2 The Path Planning Problem

In order to solve the global optimal path efficiently, the path planning problem is often divided into two subtasks: 1) establish the environment model of the mobile robot. Various types of known static environment models have been proposed by previous robot path planning researchers [13], such as Visibility graph[3], MAKLINK graph[11], and Cell Decomposition[10] are used to produce the connectivity of graph for robot path planning. 2) Searching the graph for a sequence of vertices which connect the start and goal position. After the graph model has been constructed, the path planning is thereby converted into a problem searching for an optimal path between the initial and the final of the points in the given graph. Usually, the Dijkstra algorithm or the A* algorithm is used [17].

2.1 Environment Model of the Mobile Robot

MAKLINK graph and grid model are two mainly tools to model the space of mobile robots. In this section, a free space model of the mobile robot is constructed by the MAKLINK graph theory. The free space in an environment means the space in which the robot can move freely. To establish the free space model, the following assumptions need to be made: 1) the heights of the environment and obstacles can be ignored; 2) both the environment and the obstacles have a polygonal shape; 3) in order to avoid a moving path is too close to the obstacles, the boundaries of every obstacle can be expanded and then the size of the robot can be ignored.

Assume the total number of the free MAKLINK lines on a MAKLINK graph is l, the middle points of these free MAKLINK lines can be denoted by v_1, v_2,…, v_l, respectively. If we denote the starting point S and goal T as v_0 and v_{l+1}. Then points set V={ v_0, v_1,…, v_{l+1} } is the set including points S and T and the middle points of all the free MAKLINK lines. Further more, set E is defined as a set of the lines which includes: the lines connecting point S and the middle points on the free MAKLINK lines adjacent to S; the lines connecting a pair of the middle points on two adjacent free MAKLINK lines; and the lines connecting the goal T and the middle points on the free MAKLINK lines adjacent to T. thus the undirected graph G (V, E) can be respected as the free space model, the path planning can be converted into finding the shortest path between the given starting point S and goal T on G (V, E).

2.2 Searching for Sub-optimal Path Using Dijkstra Algorithm

In order to find the shortest path between the starting point S and goal T by Dijkstra algorithm, it is necessary to define the adjacency matrix with weights for the network graph G (V, E) in which each element of the matrix represents the length of the straight line between two adjacent path nodes on G (V, E), where a path node means the intersection of the robot path and a free MAKLINK line. Each element of the matrix is defined as follows:

$$adjlist[i][j] = \begin{cases} length(v_i, v_j), & if\ edge(v_i, v_j) \in E \\ \infty, & others \end{cases} \tag{1}$$

Where $adjlist[i][j]$ is the element corresponding to the i^{th} row and the j^{th} column of the matrix, $length(v_i, v_j)$ is the straight-line distance between the path nodes v_i, v_j, i and j=0, 1, 2,…, l, l +1.

The shortest path generated by the Dijkstra algorithm is just a sub-optimal path because all the path nodes are middle points of the relevant free MAKLINK lines, but it can be adopted as the initial path for evolutionary approaches to search the globally optimal path.

2.3 Description of the Optimization Problem

Assume that the sub-optimal path is denoted in order by path nodes P_0, P_1,, …, P_d, and P_{d+1}, where P_0 and P_{d+1}denote the starting point S and the goal T, respectively. We can adjust the position of each path node in the free MAKLINK lines to construct new paths and then optimize the location of the path nodes so as to generate the globally optimal path. For each point in a free MAKLINK line, its location can be expressed as:

$$P_i = P_{i1} + (P_{i2} - P_{i1}) \times h_i,\ \ h_i \in [0,1] \tag{2}$$

So different combinations of parameters h_1, h_2,…, h_d will generate different paths. Thus finding the shortest path on MAKLINK graph can be regarded as another optimization problem, which is to find the optimal parameter set $\{ h_1, h_2,\ldots,h_d \}$ such that the following objective function has the minimum value.

$$L = \sum_{i=0}^{d} length\{P_i(h_i), P_{i+1}(h_{i+1})\} \tag{3}$$

Where length (P_i, P_{i+1}) represents the straight line distance between two adjacent path nodes P_i, and P_{i+1} on the lines L_i and L_{i+1}.

In order to solve the new optimization problem, we divide equally each of the MAKLINK lines into D portions, and thus (D+1) nodes are generated on each line, which represent the (D+1) possible values of parameter h_i (i=0,1, 2,…, d). Then the solution space is $d\times(D+1)$ nodes in total. For example, if we divide equally each of the MAKLINK lines into ten portions, and thus eleven nodes on each line represented as h_i are 0, 0.1, 0.2…, and1.0.

3 PBIL for Optimal Path Planning Problem

3.1 PBIL Algorithm

The initial PBIL uses a binary code in which the population is represented by a probability vector:

$$p_l(x) = (p_l(x_1), p_l(x_2), \ldots\ldots p_l(x_n))$$

Where p_l (x_i) refers to the probability of obtain a value 1 in the i^{th} gene position in the l^{th} generation. The algorithm begins by initializing the probability vector $p_0(x)$ with all elements set to 0.5, which means random solutions are created when sampling from this probability vector. As search progresses, the values in the probability vector are gradually learned towards values representing high fitness solutions.

The evolution process in PBIL is accomplished as follows: At each generation, M solutions are generated based upon the probabilities specified in the current probability vector $p_l(x)$. The solutions are then evaluated and N best solutions ($N\leq M$) are selected as the best solutions set. We denote them by $x_{k:M}^l$ (k=1,…, N). These selected best solutions are used to update the probability vector by using a Hebbian inspired rule:

$$p_{l+1}(x) = (1-a)p_l(x) + a\frac{1}{N}\sum_{k=1}^{N} x_{k:M}^l \tag{4}$$

Where a is the learning rate.

After the probability vector is updated, a new set of solutions are generated by sampling from the new probability vector $p_{l+1}(x)$, and the cycle is repeated until some termination condition is satisfied, e.g., the probability vector is converged to either 0.0 or 1.0 for each bit position, or the number of iterations is met.

3.2 Permutation Code PBIL

In the permutation code PBIL, the probability vector represented the direction of the evolution in binary PBIL is expanded to be a probability matrix P. Suppose the solution of a permutation optimization problem is represented by an integer string, the length of the integer string is n and the value range is from 1 to m, then an element p_{ij} in P represents the probability which the value is j in the i^{th} gene ($1 \leq i \leq n$; $1 \leq j \leq m$).

The initial value of the probability matrix is a uniform distribution. During the evolution, the values of the p_{ij} is updated by the formula (4) and roulette method is used to sample from the probability matrix in order to get the solutions of the problem.

3.3 PBIL for Optimal Path Planning Problem

The PBIL for path planning problem of a mobile robot between a given starting point S and a given goal T in a given environment can be described as follows.

Step1. Construct the free space model of the robot using the MAKLINK graph, and then find a sub-optimal collision-free path in the free space using the Dijkstra algorithm;

Step2. Define the initial probability matrix P with each element p_{ij} as $1/(D+1)$, Sample from the probability matrix P to form the initial solutions set S_1 with M paths;

Step3. Set the iteration counter t=1, Repeat for t=1,2,...until the number of iterations NC is met;

Step4.Select $N(N<M)$ individuals from S_t according to the length of different paths, the path with the minimum length is represented as $B(t)$;

Step5.Update the probability matrix P with formula (4);

Step6. Sample from the probability matrix P with roulette wheel selection method to generate M new paths to form the new set S_{t+1};

Step7. Set $t=t+1$, if $t < NC$ return to Step 4. Else, output the terminal probability matrix P, and the best solution set B.

4 Experiments and Results

4.1 The Example Problem

The example moving environment of a mobile robot is a 200 by 200 meters square and includes four obstacles. The coordinates of vertices of the four obstacles are(40 140;60 160;100 140; 60 120),(50 30;30 40; 80 80; 100 40), (120 160;140 100;180 170;165 180),(120 40;170 40;140 80),respectively. The(x,y)coordinates of the starting point S and the goal T are(20,180) and (160,90), respectively[18].

For the example problem, the free space graph of the robot is shown in Fig.1, where number of the free MAKLINK lines l=20. The sub-optimal path is obtained by

Dijkstra algorithm is *S*—v8—v7—v6—v12—v13—v11—*T*, which is shown in Fig. 2.The length of this path is180.78 meters.

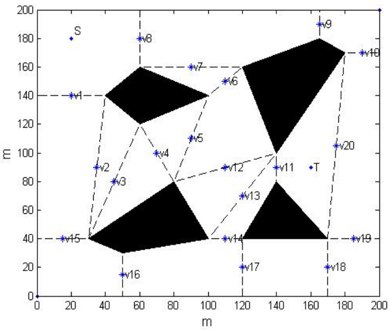

Fig. 1. The free space of the robot

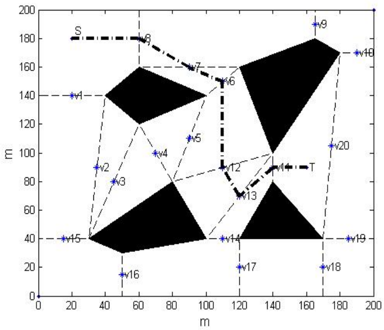

Fig. 2. The optimal path obtained by Dijkstra algorithm

4.2 Experimental Results

The solution of the PBIL is a probability matrix P in which each element p_{ij} means in the i^{th} line L_i, the probability of selects the j^{th} point. i=(1,2,…d), j=(1,2,…D, D+1). However, we can construct the optimal path by sampling from the matrix.

To demonstrate the performance of the proposed PBIL on the globally optimal path planning problem, parameters are set as follows: the sample paths in each iteration M =200; in which number of selected as the best paths set N=0.2*M; the learning rate a is set to be 0.05, D=10. Experimental results of the average value of probability matrix P based on 100 separate running with 500 iterations are presented in Table1. From the Table 1, we can conclude that the algorithm will evolve towards some positions with high probability on each line of the path and the probability of other points will converge to zero. So we can construct different paths following different demands.

Table 1. Average value of probability matrix P

No. of the lines	Probability of each position on different lines									
1	0.12	0.86	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.21	0.67	0.11
3	0.96	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.55	0.45
5	0.54	0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.39	0.06	0.00	0.47	0.08	0.00	0.00	0.00	0.00	0.00	0.00

A typical running comparing of the PBIL and ACS for the path planning problem of mobile robots is shown in Fig.3. The path founded by PBIL and that of ACS is shown in Figure 4. the operators and parameters of ACS are set same as in [7].The length of the path founded by PBIL is 168.67 meters, however, the length of the path founded by ACS is173.82 meters, so compared with the ACS, PBIL can find shorter path with faster converge speed.

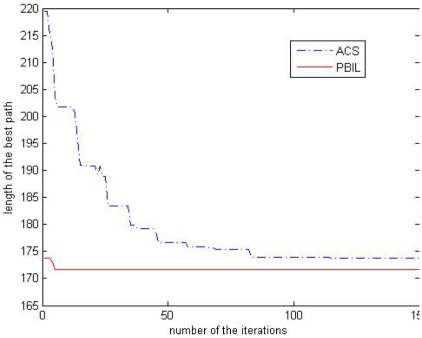

Fig. 3. Typical running of the PBIL and ACS

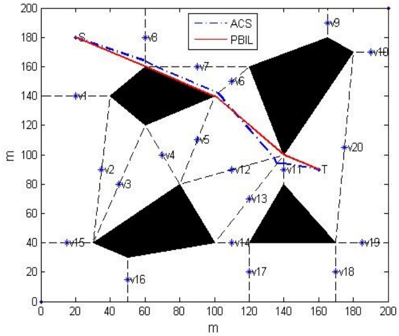

Fig. 4. The optimal path obtained by PBIL and ACS

5 Conclusion

In this paper, a permutation code PBIL is proposed to solve the path planning problem of autonomous mobile robots. The free space model of the mobile robot is constructed by the MAKLINK graph and the initial sub-optimal path is generated by the Dijkstra algorithm. The global optimal path is constructed by sampling from the probability matrix of the permutation code PBIL. Simulation results compared with that of the ACS on experimental problem show that proposed PBIL can solve the problem more simply and efficiently with less operators and parameters.

Acknowledgements . This work was supported by Key Laboratory of advanced Design and Intelligent Computing (Dalian University), Ministry of Education (ADIC2010005).

Refences

1. Raja, P., Pugazhenthi, S.: Optimal path planning of mobile robots: A review. International Journal of Physical Sciences 7(9), 1314–1320 (2012)
2. Arambula, F., Padilla, M.: Autonomous Robot Navigation Using Adaptive Potential Fields. Mathematical and Computer Modeling 40, 1141–1156 (2004)

3. Roh, S.G., Park, K.H., Yang, K.W.: Development of Dynamically Reconfigurable Personal Robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4023–4028 (2004)
4. Payton, D.W., Rosenblatt, J.K., Keirsey, D.: Grid-based mapping for autonomous mobile robot. Robot. Auton. Syst. 11(1), 13–21 (1993)
5. Hu, Y., Yang, S.X.: A Knowledge Based Genetic Algorithm for Path Planning of Mobile Robot. In: IEEE Int. Conf. on Robotics and Automation, vol. 5, pp. 4350–4355 (2004)
6. Goldberg, D.E.: Genetic Algorithms in Search,Optimization and Machine Learning. Addison-Wesley Longman publishing Co., Inc., Boston (1989)
7. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press/Bradford Books, Cambridge (2004)
8. Sun, S., Lin, M.: Path Planning of multi-mobile robots using genetic algorithms. Acta Automatica Sinica 5(26), 672–676 (2000)
9. Nagib, G., Gharieb, W.: Path planning for a mobile robot using genetic algorithms. In: Proceedings of the International Conference on Electrical, Electronic and Computer Engineering (ICEEC 2004), pp. 185–189 (2004)
10. AL-Taharwa, I., Sheta, A., Al-Weshah, M.: A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment. Journal of Computer Science 4(4), 341–344 (2008)
11. Tan, G., He, H.: Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots. Acta Automatica Sinica 33(3), 279–285 (2007)
12. Zhou, J., Dai, G., He, D.Q., Ma, J., Cai, X.-Y.: Swarm Intelligence: Ant-based Robot Path Planning. In: Fifth International Conference on Information Assurance and Security, pp. 459–463 (2009)
13. Buniyamin, N., Sariff, N., Wan Ngah, W.A.J., Mohamad, Z.: Robot global path planning overview and a variation of ant colony system algorithm. International Journal of Mathematics and Computers in Simulation 5(1), 9–16 (2011)
14. Baluja, S.: Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Carnegie Mellon University, Technical Report CMU-CS-94-163 (1994)
15. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2002)
16. Wang, Z., Zhang, Q., Ma, Y., Zhang, J., Liu, Y.: An Improved PBIL Algorithm for the Machine-Part Cell Formation. Applied Mechanics and Materials(26 - 28), 498–501 (2010)
17. Sariff, N., Buniyamin, N.: An overview of autonomous mobile robot path planning algorithms. In: 4th Student Conference on Research and Development, pp. 183–188 (2006)
18. Li, M.: Modeling and Path Planning of Mobile Robot. Yanshan University (2012)