An Autonomous Mobile Robot Using Genetic Algorithm for Finding the Shortest Track

Thiang, Hany Ferdinando, Ronald Kurniawan
Electrical Engineering Department, Faculty of Industrial Technology, Petra Christian University
e-mail: thiang@petra.ac.id, hanyf@petra.ac.id

Abstrak

Makalah ini menjelaskan robot mobil cerdas yang mampu mencari jalur terpendek dengan menggunakan algoritma genetika (GA). Mikrokontroler 89C51 digunakan untuk mengontrol robot secara keseluruhan termasuk mencari jalur terpendek menggunakan GA. Beberapa prosedur untuk proses GA dalam bahasa assembly telah dikembangkan seperti membangkitkan populasi awal, evaluasi, seleksi dan mutasi. Kromosom GA direpresentasikan dalam bentuk bit string. Sistem ini menggunakan kombinasi dari seleksi roulette wheel dan seleksi top. Robot mobil didisain sebagai line tracking robot. Sebuah program PC bantuan dikembangkan untuk proses visualisasi dan transmisi parameter ke mikrokontroler. Percobaan untuk mencari jalur terpendek telah dilakukan dan memberikan hasil yang baik. Proses GA dapat diimlementasikan dengan baik pada mikrokontroler dan robot dapat mencari jalur terpendek dari titik asal menuju ke titik tujuan.

Kata Kunci: *Robot mobil mandiri, algoritma genetika, mikrokontroler, optimasi*

Abstract

The paper describes an intelligent autonomous mobile robot that can find a shortest track by using a genetic algorithm (GA). An 89C51 microcontroller system has been implemented to perform overall robot control including shortest track searching by using GA. An assembly code has been written for GA processes such as initial population generation, evaluation, selection and mutation. The GA chromosome is represented by a bit string. Combination between roulette wheel selection and top selection scheme are used in the system. The Mobile robot is assigned as a line tracker robot. An auxiliary PC software has been developed for process visualization and parameter transmission to the microcontroller. The experiment of the shortest track searching has been done and showed a reasonable good result. The GA process was well implemented on the microcontroller and the robot could find the shortest track from the given origin location to the target.

Keywords: Autonomous mobile robot, genetic algorithm, microcontroller, optimization

Introduction

Generally, a genetic algorithm is implemented on computer by using a high level programming language like as Pascal, C++ etc. It is not easy to implement a genetic algorithm on microcontroller. There are limited instructions in microcontroller assembly language. Therefore, this research is focused on implementation of genetic algorithm on microcontroller and apply it to a mobile robot for finding the shortest track.

In this research, we used 89C51 microcontroller because this microcontroller is very familiar in Indonesia. All genetic algorithm processes like crossover, mutation, generating an initial population, selection, evaluating etc. are done by 89C51 microcontroler. This microcontroller will also control the robot to move from one location to another location according to the result of genetic algorithm searching.

Catatan : Diskusi untuk makalah ini diterima sebelum tanggal 1 Januari 2002. Diskusi yang layak muat akan dierbitkan pada Jurnal Teknik Elektro volume 2 nomor 1 Maret 2002

Genetic Algorithm

Genetic algorithm is a stochastic search algorithm based on mechanism of natural selections and natural genetics. Genetic algorithm is very useful for solving the complex optimal problems that are not easy to solve using conventional method.

A simple genetic algorithm is composed of three operators [3] i.e. reproduction operator, crossover operator and mutation operator. Genetic algorithm can be described as follow steps:

1. Generate Initial Population
   Initial population is generated randomly. This population represents initial solutions. The

population consists of the solutions represented by the chromosomes.

2. Generate New Population
   There are three operators for generating a new population i.e. reproduction, crossover and mutation. This process is done repeatedly until the system get enough chromosomes for new population. The new population represents the new solution.
3. Evaluation
   This process will evaluate each population and it is done by calculate the fitness value of each chromosomes. If stopping criteria is not achieved, the process will continue step 2.
   There are some stopping criteria. They are:
   - Stop after reach out of *n* generations.
   - Stop if the best fitness value in the last *n* generations is not change.
   - Stop if there is no better fitness value in next *n* generations.

Before we run genetic algorithm, we must define the chromosome format and determine the fitness function or objective function. Both are also important in genetic algorithm for solving the problems.

Hardware of Mobile Robot

The robot is designed as a line-tracking robot. Figure 1 shows the block diagram of robot hardware. The robot is controlled by 89C51 microcontroller. Genetic algorithm is also implemented on the same microcontroller. For detecting the line, we use infrared transmitter and receiver as the sensor. The robot will follow the defined line for moving from a location to another location. The color of the line is white and the track color is black. The motor driver is designed using IC L298 Dual Monolithic Bridge. Figure 2 and 3 show circuit of infrared sensor and motor driver.

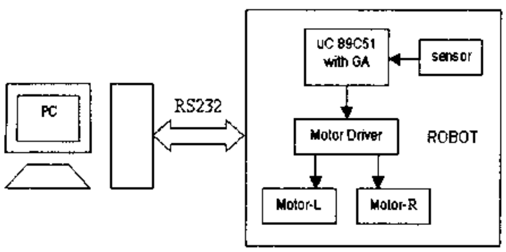

Figure 1. Block Diagram of Robot Hardware

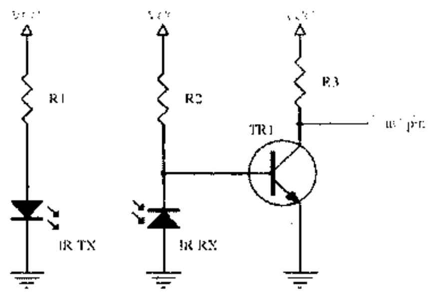

Figure 2. Infrared Sensor Circuit

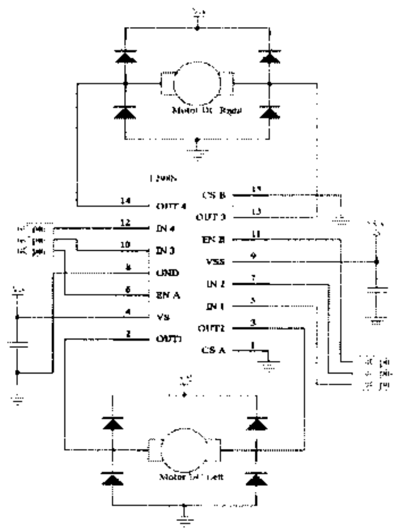

Figure 3. Motor Driver Circuit

Microcontroller is connected to PC through serial communication with RS232 protocol. PC is only used for making the map and downloading the information to microcontroller.

GA Routine Software

Some standard routines are designed for implementing genetic algorithm on microcontroller. They are:

- *Random_8* and *Random_16* routine
  This routine is designed for generating random number. Type of random number is *pseudo random*. Random_8 routine is used for generating 8 bits random number and random_16 routine is use for generating 16 bits random number.
- *Create_1st_Generation* routine
  This routine is used for generating initial population.

- *Crossover* routine
  This routine is used for doing the crossover process between two chromosomes. It results two new chromosomes. The type of crossover is one-cut point crossover.
- *Mutation* routine
  This routine is used for doing the mutation process. The type of mutation is bit mutation.
- *Roulette_ Wheel_Selection* routine
  This routine is used for doing the selection process using roulette wheel selection method.
- *Elitsm_Selection* routine
  This routine is used for doing the selection process using elitsm selection method. In this method, the best chromosome will be selected.

We must take not of the specification of this software. They are maximum five chromosomes per population, the type of chromosome is binary bit string and maximum length of chromosome is 16 bits.

Finding the Shortest Track Application

Figure 4 shows the example of the map. This map is designed by using the software run on PC. Points in the map show the locations. Each point has some branches. The branch will connect one point to another point. The distance between two points is already defined. In this system, the number of branch is limited to four branches per point. Each branch in one point has identification number. For example, in figure 4, point 1 has two branches. The identification number of the first branch is 0 and the second one is 1. Branch 0 of point 1 will connect point 1 to point 2. Branch 1 of point 1 will connect point 1 to point 6

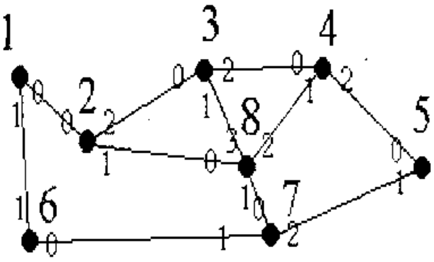

Figure 4. Example of the Map

1. Chromosome Representation

The GA chromosome was represented by a bit string. The number of bit per chromosome depends to the number of point and the number of branch per point.

The length of chromosome can be calculated by using this equation:

$$P = C \times (T-1) \qquad (1)$$

Where C is the number of binary bit of branch number per point. T is number of point. In the example of the map (see figure 4), there is 8 points and maximum branch number per point is 4 (it needs 2 binary bit). Thus, length of chromosome can be calculated and it is 14 bits.

Each chromosome gives routing information from one point to another point. Bits of chromosome represent the number of branches of the points. If number of branch is not found then it will do repeatedly. Let see the following example according to the map in the figure 4.

Start point = Point 1
Chromosome = 01 00 11 10 00 11 10
Route = 1 – 6 – 7 – 8 – 4 – 3 – 2 – 3

2. Fitness Function

In this system, we want to find the shortest track from one point to another point. Thus, fitness function is defined as total distance of the route. It is represent by this equation:

$$F = \sum_{Route} Distance\ Between\ Two\ Points \qquad (2)$$

3. Genetic Algorithm Operators

This application uses roulette wheel selection and elitsm as selection method (reproduction operator). In roulette wheel selection method, each chromosome has probability to be selected and its probability is proportional to its fitness value. While we want to find the shortest track, fitness value of each chromosome is changed by using this equation:

$$F_n' = F_{max} + F_{min} - F_n \qquad (3)$$

Where F_n' is new fitness value, F_{min} is the smallest fitness value and F_{max} is the biggest fitness value.In this application, type of crossover method is one-cut point crossover and type of mutation method is bit mutation.

4. Evaluation and Stopping Criteria

Evaluation process is done by calculating fitness value for each chromosome in one generation. A chromosome is not valid if there is no target information in that chromosome. If a chromosome is not valid then we assume the chromosome has biggest fitness value.

In this application, genetic algorithm will stop if there is no better fitness value in next 500 generations.

5. Experiment Result

The experiment has been done by run the program and showed a reasonable good result. The experiment was also done for various maps, various combination of selection method, various crossover rate and mutation rate.

Figure 5 shows one of the maps used for experiment. Table 1 shows experiment result using elitsm selection method for various crossover rate and mutation rate. Table 2 shows experiment result using roulette wheel selection method for various crossover rate and mutation rate. Table 3 shows experiment result using both elitsm and roulette wheel selection method for various crossover rate and mutation rate. In this experiment, we tried to find the shortest track from point 1 to point 16

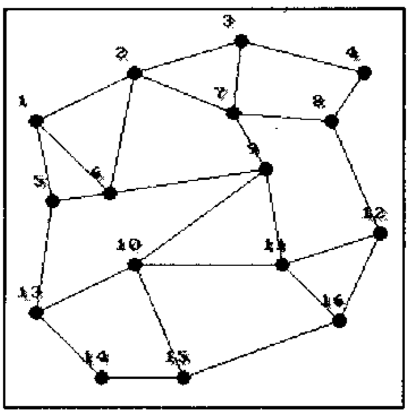

Figure 5. One of the Maps used for Experiment

Table 1, table 2 and table 3 show that the best crossover rate is varies from 0.7 to 0.9 and the best mutation rate is varies from 0.01 to 0.3. In the special case, smaller crossover rate like 0.4 and 0.1 can give good result too. It can occur if we use greater mutation rate like 0.5.

Figure 6 shows the best routing from point 1 to point 16.

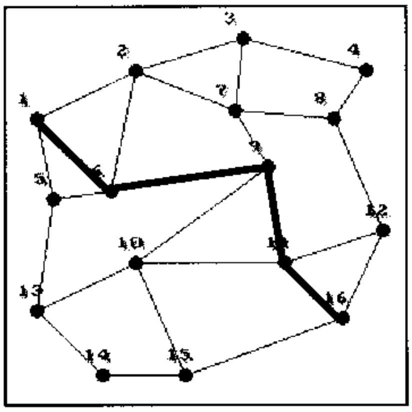

Figure 6. Best routing from point 1 to 16

Table 1. Experiment Result Using Elitsm

No.	Pc	Pm	Time (sec.)	Best Fitness	Best Routing	At Generation
1	0.9	0.01	23.47	254	1,6,9,11,16	146
2	0.9	0.1	21.92	254	1,6,9,11,16	110
3	0.9	0.3	18.44	254	1,6,9,11,16	9
4	0.9	0.5	18.81	254	1,6,9,11,16	12
5	0.7	0.01	23.41	254	1,6,9,11,16	146
6	0.7	0.1	20.3	254	1,6,9,11,16	48
7	0.7	0.3	17.95	254	1,6,9,11,16	9
8	0.7	0.5	20.17	254	1,6,9,11,16	54
9	0.4	0.01	17.97	254	1,6,9,11,16	12
10	0.4	0.1	21.81	254	1,6,9,11,16	110
11	0.4	0.3	19.45	254	1,6,9,11,16	48
12	0.4	0.5	17.89	254	1,6,9,11,16	9
13	0.1	0.01	19.67	254	1,6,9,11,16	48
14	0.1	0.1	18.92	254	1,6,9,11,16	38
15	0.1	0.3	19.85	254	1,6,9,11,16	43
16	0.1	0.5	18.39	254	1,6,9,11,16	9

Table 2. Experiment Result Using Roulette Wheel

No.	Pc	Pm	Time (sec.)	Best Fitness	Best Routing	At Generation
1	0.9	0.01	30.12	254	1,6,9,11,16	148
2	0.9	0.1	23.59	254	1,6,9,11,16	2
3	0.9	0.3	23.36	254	1,6,9,11,16	6
4	0.9	0.5	23.74	254	1,6,9,11,16	2
5	0.7	0.01	24.48	254	1,6,9,11,16	15
6	0.7	0.1	30.6	254	1,6,9,11,16	148
7	0.7	0.3	27.86	254	1,6,9,11,16	91
8	0.7	0.5	14.42	254	1,6,9,11,16	15
9	0.4	0.01	30.1	254	1,6,9,11,16	148
10	0.4	0.1	27.85	254	1,6,9,11,16	91
11	0.4	0.3	27.66	254	1,6,9,11,16	99
12	0.4	0.5	24.18	254	1,6,9,11,16	22
13	0.1	0.01	23.97	254	1,6,9,11,16	15
14	0.1	0.1	29.94	254	1,6,9,11,16	148
15	0.1	0.3	24.97	254	1,6,9,11,16	34
16	0.1	0.5	24.63	254	1,6,9,11,16	22

Table 3. Experiment Result Using Elitsm and Roulette Wheel

No.	Pc	Pm	Time (sec.)	Best Fitness	Best Routing	At Generation
1	0.9	0.01	25.7	254	1,6,9,11,16	58
2	0.9	0.1	26.7	254	1,6,9,11,16	80
3	0.9	0.3	24.33	254	1,6,9,11,16	25
4	0.9	0.5	23.56	254	1,6,9,11,16	5
5	0.7	0.01	26.53	254	1,6,9,11,16	58
6	0.7	0.1	26.92	254	1,6,9,11,16	67
7	0.7	0.3	26.2	254	1,6,9,11,16	67
8	0.7	0.5	23.29	254	1,6,9,11,16	5
9	0.4	0.01	24.9	254	1,6,9,11,16	25
10	0.4	0.1	27.79	254	1,6,9,11,16	87
11	0.4	0.3	23.23	254	1,6,9,11,16	5
12	0.4	0.5	25.9	254	1,6,9,11,16	58
13	0.1	0.01	27.37	254	1,6,9,11,16	87
14	0.1	0.1	23.76	254	1,6,9,11,16	5
15	0.1	0.3	24.53	254	1,6,9,11,16	28
16	0.1	0.5	23.56	254	1,6,9,11,16	5

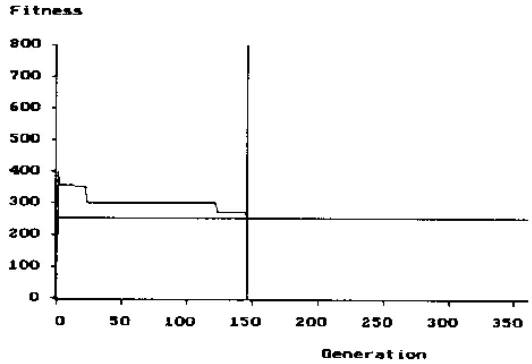

Figure 7. Graph of Fitness Value for Genetic Algorithm Using Elitsm

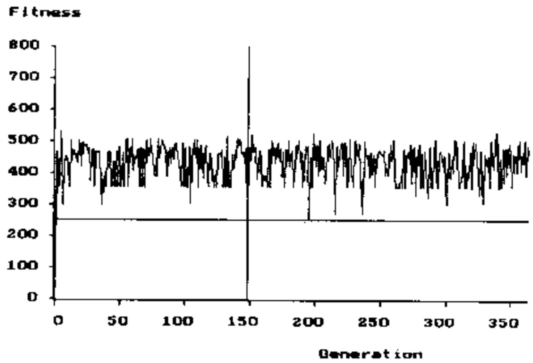

Figure 8. Graph of Fitness Value for Genetic Algorithm Using Roulette Wheel

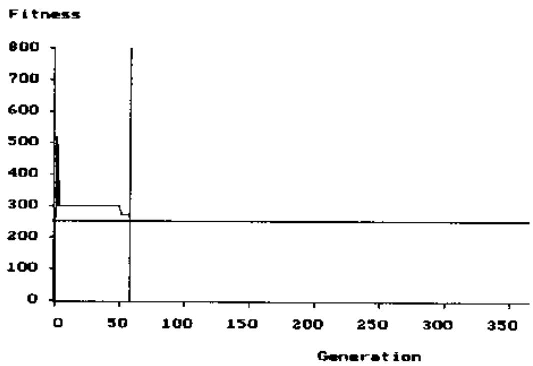

Figure 9. Graph of Fitness Value for Genetic Algorithm Using Elitsm and Roulette Wheel

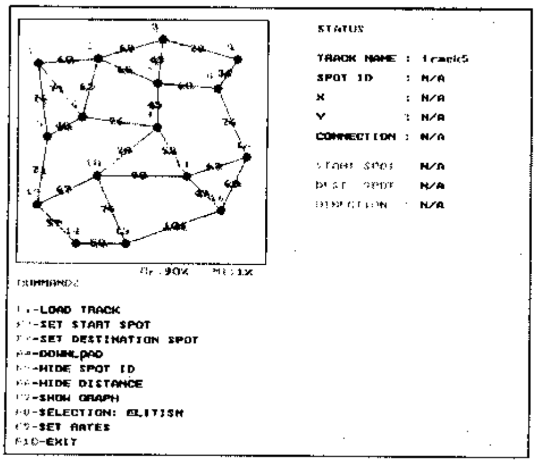

Figure 10. User interface of software for visualization and download parameter

Figure 7 shows graph of fitness value for genetic algorithm using elitsm selection method. This graph shows that the best fitness value for the next generation is always better than the best fitness value for previous generation.

Figure 8 shows graph of fitness value for genetic algorithm using roulette wheel selection method. This graph shows that best fitness value for each generation is random. Figure 9 shows graph of fitness value for genetic algorithm using both elitsm and roulette wheel selection method.

Conclusion

Starting from the experiment and experience, it can be concluded that implementation of genetic algorithm on a microcontroller has been performed well. Genetic algorithm can find the shortest track and the robot can move according to route. However, there are some limitation like number of chromosome per population, type of chromosome and length of chromosome. These are caused by the memory limitation. For the next development, the size of memory can be added, so, that limitation can be overcome.

Reference

[1] Davis, Lawrence. Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold, 1991.

[2] Gen, Mitsuo,and Cheng, Runwei.Genetic Algorithms and Engineering Design. Edited by Hamid R.Parsaei. United States Of America: John Wiley and Sons, 1997.

[3] Goldberg, David Edward. Genetic Algorithms in Search, Optimization, and Machine Learning. United States of America: Addison-Westley, 1989.

[4] Man,K.F, et al. Genetic Algorithms For Control And Signal Processing. London: Springer, 1997.

[5] Advanced Micro Device, Microcontrollers Handbook. California: Advanced Micro Device, 1988.