Intelligent motion of mobile robot in the production area

B. Vaupotič*, J. Balič
University of Maribor, Faculty of Mechanical Engineering, Smetanova ulica 17, SI-2000 Maribor, Slovenia
* Corresponding author: E-mail address: bostjan.vaupotic@uni-mb.si

Received 28.10.2006; accepted in revised form 15.11.2006

ABSTRACT

Purpose: The paper is concerned about intelligent motion of the mobile robot in the production area. The robot is self-learning and gathers the data from the environment by means of sensors. It processes the acquired information and utilizes it for making decisions.

Design/methodology/approach: The concept imitates the natural selection of living organisms, where in the struggle for natural resources the fit individuals become more and more dominant and adaptable to the environment in which they live, whereas the less fit ones are present in the next generations rarely. Some of the improved genetic operations were used for the robot motion.

Findings: The use of those improved genetic operations has proved to be appropriate. By means of them the robot became more and more intelligent in the course of evolution and performed the set task successfully.

Research limitations/implications: The tests were limited only to the space with static barriers. In future, it would be appropriate to test the proposed system also in the space with moving objects and to enable the robot to have full autonomy.

Practical implications: The proposed system enables the robot to move completely independently in the space. The robot complies with simple instructions: come to the goal fastest possible (shortest path) without causing damage to yourself and to the environment.

Originality/value: Originality value of is the implementation of the non-deterministic principles in the decision making strategy of the mobile robot. In learning and independent decision making the robot used some of the improved genetic operations.

Keywords: Artificial intelligence methods; Evolutionary method; Genetic algorithms; Mobile robot

1. Introduction

With the development of the digital computer after 1945 the question arose, when the computers would “overcome” the human in intelligence. The first serious researches, oriented towards artificial intelligence, started in the 70ies of the 20th century. At that time it was prophesied that an intelligent machine comparable in intelligence with the human would be invented in the next decade. However, regrettably (or fortunately) that has not happened so far and, believing the futuristics (science predicting technological advance in the future), it will not happen within some decades.

However, the hitherto efforts of a whole generation of scientists, dealing with the development of artificial intelligence, have given many palpable results which can be beneficially used for optimization of the navigation and searching for the path. The techniques, commonly called fuzzy computations and representing, in fact, a group of three different techniques, are in the foreground: neural networks, evolutionary algorithms and fuzzy logic and mutual combinations of these three basic techniques. In addition to the techniques of fuzzy computation, the efforts for the development of artificial intelligence include also the techniques of the control of chaotic processes (e.g. control of global weather), knowledge-based systems, artificial life etc.
In fact, those three techniques are based on imitating the natural processes, which is a permanent feature in researching artificial intelligence. The oriented neural networks are a simplified model of biological neural networks (brain), the genetic algorithms imitate the natural selection and the genetic code of living species, whereas, the fuzzy logic imitates the psychic processes in the biological brain. In other words, if comparison with the human intelligence is made, it can be imagined that:

- the human genetic code assures the innate capabilities (that one learns how to speak, that one has the memory capacity, that the human has two hands etc.), i.e. it gives the accumulated "knowledge" and/or it ensures adaptation to the environment of the biological species - a parallel to genetic algorithms,
- the brain ensures fast learning and adaptation of the individual (not species) to current circumstances in the environment - a parallel to neural networks,
- the psychic processes ensure to the brain a faster and more efficient learning than ensured by the bare neural networks with the learning technique of "the stick and the carrot" and/or conditioned learning - a parallel to fuzzy logic,
- the accumulated social knowledge in books, culture or conscience of a certain group of people is a parallel to knowledge - based systems.

This paper is limited only to one sub - technique of fuzzy computation: genetic algorithms which belong to the group of evolutionary algorithms.

Optimization of navigation and looking for the path can be ranked into the group of difficult problems. The problem appears in different areas: looking for the path by the mobile robot, transport systems in company, ship navigation, planning of visits of sights considering the wishes of the participants etc. By changing the environment the problem becomes dynamic. Due to highly specific characteristics of such problems each of them needs its heuristic algorithm which becomes complex and, consequently, difficult for additional requirements in order to satisfy all the needs. The problem can be successfully solved by the genetic algorithm. So far, similar problems have been addressed by many authors [1-10].

2. Evolutionary and genetic algorithms

The term "evolutionary algorithm" is used as a common term denoting computer - aided solving of the optimization problems which are solved by the so - called computer models of evolutionary processes. So far, different models of evolutionary algorithms have been developed, however, the following algorithms have proved to be most useful in practice [11]:

- genetic algorithms (GA)
- evolutionary programming (EP)
- classification systems (CS)
- evolutionary strategies (ES) and
- genetic programming (GP)

In fact, all five evolutionary algorithms incorporate the same concept of selection over the population of different systems (subjects). The characteristics of each subject are recorded with the so - called genetic operators (DNA, chromosomes, genes ...) which can be transmitted to the generation of descendants after the crossover of the parent generation (procreation). Due to selection only the fittest descendants can survive, whereas additional improving is ensured by mutation [12].

In fact the genetic algorithm was invented by nature on the planet Earth about three billion years ago upon the appearance of the organic life on our planet. Charles Darwin discovered it again for the mankind in the 19th century, describing it in the following way: "Only the fittest subjects survive in the nature in the struggle for the space and food; in addition, they must know how to adapt well to the changes in the environment. Each subject in the nature is a unique being, although the subjects of the same species have very similar properties. The properties themselves are defined by the genetic code. Thus, each subject is, in fact, controlled by genes. Sets of genes form a chromosome which is a key to survival in the unfriendly environment. The core of evolution is the constant changing of the properties of a species, which, in fact, is changing of the genetic material of the species. The driving force of the evolution are the natural selection (the weak and the unfit die off) and combining of the genetic material taking place during the reproduction (mating). Only the fittest subjects can survive and mate further, which is called survival of the fittest. This phenomenon is called evolution. All genes of a generation of a certain species represent the gene space. The evolution starts when the subjects start to mate; thus, they combine the genes mutually and create new genetic material belonging to the next generation. Thus, a new gene space is created, containing fitter descendants than those from the previous generation. The segments of chromosomes of two parents mutually mix; they create new chromosomes and, consequently, the possibility of existence of new fitter subjects. Repeating of mating and selection improves the gene material".

J. M. Holland was the first to programme the genetic algorithms on the computer in early 70ties of the preceding century, but, in fact, they were finally developed in 1992.

The genetic algorithm is used as an optimization algorithm. Consequently, it belongs to the so - called optimization methods of searching for optimums (global maximums or minimums) [13].

2.1. Algorithm

There are several definitions of the algorithm, the most general and simple one being that the algorithm is a sequence of precisely determined steps leading to the solution of the problem within finite time. Most often the algorithms are said to be mathematical processes or processes which we are able to automate and, sometimes, even to robotize.

Algorithms consist of a set of component parts or simple commands. It could be claimed that the complication of algorithms is their simplicity. As they consist of so simple commands (steps), they are difficult to plan [14].
2.2. Optimization

The optimization implies that the best of possible solutions is searched for. Almost always it applies that some compromise between good and bad properties of the solution reached must be made. Complex problems with only one solution are rarely met. The optimization methods or search schemes, as often called, search for maximums or minimums (optimums). Among optimums the local and global optimums are distinguished [15].

The optimization algorithms are supposed to find global optimums, but a great majority of the so-called conventional gradient optimization methods get into trouble, if low optimums are present. Figure 1 shows a two-dimensional case of optimization. The gradient optimization methods act so that they calculate the gradient of the function in a certain point, they move in that direction for a certain step, then they calculate again the gradient and move...; when the calculated gradient is small enough, the method is stopped. The gradient is positive, if the maximum is searched for, and negative, if the minimum is searched for. For the function in figure 1 such method will get into the local minimum, if the point P1 is selected as the initial value, or it will find the global minimum, when starting in point P2. The example is clear only for two dimensions; if there are several dimensions, it is considerably more difficult to estimate whether the optimum is local or global. The possibility remaining is that the space is searched from several points at a time, which, of course, increases the computational exactness of the method by the exponent degree with respect to the number of variables. The genetic algorithms do not know this type of problems and this has been the basic reason for their popularity among engineers in the area of technology after 1992.

![Fig. 1. Representation of optimization method](image)

3. Genetic algorithms

The human started to imitate nature when searching for good methods which would be enough robust not to get into local minimums and which would be enough simple to be capable of searching the spaces with many variables and finding in them the global optimum or even several optimums, where several equivalent solutions exist. From these tendencies to imitate nature the evolutionary algorithm methods have been developed; one of the most important and most successful methods is the genetic algorithm utilizing "mating" between solutions for searching for a new generation of solutions.

In fact, the genetic algorithm was invented by nature. As was mentioned, Charles Darwin called it evolution. The genetic algorithm, presented and named for the first time by J. H. Holland in early 70ies, completely imitates the natural algorithm of evolution [16].

The individual subjects and/or their genetic material were represented by sets of ones and zeros (i.e. binarily); the genetic algorithms, in fact, performed operations over those sets and not over the solutions themselves.

The Holland’s programme, like nature, selected the fittest subjects. Each of them had its "quality value" - variable which served to estimate the quality of the subject with respect to the remaining ones in the population (generation). The higher the evaluation of quality of the subject, the greater the probability of survival. Thus the reproduction operation is obtained which represents copying of the subject from the current into the next generation, fitter subjects having more copies than the less fit.

The next operation is the crossover, i.e., combining the genetic material of randomly selected pairs of subjects. Still another operation in that programme was the mutation, i.e., random changing of the genetic material of the individual subjects (not all). This operation, too, has a similarity in nature, where mutation involves the regeneration of the lost genetic material. When a certain part of the genetic material is lost in any way, the subject generates it again, but at random. This operation proves to be very important, since it may happen that without it the genetic algorithm is caught in the local optimum, from which it cannot find the way any longer. In that case the mutation ensures further evolution (creation of the diversity of the genetic material) [17].

Figure 2 shows the mutual relations of the individual algorithm component parts described so far in this sub-section.

![Fig. 2. Genetic algorithm component parts](image)
4. Looking for the path

Figure 3 presents a model of the production area. Gray rectangles represent the barriers. The aim of the robot is to travel from start (point S) to end (point E) without colliding with barriers. Different paths from initial to final point represent the population of solutions. Each path consists of the path sections which may cross the barrier or not. In case one of the path sections crosses the barrier an incorrect path is in question.

Fig. 3. Three different successful solutions

Various improvements in searching have led to a significant number of different operators which can be divided into simple ones causing a small change in the path (mutation) and more exacting ones creating a descendant with a combination of two or more paths (crossover). Each path in the evolutionary process is evaluated by means of the fitness function.

4.1. Initial population

The initial population can be created randomly or by means of the existing knowledge about the problem. In the random creation many incorrect sections are obtained (Figure 4).

Fig. 4. Initial population with randomly selected paths

To reduce the number of incorrect paths it is possible, when creating the paths, to use the information about the position of barriers and to locate the control points in the corner points of barriers. Figure 5 shows locating of points in the corner points of barriers.

Fig. 5. Initial population with location of points in corner points of barriers

4.2. Presentation of path

The path consists of sections. The individual section is described with two points. The first point represents the beginning and the last point the end of the path. All other points describe the beginning of the path of the new section and the end of the path of the previous section. As the number of sections is not known, the so-called presentation with variable length is in question.

The points are connected into a list of points representing the path (Figure 6). In addition to coordinates each point contains also the information about correctness of the previous section. The section is correct if it does not intersect the barrier. The information is used for evaluation of the path.

Fig. 6. Presentation of path with continuous list

4.3. Evaluation

Fitness of the solution is influenced by three factors, namely the path length, the number of sections and the number of incorrect sections. For evaluation of the path x the linear combination of those factors can be used. The fitness function can be described with the equation:

\[
eval(x) = k_1 \times \text{length}(x) + k_2 \times \text{sections}(x) + k_3 \times \text{incorrect}(x)
\]  

The constants \( k_1, k_2 \) and \( k_3 \) in equation 1 represent the weights for the path length, number of sections and number of incorrect sections. By means of weights the importance of the individual properties of the path \( x \) is adjusted.

4.4. Operator

The following nine operators, into which our knowledge about the problem is incorporated, are introduced for successful solution of the problem.
CROSSOVER combines the paths of two parents. In each point the crossover point is to be found and, thus, the paths are divided into the head and tail. The crossover brings about new paths (Figure 7).

Fig. 7. Path navigation operators (crossover)

MUTATION 1 randomly selects a point on the path and randomly moves it into the near vicinity. The operator proves to be excellent for optimization of the path already found (Figure 8).

MUTATION 2 similarly as the operator in mutation 1 it randomly selects the point, but it moves it into a random position in the entire space. This results in great changes of the path which, however, prove to be useful in the beginning of the evolutionary process (Figure 8).

Fig. 8. Path navigation operators (mutation 1, mutation 2)

ADDING is an operator which divides the random section into two sections by means of a new point (Figure 9).

DELETING acts oppositely to the adding operator. It randomly selects a point in the list and removes it (Figure 9).

CHANGING takes care of changing the order of two consecutive points. The operator is used to eliminate two consecutive sharp angles (Figure 9).

Fig. 9. Path navigation operators (adding, deleting, changing)

CHANGE OF CORNER POINTS is an operator which moves the point to another corner point. The corner point may be located on any barrier. The operator uses the knowledge about barriers (Figure 10).

CORRECT 1 is an operator which moves the point with coordinates inside the barrier (incorrect position) into the corner point of the barrier (Figure 10).

CORRECT 2 is more complex than the operator correct 1. It changes the section intersecting the barrier so that it changes the section into several smaller sections avoiding the barrier (Figure 10).

Fig. 10. Path navigation operators (change of corner points, correct 1, correct 2)

4.5. Different approaches

The basic algorithm for searching for the path uses the random initial population and randomly selects the operators. Such algorithm uses the basic evolutionary principle for searching for the path between barriers. Due to relatively slow progressing various derivatives of the algorithm are used.

CORNER POINT ALGORITHM: The optimum path frequently runs in the corner point of the barrier. This property is utilized for conceiving of the algorithm whose path points represent the corner points of the barriers. The mutation operator replaces the corner point by a random corner point. The crossover operator uses single point crossover Due to smaller space (only corner points are examined) the algorithm is faster and more efficient. The principal disadvantage is that algorithm does not know how to find the optimum path, when it does not run through corner points.

ALGORITHM WITH DIRECT PATH: The main characteristic of the approach is the description of the path of the initial population by means of straight line from the initial to final point. During the evolutionary process the straight line is divided into several sections by means of operators of changing (adding of points, mutation etc.) This algorithm is slower than the corner point algorithm, but more efficient than the random algorithm.

HYBRID ALGORITHM: The hybrid approach selects a part of the initial population randomly, a part by means of corner points and a part by direct paths. In such an approach a wide set of operators can be used. Due to the combination of properties of all the above algorithms this algorithm is fit and robust.
5. Example 1

To represent the working system for the path navigation two examples have been selected, the first of them being shown in figure 11.

The search space is the production area of 100 x 100 m size. It accommodates 9 machines representing the barriers. The transport robot must carry all finished products, located on the collecting place at the end of the production line, into the anticipated store room. During that operation it must not hit any barrier.

The average length of the best paths of each run was 120.67 m. The best path out of 100 runs of the system had 109.09 m length.

Figure 11 shows the initial generation - generation 0. The result of the random searching for the path is bad, of course. Apparently, the paths in the space are laid out chaotically. The paths run also on the places, where the machines are located. The best organism in the initial generation 0 has 170.18 m length.

Figure 12 shows the path of the robot in generation 10. The length of the best randomly created path is 115 m, which means that the path has been reduced for 33 % with respect to generation 0. However, the robot is in collision with the machine, which is not allowed.

The covered paths in generation 30 are much more "tidy" (Figure 13). Some paths of the robot run past all barriers, but they do not represent the shortest path. The length of the best randomly created path without collision is 7 % shorter than the best path in generation 10, which, however, was in collision.

The 50th generation of randomly created paths is already very tidy (Figure 14). Many paths are without collisions. The length of the best randomly created path without collision is 19 % shorter than the best path in generation 30 and 8 % shorter than the generation 10.
In the generation 70 (Figure 15) solutions are obtained, where all paths avoid collisions with the machines. The paths are very short. The length of the best randomly created path is 110 m.

Figure 16 shows the generation 100. The robot covered the path of 109.09 m length representing the shortest path for the example concerned. It did not hit any barrier. The robot successfully performed the set task.

Figure 17 shows the gradual development of the best solutions through generations with respect to their length.

**6. Example 2**

For the example 2 a space of identical size as for the example 1 was selected. However, this time the machines were so arranged that very narrow passages were created. The robot complies with simple instructions: come to the goal fastest possible (the shortest path) without causing damage to yourself and environment. Example 2 is shown in figure 18.
Already in generation 20 a great progress is noticeable. The path, successfully avoiding all barriers, emerges (Figure 19).

Figure 20 shows founded path in generations 56. All paths are without collisions. The shortest path of the civilization was created in generation 61. The length of the shortest path was 120 m, which is the shortest possible path for this example.

7. Conclusion

The paper discusses the approach to searching for the shortest path of the robot by means of the evolutionary method. Some improved genetic operations were used which have proved to be adequate. By means of them the robot become more and more intelligent throughout evolution and performed the set task. In future, it would be appropriate to test the improved operations also in the room with moving objects and to enable the robot to have full autonomy.

References