International Journal of Automation and Computing 1 (2006) 29-34

# A Rough Set GA-based Hybrid Method for Robot Path Planning

Cheng-Dong Wu*, Ying Zhang, Meng-Xin Li
Shenyang Jianzhu University, Shenyang 110168, PRC

Yong Yue
University of Luton, Luton LU1 3JU, UK

**Abstract:** In this paper, a hybrid method based on rough sets and genetic algorithms, is proposed to improve the speed of robot path planning. Decision rules are obtained using rough set theory. A series of available paths are produced by training obtained minimal decision rules. Path populations are optimised by using genetic algorithms until the best path is obtained. Experiment results show that this hybrid method is capable of improving robot path planning speed.

**Keywords:** Rough sets, genetic algorithms, robot, path planning.

## 1 Introduction

Robots are an outgrowth of traditional mechanisms and modern electronics technologies. The use of robots is becoming more and more popular. It is therefore necessary to improve their performance. Among factors for robot performance, path planning is of great importance.

At present, path planning methods can be divided into two categories: traditional methods and intelligent methods. Traditional methods include gradient, grid, enumeration, man-made potential, and graph methods, etc. However, the gradient method can easily sink into a local minimum; graph and enumeration methods cannot be used to deal with optimisation problems with high dimensionality; and the potential method can lose useful solution information. Some common methods for intelligent path planning, which have been used recently, include fuzzy control, neural networks, and genetic algorithms.

Sugibara and Smith[1], applied genetic algorithms to robot path planning, and improved performance. Their algorithms adopted a binary coding system, resulting in increased individual length and complexity of planning. Surmann et al.[2], proposed a path planning method using fuzzy control. Eight ultrasonic sensors were used to provide information about the environment, for path planning by a navigator based on fuzzy control. An accurate path could be obtained quickly using this method, although complexity increased considerably when the number of barriers in the environment increased; therefore decreasing path planning speed. Yu et al.[3], employed neural networks in robot path planning in order to increase speed of planning, however their method is only applicable for situations in which the position of barriers is known; preventing it from conducting real time planning.

Genetic algorithms, are more useful than other methods. However, gaining a population randomly may lead to a large search space, and low ability to delete redundant individuals, both of which affect planning speed. This shortcoming of genetic algorithms is clear, especially when an environment is complicated or there are two or more robots. In recent years, rough set theory has become a new research focus in artificial intelligence, for pattern recognition, machine learning, knowledge representation, knowledge discovery and decision analysis, etc.

This paper proposes a new hybrid method for robot path planning, based on rough sets and genetic algorithms, and described using a warehouse transport application. Rough set theory is introduced to improve population initialisation, when a genetic algorithm is used in order to improve the speed of robot path planning. Minimal decision rules are obtained using rough sets, from which a series of possible paths are pro-

Manuscript received February 15, 2005; revised September 30, 2005.

This project is partially supported by Science Research Funding from the Education Department of Liaoning Province, China (No.J9906065).

*Corresponding author. *E-mail address*: wcd@sjzu.edu.cn

duced. Path populations are optimised, using genetic algorithms, and an optimal path selected. A simulation system has been built to demonstrate this path planning process, and to obtain an optimal path.

## 2 Rough set algorithm

In recent years, rough set theory has become a new research focus in artificial intelligence for systematic analysis, and has been used in pattern recognition, machine learning, knowledge representation, knowledge discovery and decision analysis, etc[4].

### 2.1 The knowledge representation system

When dealing with intelligent data, it is necessary to use symbols to represent knowledge. The basic composition of knowledge representation systems is a set of research objects, in which knowledge is represented using basic characters and attribute values. A knowledge representation system can be expressed by:

$$S = < U, C, D, V, f >$$

where $S$ indicates the system, $U$ is the set of objects, $C \cup D = A$ is the set, the subset $C$ and $D$ are condition and result attributes respectively, $V = \cup_{a \in \wedge} V_a$ is the set of attribute values, $V_a$ indicates the category of $a \in A$, and $f : U \times A \rightarrow V$ is an information function, which appoints the attribute value of each object $x$ in $U$.

This definition of a knowledge representation system can be expressed conveniently using tables. This method of expressing knowledge using tables may be regarded as a special formal language. When expressing the relationship of equivalence using symbols, the data should be called a knowledge representation system (KRS), or an attribute values table of an information system.

### 2.2 The simplification of knowledge and core

When rough set theory is applied, it is necessary to delete abundant basic categories, and frequently to simplify knowledge while keeping an initial category in the knowledge base. In expressing the idea of knowledge simplification, two basic concepts, simplification and core, are very important. Based on these two concepts, abundant attributes are analysed, and knowledge handled, simultaneously. First, repetitive examples and abundant attributes are deleted, then, a minimal simplification is made in terms of logical rules gained. Finally, minimal rules are drawn to deal with the data.

## 3 Derivation of decision rules for robot paths

### 3.1 Workspace of the robot

This research considers a warehouse environment, in which a robot is used to transport goods. There are a number of modelling methods for path planning in mobile robots, such as grids, and images of peak methods, etc. A grid method does not have a relationship with the topology structure of barriers[5], search space is very large, and there are problems of calculation efficiency. The image pattern of peak method, depends on t-structure as a high level description of the environment, and can lead to an exact solution in path planning. However, when the number of obstacles increases, or the shape of obstacles is complex, the complexity of path planning algorithms increase exponentially, and the problem may become unsolvable.

This research therefore adopts the grid method, which divides the workspace of a mobile robot into a series of grid squares, with two-value information to describe the location of each square. For convenience of use, the workspace of the mobile robot is divided into many square grids of the same size, (a non-square rectangle can be filled up with squares). The size of grids must be appropriate, to ensure that the robot can pass through them with acceptable accuracy. The sequence law of grids is employed, as shown in Fig.1, where a grid is coded from 1 to 100. The grid square marked $S$ is the start point for the robot, and $G$ the goal point.

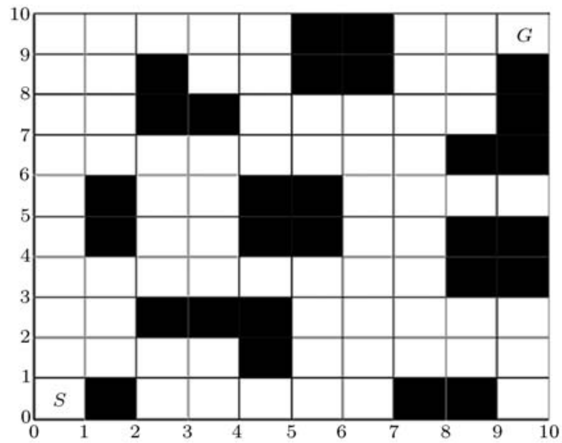



Fig.1 Workspace of the mobile robot represented by a grid

### 3.2 Construction of a decision table

The workspace of a robot can be indicated in a grid. Assuming the sequence number of the present location of a robot is $p_i$ (except boundary points), and the sequence number of the goal grid square is 99, there are 8 possible directions that the robot may move, as shown in Fig.2.

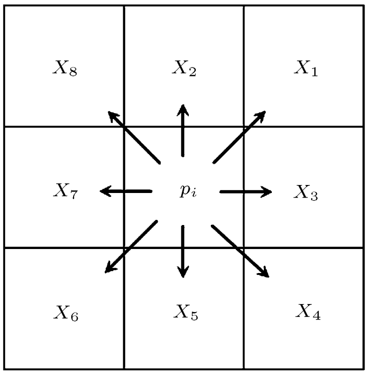



Fig.2 Direction of robot moves

These 8 directions, are regarded as condition attributes (i.e. $C = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8\}$) to be quantified; condition attributes are expressed using a value from 1 to 3. Therefore, an initial decision table is established as shown in Table 1.

Table 1 Initial decision table for path planning

| $U$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $Y$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 1 |
| 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 3 | 1 |
| 4 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 |
| 5 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 |
| 6 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 3 | 1 |
| 7 | 2 | 2 | 2 | 3 | 1 | 1 | 1 | 1 | 1 |
| 8 | 2 | 2 | 2 | 3 | 1 | 1 | 1 | 2 | 1 |
| 9 | 2 | 2 | 2 | 3 | 1 | 1 | 1 | 3 | 1 |
| 10 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 1 | 1 |
| ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ |
| 4374 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 8 |

## 3.3 Simplification of the decision table

Simplification of a decision table, reduces condition attributes, while retaining solution calculation ability. The simplification of a decision table is performed using the following steps:

1) Simplify condition attributes, i.e. delete some columns in the decision table,

2) Delete repetitive rows, and

3) Delete abundant attribute values.

The simplification of condition attributes, involves the need to delete some condition attributes while maintaining the consistency of the decision table. That is, after one of the condition attributes is deleted, the table is checked to see if a condition attribute in the same row can yield the same decision value as before. In a compatible data table, an optimum attribute set can be selected using rough set theory.

**Quality:** If $C = \{X_1, X_2, \cdots, X_n\}$ is the attribute set, and $\cap(C - \{X_i\}) = \cap C$, then $X_i$ can be omitted from $C$; otherwise, $X_i$ cannot be omitted from $C$.

First, repetitive samples in a decision table are merged, then attributes of samples are simplified, based on their characteristics, to delete every attribute column. The consistency of the decision table is then checked. After calculation, attribute $\{X_4\}$, $\{X_6\}$, and $\{X_8\}$ are considered abundant attributes, and should be deleted. Finally, repetitive rows in the decision table are merged once more.

The number of samples is decreased to 162 after deleting abundant attributes, as shown in Table 2.

Table 2 Decision table with abundant attributes deleted

| $U$ | $X_1$ | $X_2$ | $X_3$ | $X_5$ | $X_7$ | $Y$ |
|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 1 | 2 | 1 |
| 3 | 2 | 2 | 2 | 1 | 3 | 1 |
| 4 | 2 | 2 | 2 | 2 | 1 | 1 |
| 5 | 2 | 2 | 2 | 2 | 2 | 1 |
| 6 | 2 | 2 | 2 | 2 | 3 | 1 |
| 7 | 2 | 2 | 2 | 3 | 1 | 1 |
| 8 | 2 | 2 | 2 | 3 | 2 | 1 |
| 9 | 2 | 2 | 2 | 3 | 3 | 1 |
| 10 | 1 | 2 | 2 | 1 | 1 | 2 |
| ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ | ⋯ |
| 162 | 1 | 3 | 3 | 3 | 3 | 8 |

## 3.4 Minimal decision rules

It is clear that not all decision rules are necessary for decision algorithms. Some rules can be deleted, and the decision process not be affected, as in the method described below:

Order $F$ is a basic algorithm, $S = (U, A)$ is a knowledge representation system, and $F$ possesses basic rule sets with the same result $\psi$, which can be represented as $F_\psi$. The cause sets of decision rules belonging to $F_\psi$, can be represented as $P_\psi$.

When $|s \vee P_\psi \equiv \vee\{P_\psi\text{-}\{\theta\}\}$, the basic decision rules $\theta \rightarrow \psi$ in $F$ can be omitted, where $\vee P_\psi$ is the decomposition of all formulas. Otherwise, basic decision rules in $F$ cannot be omitted, and the decision rule set $F_\psi$ regarded as independent. If all rules of the subset $F'_\psi$, of the decision rule set $F_\psi$, are independent, and $|s \vee P_\psi \equiv \vee P'_\psi$, then the subset $F'_\psi$, of the decision rule set $F_\psi$, is called the simplification of $F_\psi$. When decision rules for basic algorithm $F$ are all simplified rules, and each basic decision rule $\theta \rightarrow \psi$ in $F$, then $F_\psi$ is simplified, and the basic algorithm $F$ is minimal.

Table 3 shows minimal decision rules, which cannot be simplified further.

Table 3 Table of minimal decision rules

| $U$ | $X_1$ | $X_2$ | $X_3$ | $X_5$ | $X_7$ | $Y$ |
|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 | – | 1 |
| 2 | 2 | 2 | 2 | 2 | – | 1 |
| 3 | 2 | 2 | 2 | 3 | – | 1 |
| 4 | 2 | 2 | 2 | – | 1 | 1 |
| 5 | 2 | 2 | 2 | – | 2 | 1 |
| 6 | 2 | 2 | 2 | – | 3 | 1 |
| 7 | 1 | 2 | 2 | 1 | – | 2 |
| 8 | 1 | 2 | 2 | 2 | – | 2 |
| 9 | 1 | 2 | 2 | 3 | – | 2 |
| 10 | 1 | 2 | 2 | – | 1 | 2 |
| ... | ... | ... | ... | ... | ... | ... |
| 80 | – | 1 | 3 | 3 | 3 | 8 |

## 4 Simulation experiments

A hybrid method involving rough sets and genetic algorithms can be conducted as follows[6~8]:

1) Establish the work model of a robot in a grid, within which barriers are generated randomly,

2) Produce randomly, an initial population at a fixed scale, and train it using rough set path decision rules to generate a series of paths,

3) Initialise a counter of genetic generations, options = 0, and

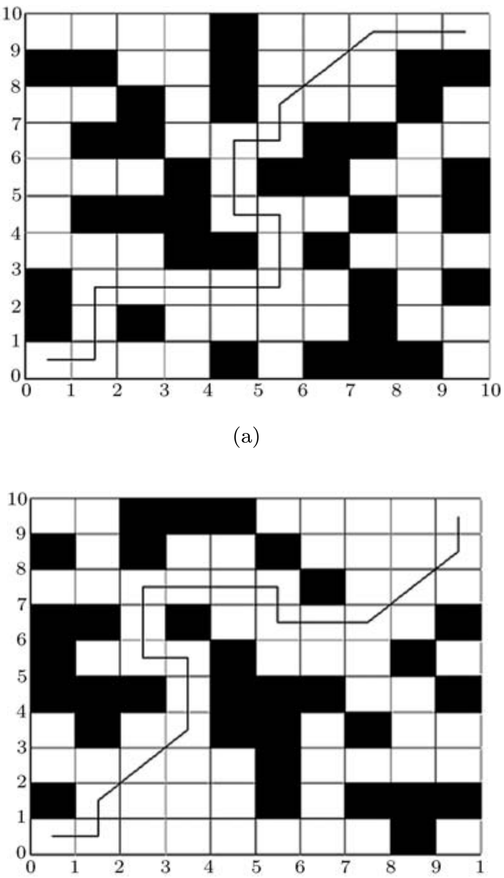
4) Plan a robot path using genetic algorithms, consisting of selection, crossover, and mutation operators, in which the crossover rate can be defined using the following self-adaptive selection formula[9~12]:

$$p_c = k_1(f_{\max} - f')/(f_{\max} - f_{\min}) + k_2$$

where $f_{\max}$ and $f_{\min}$ are maximal and minimal values of the fitness function respectively, $f'$ the layer of individuals that cross, and $k_1$ and $k_2$ contraction coefficients of the algorithms which satisfy $k_1 + k_2 = 1$. Here, it is assumed that $k_1 = 0.3$ and $k_2 = 0.7$.

Using the algorithms described above, simulation experiments were carried out. A grid of $10 \times 10$ squares, was selected to represent the working environment of a robot. A population scale was set to 30, i.e. $popsize = 30$, and the length of individuals was set to 100, i.e. $stringlength = 100$.

In addition, a mutation rate of 0.01 was used. The position and number of barriers was generated randomly, as shown in Fig.3. The result in Fig.3(a) is gained using 13 generations, i.e. $options = 13$, where the quantity of barrier is $za = 35$, and the fitness function value $f = 0.1786$. In Fig.3(b), the result is gained using 10 generations, i.e. $options = 10$, where the quantity of barrier is $za = 32$, and the fitness function value $f = 0.1734$.




(a)

(b)

Fig.3 Simulation results

The path planned, can backtrack voluntarily when a barrier blocks the way ahead and start searching new paths, as shown in grid square (5.5, 4.5) in Fig.3(a), and grid square (3.5, 5.5) in Fig.3(b).

The relationship between genetic generation and fitness for the two situations in Fig.3, are shown in Fig.4 respectively. From the figures, it can be seen that the fitness of Fig.3(a) became stable after 13 generations, while for Fig.3(b) fitness became stable after 10 generations.

The proposed hybrid method, was used to plan a work model in which there were 10, 15, 20, 25 and 30 barriers respectively, in which the position of barriers was generated randomly. Simulation results are summarised in Table 4. The table also shows results gained using a traditional genetic algorithm method, with the same parameters as the new method, to provide a comparison between the two. In the table, $\bar{f}$ is the average fitness value, $f_{opt}$ optimum fitness value, and $\bar{N}_{opt}$ the average number of generations before an optimum result was reached. Method 1, is a traditional genetic algorithm method which uses binary coding[1], while method 2, is the proposed new hybrid method. It is clear that the new hybrid method has a faster planning speed than the traditional algorithm, while still obtaining an optimal path.

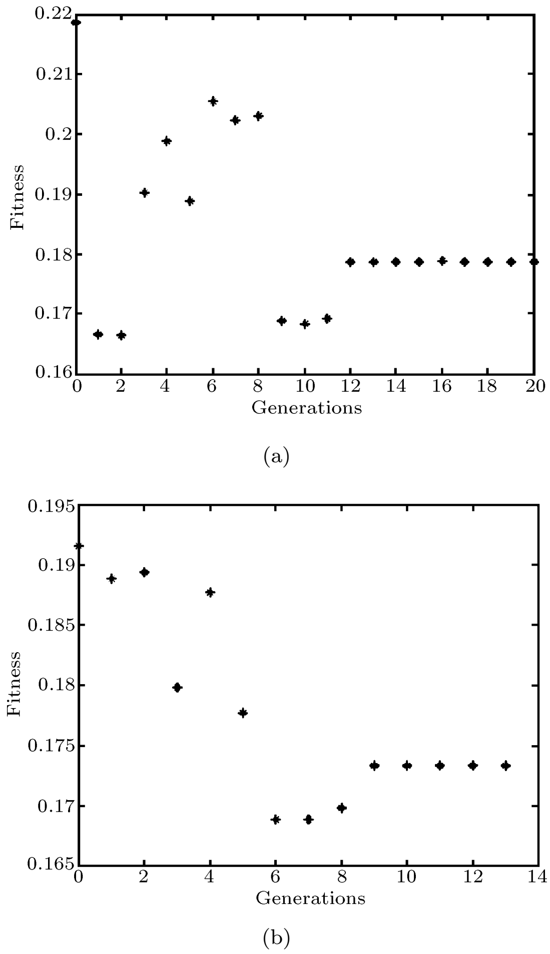



Fig.4 Relationship between genetic generation and fitness

Table 4 Comparative results between the new method and a traditional method

| Number of barriers | $\bar{f}/f_{opt}$[%] | | $\bar{N}_{opt}$ | |
|---|---|---|---|---|
| | Method 1 | Method 2 | Method 1 | Method 2 |
| 10 | 99.999 | 99.85 | 125 | 12 |
| 15 | 99.813 | 99.36 | 212 | 15 |
| 20 | 99.821 | 99.31 | 378 | 22 |
| 25 | 99.796 | 99.02 | 295 | 19 |
| 30 | 99.797 | 99.42 | 439 | 24 |

## 5 Conclusions

This paper has proposed a new hybrid method based on rough sets and genetic algorithms, for robot path planning. Rough set theory is suitable for dealing with large amounts of incomplete data, while genetic algorithms provide good search ability in large state spaces. Incorporating the strengths of the two methods, the proposed new hybrid method has improved speed in robot path planning.

## References


[1] K. Sugibara, J. Smith. Genetic algorithms for adaptive motion planning of an autonomous mobile robot. In *Proceedings of 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Hawaii, HI, USA, IEEE Press, pp. 138–143, 1997.

[2] H. Surmann, J. Huser, J. Wehking. Path planning for a fuzzy controlled autonomous mobile robot. In *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, St.Augustin, Germany, IEEE Press, vol. 8-11, pp. 1660–1665, 1996.

[3] J. Yu, K. Valeri, Z. Sun, N. Hiroyuki. Fast algorithm for path planning based on neural networks. *Robots*, vol. 23, no. 3, pp. 201–205, 2001.

[4] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Nowowiejska, Warsaw, Poland, pp. 15–19, 1990.

[5] H. D. Zhang, R. Zheng, Y.W. Chen. Present situation and future development of mobile robot path planning technology. *Journal of System Simulation*, vol. 17, no. 2, pp. 439–443, 2005.

[6] T. Shibata, T. Fukuda. Intelligent motion planning by genetic algorithm with fuzzy critic. In *Proceedings of the 8th IEEE International Symposium on Intelligent Control*, Tsukuba, Japan, IEEE Press, vol. 25-27, pp. 565–570, 1993.

[7] T. Ozyer, R. Alhajj, K. Barker. Utilizing rough sets and multi-objective genetic algorithms for automated clustering. In *Proceedings of 4th International Conference on Rough Sets and Current Trends in Computing, RSCTC 2004*, Uppsala, Sweden, vol. 1-5, pp. 567–572, 2004.

[8] M. Inuiguchi, T. Tanino. Generalized rough sets and rule extraction. In *Proceedings of the 3rd International Conference on Rough Sets and Current Trends in Computing, RSCTC 2002*, Malvern, PA, USA, vol. 14-16, pp. 105–112, 2002.

[9] Z. Cai, Z. Peng. Cooperative coevolutionary adaptive genetic algorithm in path planning of cooperative multi-mobile robot systems. *Journal of Intelligent and Robotic Systems*, vol. 33, no. 1, pp. 61–71, 2002.

[10] D. Goldberg. Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley, New York, USA, pp. 1–25, 1989.

[11] D. Beasley, R. D. Bull, R. Ralph. An overview of genetic algorithms. *University Computing*, vol. 15, no. 2, pp. 58–69 and vol. 15, no. 4, pp. 170–181, 1993.

[12] X. Xu, Y. Chen. A method for trajectory planning of robot manipulators in Cartesian space. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, Indianapolis, USA, IEEE Press, vol. 28, pp. 1220–1225, 2000.

**Cheng-Dong WU** Ph.D., Professor, received B.Sc. in Electrical Automation from Shenyang Jianzhu University of. in 1983, M.Sc. in Control Theory and its Applications from Tsinghua University in 1988, and Ph.D. in Industrial Automation from Northeastern University in 1994, China, respectively. He is now Director, Institute of Artificial Intelligence and Robotics, Northeatern University, Shenyang, China. He was the principal investigator of over 20 projects in robot control and path planning, image processing, intelligent traffic systems, smart home systems and wireless network communication technologies. He has published 9 books and over 150 journal and conference papers.

**Yong Yue** is a Principal Lecturer at the University of Luton, UK. He holds a B.Sc. in mechanical engineering from the Northeastern University, China, and a Ph.D. from Heriot-Watt University, UK. He is a Chartered Engineer and a Member of the Institution of Mechanical Engineers, UK. Dr. Yue has been working in academia for 15 years following his 8 years of industrial experience. His main research interests include CAD/CAM, geometric modelling, virtual reality and pattern recognition.

**Ying Zhang** graduated form Shenyang Jianzhu University, China, in 2003. She received an M.Sc. in Control Theory and Control Engineering from Shenyang Jianzhu University. She is currently pursuing a Ph.D. degree in Shenyang Institute of Automation, China. Her research interests include intelligent robot control, pattern recognition and image processing.

**Meng-Xin Li** is Associated Professor in the School of Information and Control Engineering, Shenyang Jianzhu University, China. She received her B.Sc. and M.Sc. in Control Theory and Control Engineering from Shenyang Jianzhu University, and her Ph.D. degree in Pattern Recognition from the University of Luton, UK. Her research interests include neural networks, rough set theory and image recognition.