Obstacle Avoidance by Using Modified Hopfield Neural Network

Panrasee Ritthipravat
Center of Operation for Field Robotics Development (FIBO), King Mongkut's University of Technology Thonburi. 91 Suksawas road Tongkrui Bangkok Thailand.
Phone (662)470-9339, Fax. (662)470-9691
e-mail: pan@fibo.kmutt.ac.th

Kenji Nakayama
Dept. of Information and System Eng. Faculty of Engineering, Kanazawa University. 2-40-20, Kodatsuno, Kanazawa 920 Japan
Phone (81)76-234-4896, Fax. (81)76-234-4900
e-mail: nakayama@t.kanazawa-u.ac.jp

Abstract

In this paper, path planning of a mobile robot by using a modified Hopfield neural network is studied. An area, which excludes obstacles and allows gradually changing of activation level of neurons from a starting point to a goal, is derived. Path can be constructed in this area by searching the next highest activated neuron. Even though asymmetric weight matrix is used, decreasing of system energy can be investigated. By comparison to a symmetric weight network, simulation results show more effective path, which could be generated by this algorithm. Simulation results showed constructed path from the starting point to the goal, which can avoid obstacle successfully.

Keywords a modified Hopfield neural network, Robot path planning, Path generator

1. Introduction

Obstacle avoidance is one of the most interested issues in mobile robot field. It has been applied on either a single robot or multiple robot system. Many researchers proposed their algorithms to solve obstacle avoidance problem either static or dynamic environment.

Y. Arai, T. Fujii, H. Asama, Y. Kataoka [1] proposed collision avoidance in multiple robot system. Their robots are equipped with LOCISS. Each robot will recognize whether another robot or obstacle. The rule matrix for collision avoidance is predetermined. Reinforcement learning is applied to acquire adaptive behavior.

Mochida, Ishiguro, Aoki and Uchikawa [2] presented a behavior control method inspired by living organisms which immune and emotion systems were introduced to cope with dynamic changing environment. The idea of emotional system was used to model frustration function, which would be used to determine robot behavior.

Ishiguro, Watanabe and Uchikawa [3] introduced using an immune system concept to cope with dynamic changing environment. Antibody and antigen matching were used to determine the robot behavior. Various kinds of robot behavior were represented as antibodies. The environment information was also set as antigens. The immune network system selected an antibody, which was most suitable for current situation or an antigen.

R. Mehrotra and D. M. Krause [4] presented an obstacle free path planning by using a quadtree coding method.

M. G. Lagoudakis [5] proposed dynamic path planning and obstacle avoidance by using the Hopfield neural network. The idea of receptive field is presented. The external input is added in a goal and obstacles. Such the target is highest activated neuron. The activation levels of neurons are spread decreasingly around the target as wave propagation. However the Hopfield neural network cannot guarantee the monotonically decreasing from the goal because it employs symmetrical connection weights. Activation level can propagate both directions. This easily causes local peaks of activation level.

In this paper, an asymmetric weight matrix is proposed in order to form a distinguishable area of activation level from the starting point to the goal. The path can be generated easily by considering highest activated neurons in this area. Next section, the Hopfield network and selected activation function are described. Then weight matrix determination is presented. System model, which compose of the Hopfield network and a path generator, is expressed. Simulation results and discussion of the paper are provided.

2. Hopfield Network

The Hopfield neural network was proposed in1982, by physicist John J. Hopfield [6]. It can be used to solve information retrieval or optimization problems. It has a recurrent feature. Output of all another neurons, v_j, are fed back, weighted and summed to be the input to a neuron u_i. Only one neuron, i, is selected randomly and updated its state in each time.

$$u_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij} v_j + \theta_i . \qquad (1)$$

where u_i is input potential of neuron i

v_j is output of neuron j

w_{ij} is connection weight from neuron j to i

θ_i is bias of neuron i

According to M. G. Lagoudakis [5], external bias are added in the goal and obstacle units as:

$$\theta_i(t) = \begin{cases} +\infty & target-unit \\ -\infty & obstacle-unit \\ 0 & otherwise \end{cases} \qquad (2)$$

The effect of adding bias in Eq(2) is that the goal unit will be maximally activated. The obstacle unit is deactivated independently of other units.

The original Hopfield network assumes zero self-coupling, $w_{ii} = 0$, and symmetrical weights, $w_{ij} = w_{ji}$. This symmetrical property guarantees that the system energy function always converges to the basin of attraction or the equilibrium state. This means no state changed anymore.

Many forms of activation functions can be used to classify an output level of a neuron accordance with its input. In the discrete Hopfield network, we can use a sign function to classify output to be either 1 or 0 as:

$$sign(u_i) = \begin{cases} 1 & u_i \geq 0 \\ 0 & u_i < 0 \end{cases} . \qquad (3)$$

Such output is discrete with value $\{1,0\}$. The output is 1 means the neuron is activated or fired. While the output is 0 means the neuron is deactivated or quenched. For the analog Hopfield network, a sigmoid function or a hyperbolic function can be used. For the sigmoid function, the output is analog with value [0,1] as:

$$\phi(u_i) = \frac{1}{1+e^{-u_i}} . \qquad (4)$$

For the hyperbolic function, the output is analog with value [-1,1] as:

$$\phi(u_i) = \frac{1-e^{-u_i}}{1+e^{-u_i}} . \qquad (5)$$

In this paper, system state is set with value $[0,1]$.

By using the hyperbolic function, the activation function can be presented as:

$$\phi(u_i) = \begin{cases} \dfrac{1-e^{-u_i}}{1+e^{-u_i}} & u_i \geq 0 \\ 0 & u_i < 0 \end{cases} . \qquad (6)$$

Fig. 1 shows activation functions in each form. The activation function used in this paper is shown with a thickest line in this figure.

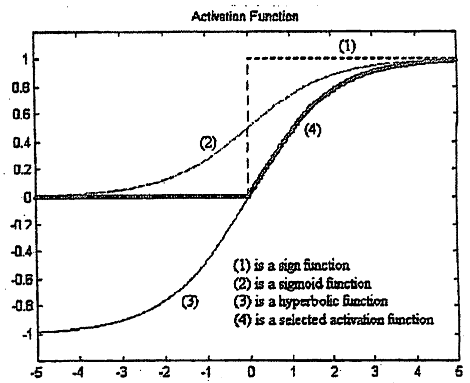

Fig. 1 shows activation functions

3. Weight Matrix Determination

M. G. Lagoudakis [5] mapped the Configuration Space, C into neuron space. Each neuron corresponds to subset C_i of C. Such all neurons can be represented as overall Configuration Space. Each neuron has a receptive field, RF_i which is subset of units of its neighborhood. Each unit i connects to the units in its RF_i. There are no connections outside the receptive field. The symmetrical property is concerned such if any unit i is in RF_j then unit j is in RF_i and vice versa. Weight is determined as a decreasing function, which depends on Euclidean distance $\rho(i,j)$ between the units. Weight between unit i and unit j can be represented as:

$$w_{ij} = f(\rho(i,j)), \qquad (7)$$

The function can take various forms as:

$$f(\rho) = e^{-\gamma\rho^{\alpha}} . \qquad (8)$$

where γ is a positive real number

α is used to span the width of a decreasing function. It may be 2, 4 or more.

In this paper, weight matrix is determined by using the distance between the goal and each neuron. The strongest weight will be set on a neuron, which is nearest to the goal direction. The weight will be decreased spread out of the strongest weight as shown in Fig. 2

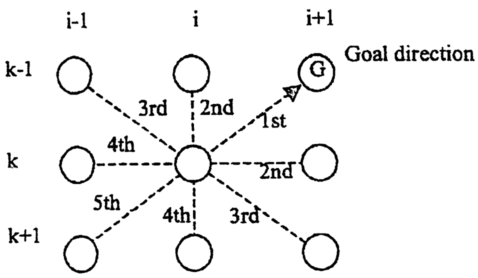

Fig. 2 shows relations of weight in each unit

As shown in Fig. 2, the largest weight is the weight from a neuron (k,i) to a neuron (k-1,i+1), which locates toward the goal direction. The connection weights from the neuron (k,i) to (k-1,i) and (k,i+1) have the 2[nd] largest value. By setting the weight in this manner, neurons, which close to the goal direction, have higher activation level and are activated easily. Such the level of activating can be distinguished obviously. However the symmetrical property could not be conserved as shown in Fig.3. The weight from a neuron (k+1,i+1) to (k,i) is stronger than weight from the neuron (k,i) to (k+1,i+1).

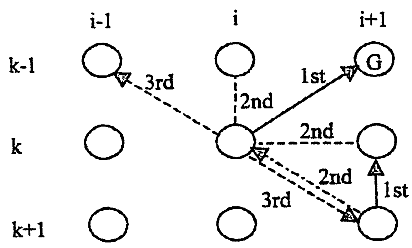

Fig. 3 shows asymmetric weight feature

As expressed in Eq (8), the connection weigh can be set accordance with the nearness to the goal by varying γ in each direction.

4. System Model

As discussed above, the level of activated neuron can be separated noticeably. Path can be generated from the highest activated neuron from the starting point to the goal. Such system model is composed the modified Hopfield network and the path generator as shown in Fig. 4.

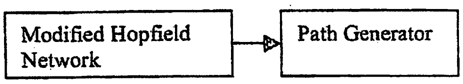

Fig. 4 shows system model

5. Mechanism of Path Generator

In the modified Hopfield neural network, the starting point and goal states are always set to be maximally activated neurons. Such there are two-peaked surface in an activated neuron space. The levels of activated neurons are propagated decreasingly from them. The obstacles are also set to be deactivated neurons and they show the valley-like. By setting the connection weight as discussed above, the levels of activated neurons can be distinguished obviously from the starting point to the goal. By searching the highest activated neuron from the starting point to the goal, path can be easily obtained. However in the Hopfield neuron network, the high unit state trends to activate its neighboring neurons. Some neurons are activated improperly. For example, the goal is set in one side of a wall of obstacles. The starting point is set to another side of the wall as shown in Fig. 5.

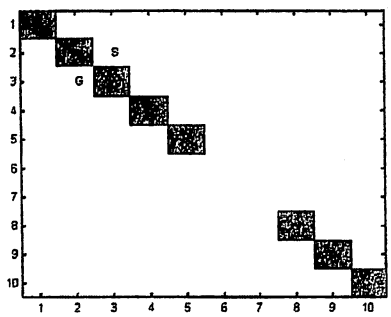

Fig. 5 Starting point (S) and goal (G) are set in different side of wall of obstacles (O).

The starting, goal and obstacle units are marked by 'S', 'G' and 'O', respectively. The neurons, which are in corner, may be highly activated because they close to the goal and the starting point. Such selecting highest neuron from its neighborhood is insufficient condition for constructing the good path. 'good path' means there are less numbers of stepping to move from the starting point to the goal. Path generator should concern next step of its neighboring neurons. The starting neuron is set to be a current unit. By considering its closet neurons, two highest activated neurons are searched. The level of activating of their adjacent units, excluding the current neuron and its neighborhood, are compared and selected the highest one. The selected neuron will be updated to be the current unit. Following this manner, the path can be constructed. Not only considering the next step of its neighboring units but the path generator should also concern prevention of moving back to the previous unit in order to avoid loop constructing.

In order to compare the level of activating of adjacent units, a diagram of current neuron (k,i) and its vicinity are presented in Fig. 6.

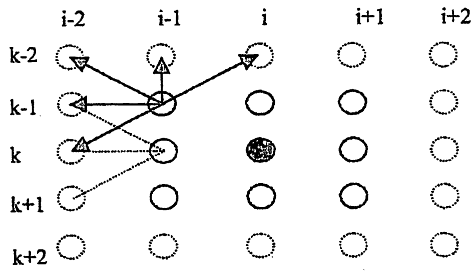

Fig. 6 shows a neuron (k,i) and its vicinity

There are 8 neighboring neurons around the neuron (k,i). Each neighboring unit has its nearby units. Neuron (k-1,i-1) has 5 next closest units, i.e., (k,i-2), (k-1,i-2), (k-2,i-2), (k-2,i-1), (k-2,i). Neuron (k,i-1) has 3 next neighboring units, i.e., (k+1,i-2), (k,i-2), (k-1,i-2). By considering in this way, the adjoining units in each direction can be obtained. Three highest units in each direction are averaged. Two of them corresponding to two highest neighboring units are used in comparing for next step selection.

6. Simulation Results

The simulation results are composed of 2 parts, i.e., comparison to a symmetric weight matrix and applying to another applications. A network composes of 10x10 neurons. The starting, goal and obstacle units are marked by 'S', 'G' and 'O', respectively. For the modified Hopfield network, the starting and goal states are fixed to be maximally activated neurons. The obstacle states are set to be deactivated units. Initial state of another neurons are selected randomly and scaled in a range of [0,0.5]. The simulation was done until approached to an equilibrium state. The surface of equilibrium states is plotted for more understanding. The system energy is shown to investigation of the decreasing of energy. Path generator constructs a path from the activated neurons in an equilibrium state. The constructed path is mapped to a Configuration space of robot with 14x14 units.

By comparison to a symmetric weight matrix, weight matrix determination by using M. G. Lagoudakis [5] method is used. The environment as shown in Fig. 5 is tested. The simulation results are shown in Figs. 7-8. Another three applications are simulated to show an effective path generated by our model. The simulation results are shown in Figs.9-13.

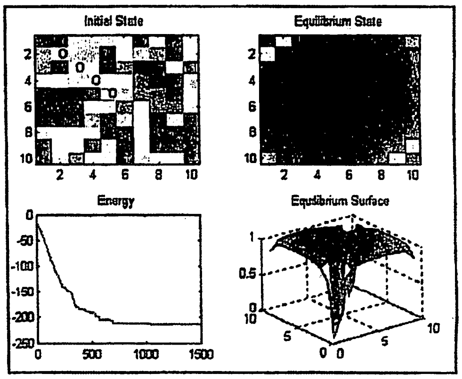

Fig. 7a shows system states and system energy for symmetric weight matrix approach.

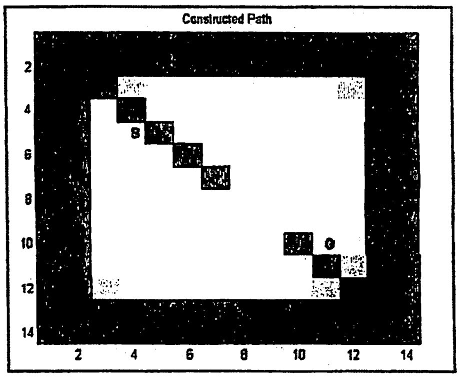

Fig. 7b shows constructed path by using symmetric weight matrix approach for application 1.

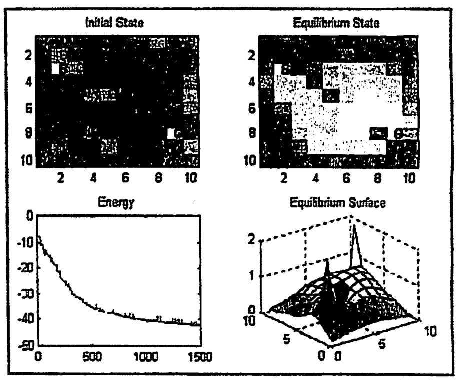

Fig. 8a shows system states and system energy by using asymmetric weight matrix approach for application 1.

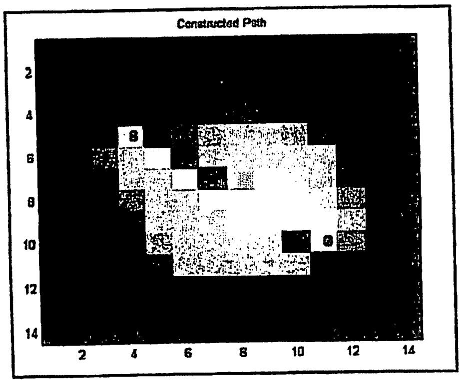

Fig. 8b shows constructed path by using asymmetric weight matrix approach for application 1.

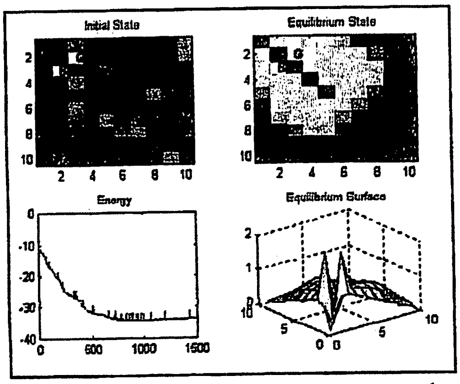

Fig. 9a shows system states and system energy by using asymmetric weight matrix approach for application 2.

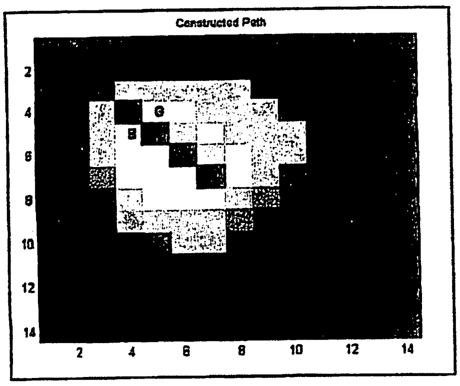

Fig. 9b shows constructed path by using asymmetric weight matrix approach for application 2.

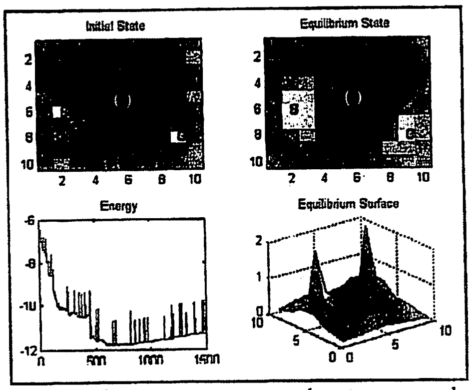

Fig. 10a shows system states and system energy by using asymmetric weight matrix approach for application 3.

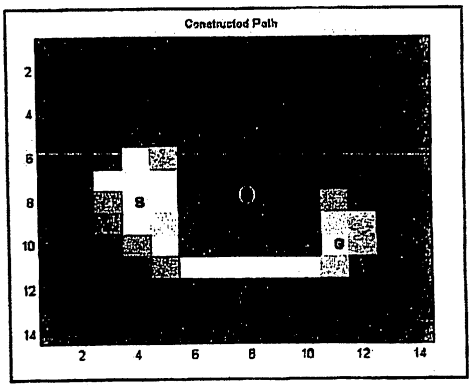

Fig.10b shows constructed path by using asymmetric weight matrix approach for application 3.

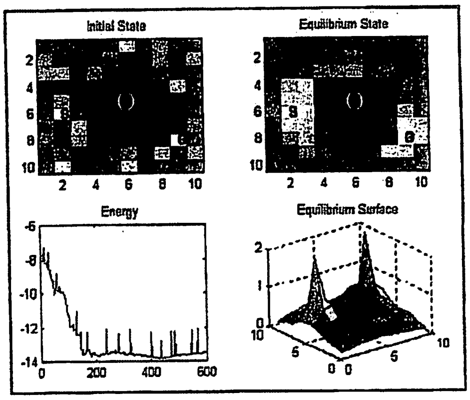

Fig. 11a shows system states and system energy by using asymmetric weight matrix approach for application 3. (600 steps of learning)

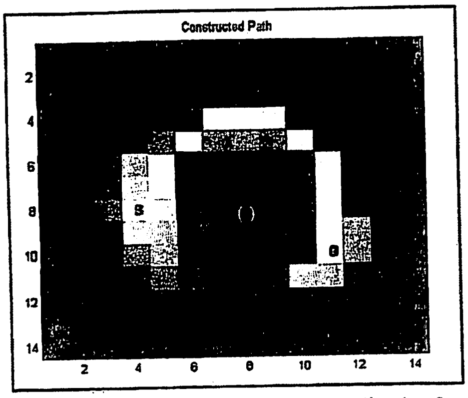

Fig.11b shows constructed path for application 3. (600 steps of learning)

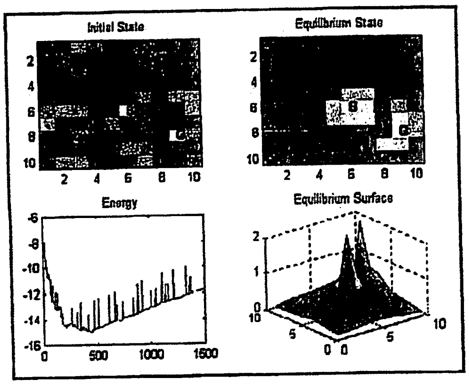

Fig. 12a shows system states and system energy by using asymmetric weight matrix approach for application 4.

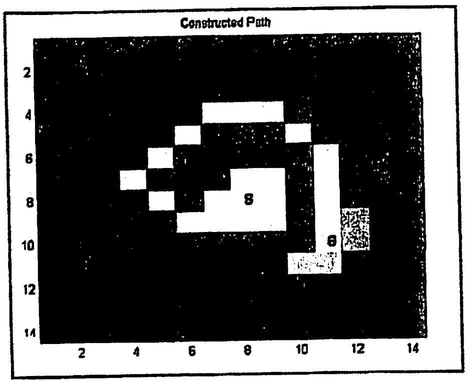

Fig. 12b shows constructed path by using asymmetric weight matrix approach for application 4.

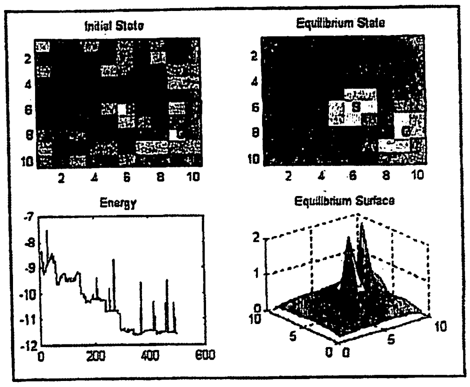

Fig. 13a shows system states and system energy by using asymmetric weight matrix approach for application 4. (500 steps of learning)

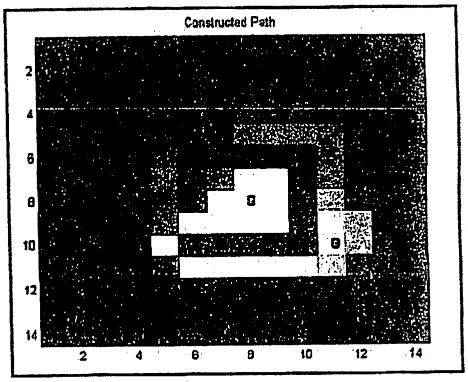

Fig.13b shows constructed path for application 4. (500 steps of learning)

7. Discussion

By comparing to the symmetric method, the simulation results show more effective paths generated by the modified Hopfield network as shown in Figs. 7 and 8. A symmetric property leads to propagation of the activation level around the goal. It is quite difficult to construct a good path in the meaning of less number of stepping. In Figs. 10 and 12, the system energy was decreased toward the minimum point and grown up after long learning. This is a result of asymmetric weight matrix. Thus decreasing of the system energy cannot guarantee. By learning until the minimum of system energy, the path was constructed more effective than long learning as shown in Figs. 11 and 13. Such path can be constructed effectively by leaning until it approachs to the minimum energy point. Learning until it approaches to the Equilibrium state is not necessary for our method. Path can be generated in a shorter time than using the original Hopfield network.

8. Conclusion

In this research, obstacle avoidance for mobile robot is studied. Asymmetric weight matrix has been introduced. Even though the decreasing of the system energy cannot be guaranteed, it can be demonstrated. The modified Hopfield network will be learned until the minimum energy point. Simulation results showed that path generator constructed an effective path. This leads to reducing of path planning time.

10. Reference

[1] Y. Arai, T. Fujii, H. Asama, Y. Kataoka, 'Adaptive Behavior Acquisition of Collision Avoidance among Multiple Autonomous Mobile Robots', Proceeding of IROS, 1997

[2] Mochida, T.; Ishiguro, A.; Aoki, T.; Uchikawa, Y., 'Behavior arbitration for autonomous mobile robots using emotion mechanisms', International Conference on Intelligent Robots and Systems, IROS, Pittsburgh, Pennsylvania, USA,1995

[3] Ishiguro, A., Watanabe, R., Uchikawa, Y., 'An immunological approach to dynamic behavior control for autonomous mobile robots', International Conference on Intelligent Robots and Systems, IROS, Pittsburgh, Pennsylvania, USA, 1995

[4] R. Mehrotra and D. M. Krause, 'Obstacle-free Path Planning for Mobile Robots'

[5] M. G. Lagoudakis, 'Hopfield Neural Network for Dynamic Path Planning and Obstacle Avoidance', Technical Paper, University of Southwestern Louisiana.

[6] S. Haykin, 'Neural Networks: A Comprehensive Foundation', Prentice Hall International, Inc., New Jersy