A Kind of New Multicast Routing Algorithm for Application of Internet of Things

G. Li[1,2,3], D. G. Zhang*[1,2], K. Zheng[1,2], X. C. Ming[1,2], Z. H. Pan[1,2], K. W. Jiang[1,2]

[1]Key Laboratory of Computer Vision and System
(Tianjin University of Technology), Ministry of Education, China
[2]Tianjin Key Lab of Intelligent Computing & Novel Software
Technology, Tianjin University of Technology, China
[3]Daqing Geophysical Exploration Company, CNPC, Daqing,
Heilongjiang Province, 163357, China
* gandegande@126.com

ABSTRACT
Wireless Sensor Networks (WSN) is widely used as an effective medium to integrate physical world and information world of Internet of Things (IOT). While keeping energy consumption at a minimal level, WSN requires reliable communication. Multicasting is a general operation performed by the Base Station, where data is to be transmitted to a set of destination nodes. Generally, the packets are routed in a multi-hop approach, where some intermediate nodes are also used for packet forwarding. This problem can be reduced to the well-known Steiner tree problem, which has proven to be NP-complete for deterministic link descriptors and cost functions. In this paper, we propose a novel multicast protocol, named heuristic algorithms for the solution of the Quality of Service (QoS) constrained multicast routing problem, with incomplete information in Wireless Sensor Networks (WSN). As information aggregation or randomly fluctuating traffic loads, link measures are considered to be random variables. Simulation results show that the Hop Neural Networks (HNN) based heuristics with a properly chosen additive measures can yield to a good solution for this traditionally NP complex problem, when compared to the best multicast algorithms known.

Keywords: Multicast routing, WSN, hop neuron network

1. Introduction

It is well known that wireless sensor networks (WSN) is a self-organization wireless network system constituted by numbers of energy-limited micro sensors under the banner of Internet of Things (IOT). Nowadays, WSN is widely used as an effective medium to integrate physical world and information world of IOT [1-3]. A wireless sensor network consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants [4-6]. Multicast service is an efficient model. It can optimized the network resource and adapt to the bandwidth of wireless sensor network. Thus, the performance of the IOT will be improved [7-9]. The key service of the WSN is Multicast routing for the problem of multicast routing, generally we build a multicast tree with the least cost. And the cost of the Steiner tree is lowest, therefore, Sterner tree is regard as the best method to solve multicast correspondence [10-13]. In this paper, we present a kind of new multicast routing algorithm for application of Internet of things [14-17].

2. The fundamental of Steiner tree

Before discussing the multicast algorithms, we need to introduce some notations [18-23]. A network is represented as a graph N = $(\mathrm{V}_N, \mathrm{E}_N)$, where V_N is the set of nodes and $\mathrm{E}_N \subset \mathrm{V}_N$ x V_N is the set of edges. The average number of edges that depart from a node is referred to as out degree. Over the set of edges we define the two functions delay Delay: $\mathrm{E}_N \rightarrow \mathrm{R}^{+}\backslash\{0\}$ and cost Cost: $\mathrm{E}_N \rightarrow \{1\}$.The delay and the cost of a path are defined as the sum of the delay or cost of all the edges of the path.

The multicast receivers are referred to as multicast group and Q $\in \mathrm{V}_N$ is the source of the multicast group. Computing the Steiner Tree is an NP-complete problem [5-8]. The main contribution of this paper is the method to transform the original probabilistic link descriptors, which reduces the tree selection to a deterministic problem. Then we apply the HNN which then ensures fast convergence to a suboptimal solution.

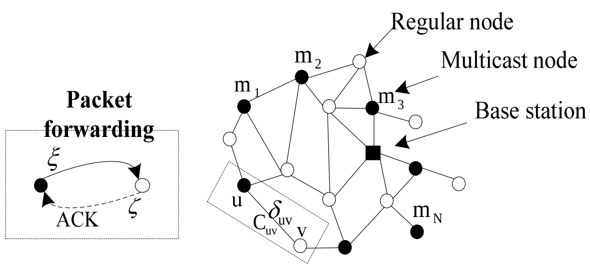

Figure 1. A typical Wireless Sensor Network, with Base Station denoted as a white box, multicast nodes as black boxes and regular nodes as circles.

3. The packet transmission model for multicast routing

We model the network as a graph $G(E, V, \delta_{uv}, (u, v) \epsilon E)$, where E denotes the set of edges, V the set of vertices and δ_{uv} is the descriptor of the link (u, v) ϵ E that can be delay, bandwidth or any other type of metric. The information source, typically a Base Station, is denoted by s ϵ V and the set of multicast nodes by $M = \{m_1, m_2, \cdots, m_N\} \subset V$. We assume that equal transmission power is used by each node in the network

$$P_{uv}^{TX} = g, \quad \forall (u, v) \in E \tag{1}$$

The probability of successfully packet reception is calculated by using the Rayleigh fading model as follows:

$$P(\text{packet reception}) = P_{uv} = \exp\left(-\frac{\Theta \cdot \sigma_z^2 \cdot d_{uv}^{\gamma}}{g}\right) \tag{2}$$

Where Θ is a hardware related threshold, σ_z^2 represents the noise power, d_{uv} denotes the distance between node u and v, and γ is the path loss exponent.

We assume communication scheme where receiver nodes use acknowledgement packets (ACK), and we assume that the number of ACKs are not limited. Table I shows a sequence of packet transmissions between source node u and destination v, where v can either receive or fail to receive the data packet, which cases are denoted by 1 and 0 respectively. The ACK can also be decoded correctly by u or not. The communication ends when both packets are received correctly.

Events	Outcomes					
Reception of data from u to v	1	0	1	···	1	1
Reception of ACK from v to u	0	0	0	···	0	1

Table 1. A sequence of packet reception until both dataand ACK are received.

Let us denote the event of data reception followed by no ACK reception – first column – with ξ and the event of unsuccessful data reception – second column – with ζ. The corresponding distributions can be expressed as follows

$$P(\xi_{uv} = m) = (1 - P_{vu})^m \cdot P_{vu} \tag{3}$$

$$P(\varsigma_{uv} = n) = \frac{P_{uv} \cdot P_{vu} \cdot (1 - P_{uv})^n}{(1 - P_{uv} + P_{uv} \cdot P_{vu})^{n+1}} \tag{4}$$

Let us define χ as the random variable corresponding to the power consumption over the link (u, v) until successful data transmission

$$\chi_{uv} = \xi_{uv} \cdot 2g + \zeta_{uv} \cdot g + 2g \tag{5}$$

The expected value of the transmit power over link (u, v) then is

$$E(\chi_{uv}) = E(\xi_{uv}) \cdot 2g + E(\zeta_{uv}) \cdot g + 2g \tag{6}$$

In this case the distribution of the delay δ_{uv} on link (u, v) can be calculated as follows:

$$P(\delta_{uv} = l) = P_{uv} \cdot P_{vu} \cdot (1 - P_{uv} \cdot P_{vu})^{l-1}, l = 1,2 .. \tag{7}$$

4. Multicast routing with incomplete information

Because of incomplete information about the link states in the network, link metrics $\delta_{(u,v)}$are described by random variables. These link metrics are not additive thus the deterministic multicast tree search algorithms are not applicable. We transform the random link metrics into deterministic descriptors which can be fed to the traditional or heuristic algorithms. In order to do this, we introduce two types of common requirements for designing a multicast tree: a bottleneck type and an end-to-end additive type requirement.

We formulate the deterministic bottleneck Steiner tree problem as

$$\max_{(u,v)\in T} P_{uv} < \alpha, T\epsilon T \tag{8}$$

where T is the set of all trees containing{s}□M, P_{uv} is the transmitter power consumed during transmission, and α is the limit that we do not want to exceed in order to economize battery power.

The deterministic end-to-end additive type problem can be formulated as

$$\begin{aligned} &\underset{T\epsilon T}{\operatorname{argmin}} \sum_{(u,v)\in T} C_{uv}\,, \\ &\text{s.t.} \sum_{(u,v)\in R_{sm}} D_{uv} < \beta, \forall m \in M \end{aligned} \tag{9}$$

Where R_{sm} is a path from s to $m \in M$ in T, and L_{uv} is an additive metric with QoS requirement β.
Although no polynomial algorithm is known for finding Steiner trees but suboptimal algorithms exist. In the next subsections we extend the problem to the case of random link descriptors.

4.1 Bottleneck type requirement

When using random link descriptors, the optimal multicast tree problem for bottleneck measure is defined as follows

$$T_1^*: \underset{T\epsilon T}{\operatorname{argmax}} P\left(\max_{(u,v)\epsilon T} \chi_{uv} < \alpha\right) \tag{10}$$

If χ_{uv}<α holds for the maximal χ_{uv} then it holds for all the values in the tree, which yields

$$P\left(\max_{(u,v)\epsilon T} \chi_{uv} < \alpha\right) = \tag{11}$$

$$P\left(\bigcap_{(u,v)\epsilon T} \{\chi_{uv} < \alpha\}\right) = \tag{12}$$

$$\prod_{(u,v)\epsilon T} P(\chi_{uv} < \alpha) = \tag{13}$$

$$\sum_{(u,v)\in T} \log[P(\chi_{uv} < \alpha)] \tag{14}$$

Where we used the independence property of these random variables.
This results in the following objective function over additive measures:

$$T_1^*: \underset{T\epsilon T}{\operatorname{argmax}} \sum_{(u,v)\in T} -\log[P(\chi_{uv} < \alpha)] \tag{15}$$

Which is well suited for the later introduced HNN.

4.2 End-to-end additive requirement

In the traditional multicast setting with incomplete information we search for a tree such that the probability that the delay of the longest route being smaller than β is maximal

$$\underset{T\epsilon T}{\operatorname{argmax}} P\left(\max_{R_{sm}\epsilon R_T} \sum_{(u,v)\in R_{sm}} \delta_{uv} < \beta\right) \tag{16}$$

The WSN setting requires us to also minimize the transmit power used by the network in order to prolong node life span.

To incorporate this additional constraint we define the following optimization problem, and by increasing the κ parameter in our algorithm we converge to the optimal tree:

$$\begin{aligned} &T_2^*: \underset{T\epsilon T}{\operatorname{argmin}} \sum_{(u,v)\in T} C_{uv}\,, \\ &\text{s.t.}\ P\left(\max_{R_{sm}\epsilon R_T} \sum_{(u,v)\in R_{sm}} \delta_{uv} > \beta\right) < \kappa \end{aligned} \tag{17}$$

In the optimal Steiner tree there are common links between paths for different multicast nodes, meaning that the random measures for different paths statistically dependent, which can be described by the joint distribution function.

We can use large deviation theory to approximate the previous probability

$$\mathrm{P}\left(\max_{\mathrm{R_{sm}} \epsilon \mathrm{R_T}} \sum_{(\mathrm{u,v}) \in \mathrm{R_{sm}}} \delta_{\mathrm{uv}} > \beta\right) \leq \exp(\mu\mathrm{R_{sm}}(\sigma) - \sigma \cdot \beta) \tag{18}$$

for all σ, where

$$\mu\mathrm{R_{sm}}(\sigma) = \sum_{(\mathrm{u,v}) \in \mathrm{R_{sm}}} \mu_{\mathrm{uv}}(\sigma) \tag{19}$$

and

$$\mu_{\mathrm{uv}}(\sigma) = \ln\{\mathrm{E}(\exp(\sigma \cdot \delta_{\mathrm{uv}}))\} \tag{20}$$

The optimal σ value can be calculated by solving

$$\sigma_{\mathrm{opt}}: \sum\nolimits_{(\mathrm{u,v}) \in \mathrm{R_{sm_j}}} \frac{\mathrm{d}}{\mathrm{d}\sigma} \mu_{\mathrm{uv}}(\sigma) = \beta \tag{21}$$

for a given route from s to $\mathrm{m_j}$。

We are looking for the tree approximated by Equation (18) and our suboptimal solution will be the one with the minimal $\sum_{(\mathrm{u,v}) \in \mathrm{T}} \mathrm{C_{uv}}$ for which

$$\exp(\mu\mathrm{R_{sm}}(\sigma) - \sigma \cdot \beta) < \kappa \quad \leftrightarrow \tag{22}$$

$$\mu\mathrm{R_{sm}}(\sigma) < \log(\kappa) + \sigma \cdot \beta \tag{23}$$

and translates Equation (17) into

$$\mathrm{T_2^*}: \operatorname*{argmin}_{\mathrm{T} \epsilon \mathrm{T}} \sum\nolimits_{(\mathrm{u,v}) \in \mathrm{T}} \mathrm{C_{uv}}\,, \qquad \mathrm{s.t.} \;\; \mu\mathrm{R_{sm}}(\sigma) < \log(\kappa) + \sigma \cdot \beta \tag{24}$$

which is in the form of the well-known Constrained Steiner Tree5, for which a Discrete Hop Neural Network (DHNN) based approximation is given in Section 5.

For a given β parameter then we can find better and better trees by increasing κ in an iterative or gradient fashion, which yields Algorithm 1.

Algorithm 1 find optimal tree for end-to-end requirement

Require: G, κ=0, β>1

Repeat

T = find tree with HNN (G,κ,β)

If T is found then

Decrease κ

Else

Increase κ

End if

Until no significant increase in performance

Figure 2 illustrates a case, when there are two sets of trees having equal longest path delay properties – from each set we choose based on the sum of transmit powers –. Assume that it is required that the probability of the longest route exceeding 6 is to be maximized. At the first step of the algorithm both of the routes can guarantee a longest delay of 6 with probability κ_1, we can decrease κ. In the second step this holds again while in the third step there is only one tree below delay 6. This way we can determine the minimal value for κ for which there exists at least one tree.

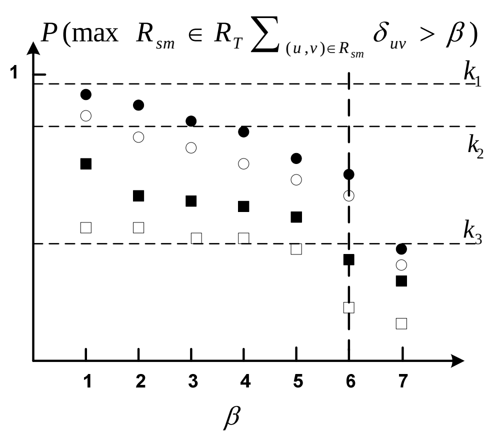

Figure 2. The probability of longest route exceeding threshold β for two trees ● and ■.

5. Solution by hop net

In this section we first describe the structure of the HNN that is capable of approximating the multicast tree for deterministic link measures.

Hop Networks in general have successfully been applied to combinatorial optimizations and solved many practical tasks 7-10. So this problem's solution can be approximated by HNN5, 11-12 as well, based on the energy function proposed in 5, 11. We use DHNN, because it is reported to be computationally more effective 13-15. The energy function is a weighted linear combination of terms which are describing the objective function to be minimized (E_1^m) and the press of the constraint function subjected by the minimization task (E_5^m). The feasibility of the solution is guaranteed by the neuron update selection rule, which ensures transitions to only valid candidate solutions. The HNN searches for routes and for a tree solution we use the union of the chosen routes. Thus we implicitly assume that the union of routes satisfying the constraints is a good Steiner tree. Every neuron represents an edge in the graph 5 and the neuron's output variable is noted by V_{uv}^m which is defined as

$$V_{uv}^m = \begin{cases} 1, \text{if the link } (u,v) \text{ is chosen into } R_{sm} \\ 0 \text{ otherwise} \end{cases} \quad (25)$$

Cost and constraint terms: The cost value for the edge (u, v) ∈ R_{sm} is noted by C_{uv}.

$$E_1^m(V) = \sum_{u=1} \sum_{\substack{v=1 \\ u \neq v \\ (u,v)\neq(m,s)}} \frac{C_{uv}V_{uv}^m}{1+\sum^n_{\substack{j=1 \\ j\neq m, j\in D}} V_{uv}^j} \quad (26)$$

Note that the cost term is dependent on the edges previously elected in the multicast tree, so edge are reuse are preferred.

E_5^m is the term which presses the constraint function to be true. L_{uv} > 0 is the delay value on the link (u, v).

$$\sum_{u=1} \sum_{\substack{v=1 \\ u \neq v \\ (u,v)\neq(m,s)}} L_{uv}V_{uv}^m \leq \Delta \quad (27)$$

We use the same approach as [5] for dealing with inequality constraints; introduce a linear programming type neuron.

$$h(z) = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}, z = \sum_{u=1} \sum_{\substack{v=1 \\ u \neq v \\ (u,v)\neq(m,s)}} L_{uv}V_{uv}^m - \Delta \quad (28)$$

$$E_5^m = H(z) = \int h(z)dz = \begin{cases} z^2/2 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

Neuron selection rule for DHNN: We initialize the network, that there is only one edge in the chosen path: V_{uv} = 1, if (u, v) = (s, m) 0otherwise, □(u, v) □ R_{sm}. We update exactly 3 neurons for one discrete time step as follows: Selection A (edge splitting): choose an edge which is in the path (u, v) □ R_{sm}, choose two edges which are not in the path, but join at a common node and start and terminate at u and v. (u, w), (w, v) /□ R_{sm}. Selection B (edge joining): choose two edges which are in the path as (u, w), (w, v) □ R_{sm}, thus (u, v) /□R_{sm}.

Either A or B used, update the three neurons (flip V_{uv}, V_{uw}, V_{wv} triangle) if the state transition yields to a better energy state of the network. Use A and B alternatively until no state transition occurred. This selection rule ensures that if we started from a valid route we end up in a valid route from s to m. Cost and constraint terms for DHNN: The conventional energy function of the discrete Hop model is

$$-\varepsilon_{DTHNN}(y) = y^T W_y + 2y^T b, y \in \{-1,+1\}^K \quad (30)$$

We transform the neurons output variable to a column vector with elements of {−1, +1} for the DHNN structure. We treatV_{uv}^m an element of a series of matrices: from the m^{th} matrix the element at the u^{th} row and the v^{th} column. We have N nodes in the graph and we read out Vm column wise, u, v = 1, . . . , N

$$v^m: v_k^m = V_{uv}^m = (y_k^m)/2 \quad (31)$$

$$y^m: y_k^m = 2v_k^m - 1 \quad (32)$$

$$u = \left\lfloor \frac{k-1}{N} + 1 \right\rfloor, v = \left((k-1) \bmod N\right) + 1 \tag{33}$$

We define and $\sum_{(u,v)\neq(m,s)}$ for the elimination of the terms "u =v" and" (u, v) = (m, s)" respectively by the multiplication with these matrices.

$$\sum_{u=1}^{n} \sum_{\substack{v=1 \\ u\neq v}}^{n} V_{uv}^{m} = v^{m} \cdot \sum_{u\neq v} \tag{34}$$

$$\sum_{u=1}^{n} \sum_{\substack{v=1 \\ (u,v)\neq(m,s)}}^{n} V_{uv}^{m} = v^{m} \cdot \sum_{(u,v)\neq(m,s)} \tag{35}$$

For the term E_1^m after the transformation the parameters of Equation (30) will be

$$W^{m,1} = 0 \tag{36}$$

$$b^{m,1+} = 1/2\, b^{m,1} \cdot \textstyle\sum_{u\neq v} \cdot \sum_{(u,v)\neq(m,s)}$$
$$b_k^{m,1+} = \frac{C_{uv}}{1 + \sum_{j\neq m, j\in D}^{n} V_{uv}^{j}} \tag{37}$$

Similarly for E_5^m the calculation can be done thus getting the parameters $W^{m,i}$, $b^{m,i}$. The energy function of the HNN as in (30) is

$$-\varepsilon(y) = y^T(u_i W^{m,1})y + 2y^T(u_i b^{m,1}) + H(y^T u_5 W^{m,5} y + 2u_5 y^T b^{m,5}) \tag{38}$$

6. Performance analysis

The T1 and the T2 objective functions were evaluated by exhaustive search and the HNN based algorithm on a graph with the following parameters [8-13]: The size of the network N = 8, the Rayleigh channel parameters were chosen to typical or better indoor environment: γ = 3, g = 1, θ = 10, σ2 = 1. The positions of the nodes were randomly generated according to i.i.d. uniform distributions in the unit square. The group of the multicast nodes consisted 3 randomly chosen nodes. We have performed the exhaustive search by enumerating all the possible trees and evaluating the objective function on the trees. We have compared the results of the HNN algorithm to the exhaustive solution. For the T2 objective function. We have evaluated the performance given by the Chertoff bound and also the corresponding theoretical probability by performing convolutions on the known distributions.

The HNN algorithm can find almost always the optimal solution for the T1 objective function of the bottleneck problem [11-17]. This figure is typical in the sense that throughout the simulation runs we have seen the same behavior. For the T2 objective function the figures show the probability of meeting the delay constraint and the found tree's energy consumption [18-23].

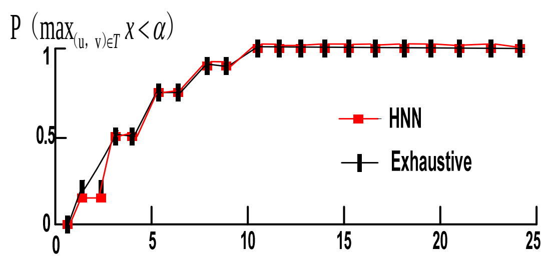

Figure 3. A typical evaluation of the T1 obj. function.

It can be seen in Figure 3 that the HNN algorithm can find almost always the optimal solution for the T1 objective function of the bottleneck problem. This figure is typical in the sense that throughout the simulation runs we have seen the same behavior [14-17].

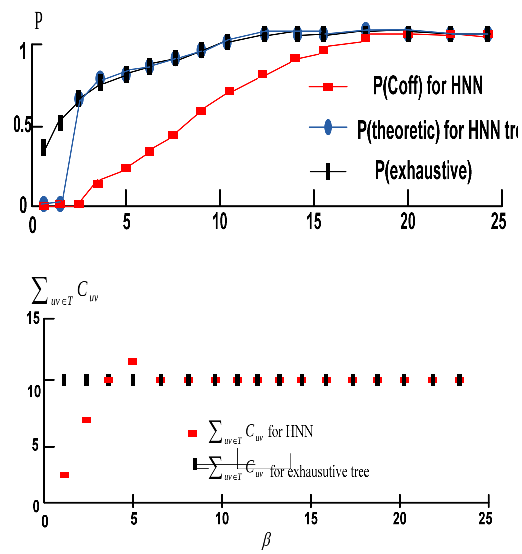

Figure 4. Near optimal solution for T2 obj. function.

In Figure 4 a case can be seen at β = 4 that the HNN finds a solution that satisfies the delay constraint with a higher probability in the expense of larger transmit power [18-20].

It can be seen in Figure 5 for small β values it can happen that individual link measures approximated by the Chernoff bound could not give a positive probability of meeting the delay constraint; hence the HNN could not supply a valid tree [21-23].

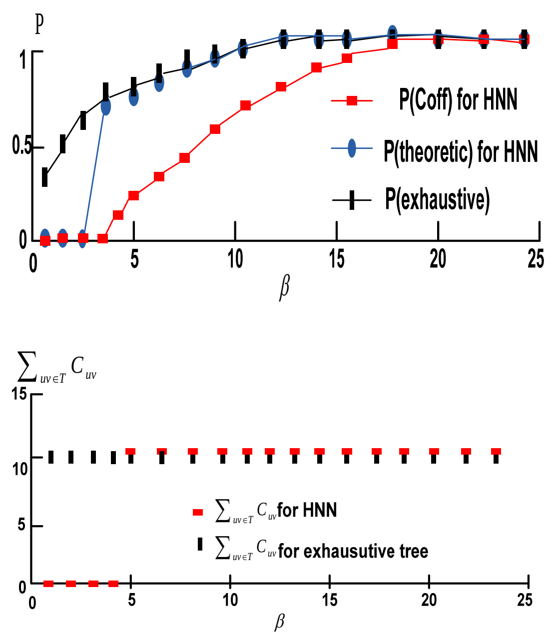

Figure 5. A typical evaluation of the T2 obj. function.

7. Conclusion

WSN is widely used as an effective medium to integrate physical world and information world of IOT. While keeping energy consumption at a minimal level, WSN requires reliable communication. For networks sizes as small as 10 nodes exhaustive search is infeasible so heuristics are needed to approximate a good solution. We have shown that a HNN based heuristics with a properly chosen additive measures can yield to a good solution for this traditionally NP complex problem. Because of the conservativeness of the Chernoff approximation the delay bound is always met in the expense of consuming more transmit power.

Acknowledgments

This work is supported by "863" project plan of China (No.2007AA01Z188), National Natural Science Foundation of China (No.61170173 & No.60773073 & No.61202169 & No.61001174), Program for New Century Excellent Talents in University of China (No.NCET-09-0895), Key project of Ministry of Education of China (No.208010) Tianjin Natural Science Foundation (No. 10JCYBJC00500),Tianjin Key Natural Science Foundation (No. 13JCZDJC34600).

References

[1] A. Gamil, "Neural networks for solving quadratic assignment problems," Neural Information Processing - Letters and Reviews, vol. 10, no.3, 51-57,2006.

[2] W. Liu and L. Wang, "Solving the shortest path routing problem using noisy hopfield neural networks," in Proceedings of the Communications and Mobile Computing, 2009,1:56-67.

[3] J. J. Hopfield and D. W. Tank, "neural computation of decisions in optimization problems," Biological Cybernetics, vol. 52, 141-152, 1985.

[4] M. Qiao and D. G. Zhang, "Efficiently matching frequent patterns based on bitmap inverted files built from closed itemsets," International Journal on Artificial Intelligence Tools, 21(3):1250011, 2012.

[5] S. Basagni et al, "Location aware, dependable multicast for mobile ad hoc networks," Computer Networks, 659- 670, 2001.

[6] K. Chen and K. Nahrstedt, "Effective Location- Guided Tree Construction Algorithms for Small Group Multicast in MANET," in Proc. INFOCOM, 2002, 1192- 1201.

[7] D. Tank and J. Hopfield, "Simple neural optimization networks: Ana/d converter, signal decision circuit, and a linear programming circuit," Circuits and Systems, IEEE Transactions on, vol. 33, no. 5, 533-541,1986.

[8] X. D. Zhang and D. G. Zhang, "A Kind of Differentiator Design Algorithm Based on IQP," INFORMATION - An International Interdisciplinary Journal. 15(1):7-14, 2012.

[9] I. K. SOHRAB et al, "Protocols for Self-Organization of a Wireless Sensor Network," IEEE Personal Communications, 16-27, 2000.

[10] D. G. Zhang, "A new approach and system for attentive mobile learning based on seamless migration," Applied Intelligence,36(1):75-89, 2012.

[11] K. A. Smith, “Neural networks for combinatorial optimization: a review of more than a decade of research,” INFORMS J. on Computing, vol. 11, no. 1, 15-34,1999.

[12] D. G. Zhang and X. J. Kang, “A new method of non-line wavelet shrinkage denoising based on spherical coordinates,” INFORMATION - An International interdisciplinary Journal, 15(1):141-148, 2012.

[13] D. G. Zhang and X. D. Zhang, “A new service-aware computing approach for mobile application with uncertainty,” Applied Mathematics & Information Sciences, 6(1): 9-21, 2012.

[14] D. G. Zhang and Y. N. Zhu, “A New Method of Constructing Topology Based on Local-World Weighted Networks for WSN,” Computers and Mathematics with Applications, 64(5): 1044–1055, 2012.

[15] D. G. Zhang and X. J. Kang, “A novel image de-noising method based on spherical coordinates system,” EURASIP Journal on Advances in Signal Processing, 2012,1:110, 2012.

[16] D. G. Zhang and C. P. Zhao, “A new medium access control protocol based on perceived data reliability and spatial correlation in wireless sensor network,” Computers and Electrical Engineering, 38(3): 694-702, 2012.

[17] D. G. Zhang and X. D. Zhang, “Design and implementation of embedded un-interruptible power supply system (EUPSS) for web-based mobile application,” Enterprise Information Systems, 6(4): 473- 489, 2012.

[18] O. S. Joshua and I. S. Oluwarotimi, “Mathematical Modeling of Leachate Production from Waste Contained Site,” International Journal of Engineering and Technology Innovation, 2(3):195-206,2012.

[19] H. J. Kwak and G. T. Park, “Study on the Mobility of Service Robots,” International Journal of Engineering and Technology Innovation, 2(2): 97-112, 2012.

[20] M. Henzel and K. Falkowski, “The analysis of the control system for the bearingless induction electric motor,” Journal of Vibroengineering, 14(1):711, 2012.

[21] A. Mystkowski, “An application of mu-synthesis for control of a small air vehicle and simulation results,” Journal of Vibroengineering, 14(1):721, 2012.

[22] J. F. Hsieh, “Design and analysis of offset slider-crank with translating roller-follower,” Transactions of the Canadian Society for Mechanical Engineering, 35(3):419-436, 2011.

[23] S. Ozturk, “Slip-line modeling of machining and determine the influence of rake angle on the cutting force,” Transactions of the Canadian Society for Mechanical Engineering, 36(1):23-35, 2012.