

基于最优能耗多播树构造的 Ad hoc 网络节点路由算法研究

李 渊 杨立波

(太原大学计算机工程系 太原 030009)

摘 要 针对 Ad hoc 网络中最小能耗多播树的生成和优化问题,提出了基于最优能耗多播树构造的 Ad hoc 网络节点路由算法。在该算法中,首先将最小能耗多播树生成问题转化为不同中继节点集合幂空间中的动态寻优问题,构建基于最优能耗多播树求解模型;然后利用改进的粒子群算法对不同维度空间上代表中继节点链路的粒子的权值进行映射和修正计算,再依据粒子适应度值对粒子的局部极值和全局极值进行更新;最后根据粒子位置和速度更新机制进行迭代计算,将最终的全局极值点和极值作为最优多播树的节点位置和能耗值。实验仿真证明,该算法具有较好的粒子多样性,全局搜索和局部搜索能力较好,并且优化能力较强。

关键词 Ad hoc 网络,多播树,粒子群,最优能耗

中图分类号 F257 **文献标识码** B

Ad hoc Network Node Routing Algorithm Based on the Optimal Energy Consumption Multicast Tree Structure

LI Yuan YANG Li-bo

(Department of Computer, Taiyuan University, Taiyuan 030009, China)

Abstract In order to solve Ad hoc least energy consumption multicast tree generation and optimization problems, this paper put forward Ad hoc network node routing algorithm based on the optimal energy consumption multicast tree structure. In this algorithm, the minimum cost multicast tree generation problem is first transformed into different relay node set power space of the dynamic optimization problem, and solving model based on the optimal energy consumption multicast tree is built. And the improved particle swarm optimization (pso) algorithm is used to make mapping and correction calculation for the right value of particle representing relay point link in different dimension space and then according to the particle fitness value, the particle's local extremum and global extremum are updated. According to the particle position and velocity update mechanism of the iterative calculation, eventually global extreme value point and extreme value are used as the most optimal multicast tree node position and energy consumption value. The simulation results show that this algorithm has better particle diversity and global search and local search ability are good, and the optimization ability is strong.

Keywords Ad hoc working group on the network, Multicast tree, Particle swarm, Optimal energy consumption

1 引言

Ad hoc 网络节点的能量有限,一旦耗尽,对整个网络系统都有严重的影响,因此如何针对网络节点进行最小能耗多播树的构造是该领域研究的重要课题。文献[1]提出在全向天线模式下对最小能耗广播树进行构造的 BIP(Broadcast Incremental Power)算法和最小能耗多播树 MIP(Multicast Incremental Power)算法, MIP 算法是基于 BIP 算法广播树进行修剪生成的多播树,但是不同中继节点对多播树生成的计算误差对该算法影响较大;文献[2]提出了利用模拟退火算法对最小能耗广播树进行构造,利用 BIP 算法随机最优问题的初始解,计算结果优于 BIP 算法,但其仅对中小网络系统求解较好;文献[3]提出用蚁群算法对多播树进行构造,并对构造树利用 r-shrink 算法^[4]进行优化,算法的优化效果较好,但是执行时间较长;文献[5]提出了利用粒子群优化算法求解带权

无向图,通过中继节点寻优来生成 Steiner 树,但是该算法在 Ad hoc 网络节点最优能耗路径问题上适应度较低。针对以上研究,本文提出了一种最优能耗多播树构造算法,即在对粒子群算法进行粒距聚类度和粒子信息熵计算的基础上,对粒子的权值进行了修正,提升了粒子的局部寻优和全局寻优能力。在全向天线模式和有向天线模式下对最优能耗多播树的构造进行了仿真实验,证明了本文算法的优化性能。

2 Ad hoc 网络节点路由模型

在静态的 Ad hoc 网络中,节点位置已知,且每个节点持有信息收发设备,可动态地调整各自节点的能量,本文算法模型仅考虑节点的能量发送,不考虑节点对能量的处理。本文的节点路由模型包括天线模型和无线传播模型。在天线模式下,若节点进行均匀的路由转播,则节点每次的转播能量按照 $r^{-\infty}$ 进行衰减, r 为节点间的距离, ∞ 为媒介参数,值域为[2,

到稿日期:2012-06-24 返修日期:2012-09-11 本文受国家自然科学基金(J1526987435)资助。

李 渊(1980—),男,硕士,讲师,主要研究方向为计算机网络, E-mail: liyuanty@163.com; 杨立波(1979—),男,硕士,副教授,主要研究方向为计算机网络、人工智能。

4]. 当节点 i 的传输能量 $p_i \geq r_{ij}^\infty$ 时, 节点 j 可以接收到节点 i 的数据信息, 其中 r_{ij} 为节点间的距离, β_j 为节点接收数据信息的能量门限. 节点 i 数据传输信息的范围由节点自身的能量决定, 设节点的能量均匀地分布在天线覆盖范围内, 则节点的传输能量 $p_i \geq \frac{\theta_i * r_{ij}}{360}$ 时, 在节点 i 的天线覆盖范围内的节点 j 可以成功地接收到节点 i 发送的数据信息, 其中 θ_i 为节点 i 的天线波束宽度. 节点 i 的数据传输范围同时受节点 i 的天线波束宽度的影响, 则节点 i 的数据传输能量按照均匀传输时, $p_i = \frac{\max(\theta_i, \theta_{\min}) * r_i^2}{360}$, 其中 θ_{\min} 为天线的最小波束宽度. 由于有向天线的波束宽度较窄, 因此有向天线在进行节点信息传输时能耗较低, 同等能耗下信息传输的距离也越远.

设 Ad hoc 网络中节点的传输能量分布已确定, 则节点间数据信息的传输可以看作有向图模型 $G=(V, E)$, 其中 V 为网络系统中的节点集合, E 为节点的链接路径. 节点的能量在特定范围内进行调节, 当节点传输距离发生变化时, 节点间的网络路径被修改或删除, 节点的拓扑图也会发生变化. 对于特定的网络系统 G , 数据信息的发送节点 s 和数据信息的接收节点集 D , 网络系统中的其他节点都可以当作参与节点信息转播的中间节点, 本文的研究问题就是在这个网络系统中如何建立最优多播树 T , T 的根节点为 s , 叶子节点 $l \in D$, T 中的节点除了源节点 s 和接收节点 D 外均为中继节点.

在全向天线模型中, T 中节点 i 的数据传输能量 $p_i = \max r_{ij}^\infty$, 即在最大传输路径范围内, 在有向天线模型中, T 中节点 i 的数据传输能量 p_i 满足 $p_i = \frac{\max(\theta_i, \theta_{\min}) * \max r_{ij}^\infty}{360}$, 即受最大传输路径和波束宽度的影响. T 树的传输能耗为各节点的能耗总和, 即 $p(T) = \sum_{i \in V_T} p_i$.

设网络系统 G 中的节点位置已确定, 各节点的数据传输能量已知, 能量上限为 p_{\max} , 则最小能耗的多播树可看作: 在网络系统 G 中, 从数据传输源节点 s 到接收节点 D 中, 寻找以源节点 s 为起始到达目的接收节点 D , 且满足 $p \leq p_{\max}$ 的最优能耗树 T .

3 改进粒子群优化算法

3.1 粒子群算法

粒子群算法是粒子在空间维度中以特定的速度进行飞行, 动态地对粒子的位置进行更新, 在维度方向上搜索粒子当前环境中的最优解和当前群体的最优解. 设粒子群体由 M 个粒子组成, 维度空间为 D , 则第 i 个粒子在第 d 维空间中的位置为: $X_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$, 粒子的速度为: $V_i = \{v_{i1}, v_{i2}, \dots, v_{id}\}$, 个体的最优位置为: $P_i = \{p_{i1}, p_{i2}, \dots, p_{id}\}$, 种群的全局最优位置为: $P_g = \{p_{g1}, p_{g2}, \dots, p_{gd}\}$, 则粒子 i 在第 $t+1$ 时刻的速度和位置更新为:

$$\begin{aligned} v_{id}^{t+1} &= v_{id}^t + c_1 * r_1 (p_{id}^t - x_{id}^t) + c_2 * r_2 (p_{gd}^t - x_{id}^t) \\ x_{id}^{t+1} &= x_{id}^t + v_{id}^{t+1} \end{aligned}$$

由于粒子群算法中粒子都以最优解进行当前位置的迭代更新, 因此粒子的整体变化趋向相同, 这易造成粒子的多样性较低, 从而引起粒子群后期迭代计算的收敛速度较慢, 难以在有限的时间和迭代次数内获取较好的优化效果. 文献[6]分析了将惯性权重系数引入粒子群算法, 并对粒子群算法的全局寻优和局部寻优进行调节. 则粒子的速度更新为:

$$v_{id}^{t+1} = \omega * v_{id}^t + c_1 * r_1 (p_{id}^t - x_{id}^t) + c_2 * r_2 (p_{gd}^t - x_{id}^t)$$

式中, $\{c_1, c_2\}$ 是加速常数, $\{r_1, r_2\}$ 的值域为 $[0, 1]$, ω 为惯性权重系数. 粒子群优化算法在寻优过程中主要包括了当前粒子对邻近粒子的信任判定, 依据当前粒子运行趋势进行惯性运动决策, 同时粒子要对自身运行所经历的历史位置进行权衡, 以决定历史运行轨迹对当前粒子运行趋势的影响, 最后粒子在优化过程中要考虑粒子间的搜索能力. 基于此, 对惯性权重系数 ω 的权值映射和粒子间的相关关系进行研究.

文献[7]对 LDIW 策略分析方法进行了研究. 设 $\Delta\omega = \omega_{\max} - \omega_{\min}$, 则:

$$\omega(t) = \omega_{\max} - \Delta\omega * \frac{t}{t_{\max}}$$

式中, ω_{\max} 为最大权值, ω_{\min} 为最小权值, t 为当前粒子计算的次数, t_{\max} 为粒子迭代总次数.

3.2 改进的粒子群算法

在粒子群惯性权重系数赋值时, 若其映射函数为线性递减, 则算法的极值点未必为真实动态系统的极值点, 较大的 ω 会影响粒子速度 v_{ij} 使其偏离当前环境的极值点, 从而有利于快速进入局部极值搜索, 使整个算法的极值更优. 但是当 ω 惯性权重系数在算法执行到一定阶段时, 粒子群粒子的多样性降低, 这势必会导致粒子较难寻找到全局极值的收敛点. 文献[8]提出了粒距, 设粒子群在第 t 次迭代计算时第 j 个粒子在第 i 维空间中的粒距为 $d_{ij}(t)$, 则:

$$d_{ij}(t) = |x_{ij}(t) - g_{best_i}(t)|$$

根据更新速度的大小对该次计算中粒子的粒距进行排序, 形成该次迭代环境下粒子粒距的速度序列, 然后计算粒子当前速度在粒距序列中的概率. 设当前环境中粒距序列对为 $X(d_m)$, 当前粒子速度 v_i 属于第 n 个粒距的概率为:

$$p_{ij} = \frac{|n|}{|m|}$$

式中, $|n|$ 为当前粒子速度在粒距中计算的最接近粒距对, 其约束条件为:

$$0 < |v_i - \overline{v_{i1}}| \ll X(d_m)$$

且 $\overline{v_{i1}} = \frac{1}{n} \sum_{i=1}^n v_{mji}$, 表示第 m 个粒距中 i 个粒子的第 j 维速度均值. $|m|$ 为当前计算环境中总的粒距数目, 则粒子 i 在当前迭代计算中的信息熵为:

$$H(v_i) = - \sum_{j=1}^n p_{ij} \ln p_{ij}$$

设在当前的 m 个依据粒子速度排序的粒距对中, 随机地在每个粒距中提取一个粒子速度, 则 $\overline{v_1} = \frac{1}{m} \sum_{i=1}^m v_{im}$.

设 $0 < |\overline{v_1} - k * |\overline{v_1}|| \ll X(d_m)$, $k=1, 2, \dots, n$, 其中 $|\overline{v_1}|$ 为粒距的中位数速度均值, 则满足约束条件的粒距为粒距对中的第 n' 个粒距, 粒子在粒距中的平均信息熵为:

$$\overline{H(v_1)} = - \sum_{j=1}^n p'_{ij} \ln p'_{ij} = - \sum_{j=1}^n \frac{|n'|}{|m|} \ln \frac{|n'|}{|m|}$$

故对于任意迭代计算的粒子惯性权重函数, 有:

$$\begin{cases} H(v_i) > \overline{H(v_1)}, \omega = -(\omega_{start} - \omega_{end}) \left(\frac{t}{t_{\max}}\right)^2 + \omega_{start} \\ H(v_i) \leq \overline{H(v_1)}, \omega = (\omega_{start} - \omega_{end}) \left(\frac{t}{t_{\max}}\right)^2 + (\omega_{end} - \omega_{start}) \left(\frac{2t}{t_{\max}}\right) + \omega_{start} \end{cases}$$

式中, $\omega_{start} = 0.95$, $\omega_{end} = 0.05$, $t_{\max} = 1000$.

在实际的粒距计算时, 由于粒子速度的聚集特性, 会导致

粒子按照速度分类时产生紧集合,当粒距较小时,种群易陷入局部极值;而距离较大时,算法收敛较难。因此在进行粒距划分后,应对粒子的聚类程度进行记录,当粒子的聚类度较高时,粒距间距离较小,此时应增大惯性系数;当粒子聚类度较低时,降低惯性系数,以提高算法的自适应收敛。

$$\text{设} \begin{cases} d_{\text{mean}}(t) = \frac{[\sum_{j=1}^n \sum_{i=1}^n d_{ij}(t)]}{(n * d)} \\ d_{\text{max}}(t) = |\max[d_{ij}(t)]|, k = \frac{[d_{\text{max}}(t) - \overline{d}(t)]}{d_{\text{max}}(t)} \end{cases}$$

式中, $d_{\text{mean}}(t)$ 是粒子种群的平均粒距, $d_{\text{max}}(t)$ 为最大粒距, k 为当前粒子状态的聚类度,取值为 $[0, 1]$, 则粒子权重计算时,其惯性系数应调整为:

$$\begin{cases} w(t) = w * w_{\text{start}}, & k \geq \alpha \\ w(t) = w * \frac{1}{w_{\text{end}}}, & k < \beta \end{cases}$$

式中, $\{\alpha, \beta\}$ 为惯性系数调整参数。

4 最优能耗多播树构造算法

MIP(D-MIP)算法^[9]利用 DIP(D-BIP)算法首先构造一棵广播树,然后通过修剪该树生成多播树,其算法主要利用了无线传输的 WMA 特征。在该算法中,选取不同的中继节点对算法的多播树生成精度影响较大,因此通过对网络系统中不同中继节点模式下的最优树进行粒子群算法寻优,来降低该算法的计算误差。设集合 R 为除源节点和目的接收节点外的所有中继节点的集合, R 的幂集就是粒子的解空间,则粒子群算法中的每个粒子位置就代表了 R 中的一个中继节点集,粒子 i 的位置 x_i 的第 j 维表示该维空间中响应的中继节点进行多播树的生成计算,粒子的维数为中继节点的个数。则最优能耗多播树的构造算法为:

Step1 初始化粒子

随机地在 $x_{ij} \in (0, 1)$ 范围内对粒子位置进行赋值,在 $|v_{ij}| \leq v_{\text{max}}$ 范围内对粒子速度进行赋值,令粒子个体极值 $p_{\text{best}_i} \leftarrow -\infty$, 全局极值 $g \leftarrow -1$, 迭代计算次数为 duration 。

Step2 粒子权重赋值

对粒子进行粒距聚类,依据粒子权重映射函数进行粒子权重系数的计算。

Step3 粒子权重系数修正

对粒子信息熵和粒距聚集度进行计算,对粒子的权值进行修正。

Step4 粒子适应度值

利用 MIP(D-MIP)算法计算粒子的适应度值 $f(x_i)$ 。

Step5 粒子个体极值更新

若 $f(x_i) < p_{\text{best}_i}$, 则 $p_{\text{best}_i} = f(x_i)$, 将该粒子的适应度值赋值给个体极值。

Step6 粒子全局极值更新

若 $p_{\text{best}_g} < p_{\text{best}_i}$, 则 $p_{\text{best}_g} = p_{\text{best}_i}$, 将当前粒子的最优极值赋值给全局极值。

Step7 粒子更新

依据粒子速度和位置更新公式对粒子进行更新。

Step8 迭代计算

依据迭代计算条件判定是否终止计算,输出全局极值点的位置和极值。

5 实验仿真与分析

实验环境为一台 IntelCorei3-5301G 内存 Windows XP 系统的 PC 机,算法运行平台为 Java6.0 的 MyEclipse 8.5。为了对该算法的寻优能力进行验证,采取文献[8]中的二进制编码的多样性测度方法对粒子群多样性进行度量,则 $\text{div} = \frac{\sum_{i=1}^{n-1} \sum_{n=i+1}^n h_{ij}}{\frac{1}{2}mn(n-1)}$, 其中 m 为粒子维度, n 为粒子总数, h_{ij} 为粒子 i 到粒子 j 的距离。以 MIP 算法的适应度对本文算法^[10]与 DPSO 算法^[11]的粒子多样性进行对比。随机地生成具有 50 个节点的网络系统,节点分布区域为 1000m^2 , 目的接收节点为 16, 源节点和目的节点集随机生成,节点的最大能量不限,粒子规模为 30, 迭代次数为 100, 权值域为 $[0.4, 1.0]$, 两种算法均运行 20 次,取其均值作为多样性度量,则具体的算法粒子多样性变化如图 1 所示。

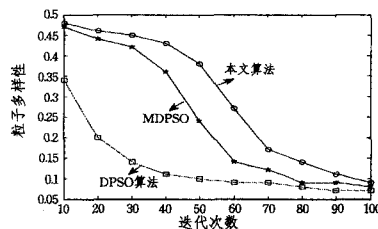


图 1 3 种算法的粒子多样性变化情况图

从图 1 可以看出, DPSO 算法的粒子多样性降低较快, 算法收敛速度较快, 导致算法的全局搜索能力较差, 较早地进入局部最优搜索。MDPSO 算法在搜索前期粒子的多样性较高, 全局搜索能力有所提高, 而本文算法通过对粒子信息熵和粒距聚集度进行计算, 并对粒子权重进行了修正, 因此粒子的全局搜索和局部搜索性能较好^[12]。

为了验证该算法的节点能耗优化能力, 在全向天线模式下用 MIP 算法对粒子适应度值进行计算, 在有向天线模式下用 D-MIP 算法计算粒子适应度值, 每种算法运行 20 次, 取其平均值作为度量值。实验结果如表 1 所列。

表 1 3 种算法不同节点数时的优化能力比较表

节点数	算法	全向天线	有向天线
20	DPSO	291999.18	74938.95
	MDPSO	291999.18	74938.95
	本文算法	291998.16	74936.86
40	DPSO	218518.27	67467.94
	MDPSO	217837.1	67393.49
	本文算法	216923.3	67129.82
60	DPSO	395439.1	113418.64
	MDPSO	391425.39	112762.8
	本文算法	387924.21	111924.9
80	DPSO	325292.47	86179.52
	MDPSO	323976.18	86024.2
	本文算法	321991.83	85928.23
100	DPSO	290459.28	78805.17
	MDPSO	290188.45	78678.13
	本文算法	289923.21	78392.87

从表 1 可以看出, 本文算法的优化性能要优于 MDPSO 算法和 DPSO 算法。

为了对该算法构造最优能耗多播树的有效性进行验证, 选取不同的能耗门限在全向天线模式和有向天线模式下进行仿真实验, 对 30 个不同的网络系统各进行 30 次实验, 取其平

均值进行度量。网络系统节点数为 50,分布区域为 1000m²,迭代次数为 30,权值域为[0, 6, 1. 0],粒子数在每 10 次仿真中递减,分别取 30、25、20。实验数据如表 2、表 3 所列。

表 2 3 种算法的计算效果对比表

节点最大能量	MIP 算法		MDPSO 算法		本文算法	
	平均能耗	平均时耗 /ms	平均能耗	平均时耗 /s	平均能耗	平均时耗 /s
250	392095.16	0.45	357327.15	0.21	324875.03	0.18
300	400769.08	0.3	362011.04	0.22	339823.71	0.19
350	393845.72	0.37	359141.87	0.22	328172.23	0.19
400	398793.62	0.42	361928.57	0.22	338232.87	0.19
450	401212.08	0.42	364782.39	0.22	339723.25	0.19
500	409938.02	0.53	372381.89	0.22	349233.27	0.19
600	412003.21	0.32	367321.72	0.22	329874.34	0.19
700	405344.73	0.43	367327.78	0.22	338764.83	0.19
800	396901.75	0.41	360517.97	0.22	338794.23	0.19
1000	403447.58	0.48	367786.46	0.22	339973.82	0.19

表 3 3 种算法的计算效果对比表

节点最大能量	MIP 算法		MDPSO 算法		本文算法	
	平均能耗	平均时耗 /ms	平均能耗	平均时耗 /s	平均能耗	平均时耗 /s
200	108684.62	4.01	100558.43	1.60	99776.82	1.21
250	110694.67	3.48	101867.10	1.60	99983.74	1.22
300	108565.93	3.72	99793.46	1.61	98293.22	1.22
350	108959.22	3.82	101256.82	1.61	99939.29	1.22
400	110649.82	3.77	102124.34	1.61	99873.98	1.22
450	109466.72	3.92	100520.07	1.60	99923.87	1.21
550	108923.65	3.76	100882.76	1.61	99843.45	1.22
650	108466.23	3.46	99903.29	1.61	98329.32	1.22
750	107687.62	3.68	99035.85	1.61	98231.28	1.21
950	105827.73	3.89	102360.27	1.62	100217.62	1.22

表 2 为全向天线模式下 3 种算法在不同能耗门限时的算法执行效率;表 3 为在有向天线模式下,最小波束宽度为 90 度时 3 种算法在不同能耗门限下的算法执行效率。实验证明,在不同能耗门限下,本文算法具有较好的最优能耗多播树构造效率。

结束语 Ad hoc 网络中最优节点路由路径构造是节点能耗优化的重要课题。本文通过对全向天线和有向天线模式下节点能耗模型进行分析,针对特定网络系统中最优能耗多播树的动态生成问题进行研究,提出了基于最优能耗多播树构造的 Ad hoc 网络节点路由算法。其通过对粒子群算法基于粒距聚类度和粒子信息熵计算的改进,提升了最优能耗多播树生成的搜索能力。实验仿真表明,该算法能够有效地降低中继节点选取对最优能耗多播树构造的影响,提升了算法的鲁棒性,降低了算法的执行时间和能耗。

参考文献

- [1] Wieselthier J E, Nguyen G D, Ephremides A. On the construction of energy-efficient broadcast and multicast trees in wireless networks [C]//Proceedings of IEEE INFOCOM'. Tel Aviv, Israel, 2000; 585-594
- [2] Montemanni R, Gambardella L M, Das A K. The minimum power broadcast problem in wireless networks; a simulated annealing approach [C]// Proceedings of the 2005 IEEE Wireless Communications and Networking Conference. New Orleans, LA, USA, 2005; 2057-2062
- [3] Hernandez H, Blum C. Energy-efficient multicasting in wireless ad-hoc networks; an ant colony optimization approach [C]//Proceedings of the 2008 IEEE International Symposium on Wireless Communication Systems. Reykjavik, Iceland, 2008; 667-671
- [4] Das A K, Marks R J, El-Sharkawi M, et al. R-shrink; a heuristic for improving minimum power broadcast trees in wireless networks [C]//Proceedings of IEEE GLOBECOM'03. San Francisco, A, USA, 2003; 523-527
- [5] Zhong W L, Huang J, Zhang J. A novel particle swarm optimization for the Steiner tree problem in graphs [C]//Proceedings of the 2008 IEEE Congress on Evolutionary Computation. Hong Kong, China, 2008; 2460-2467
- [6] 黄泽霞,俞攸红,黄德才. 惯性权自适应调整的量子粒子群优化算法[J]. 上海交通大学学报, 2012, 2: 228-232
- [7] EBERHART R. Fuzzy adaptive particle swarm optimization [C]//Proceedings of 2001 IEEE International Conference on Evolutionary Computation. San Francisco; IEEE Press, 2001; 101-106
- [8] 阳春华,谷丽珊,桂卫华. 自适应变异的粒子群优化算法[J]. 计算机工程, 2008, 16: 188-190
- [9] Wieselthier J E, Nguyen G D, Ephremides A. Energy-aware wireless networking with directional antennas; the case of session-based broadcasting and multicasting [J]. IEEE Transactions on Mobile Computing, 2002, 1(3): 176-191
- [10] 朱晓建,沈军. 基于粒子群优化的 ad hoc 网络最小能耗多播路由算法[J]. 通信学报, 2012, 33(3): 52-58
- [11] Kennedy J, Eberhart R. A discrete binary version of the particle swarm algorithm [C]//Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Orlando, FL, USA, 1997; 4104-4108
- [12] 袁辉勇,阙清贤,羊四清. 传感器网络中基于能耗均衡的节点优化部署[J]. 计算机仿真, 2010, 27(8): 100-102
- [13] 杨春德,邓超. 一种动态的时延约束费用优化多播路由算法[J]. 重庆邮电大学学报:自然科学版, 2011, 23(1): 96-100
- [14] Di Stefano A, Morana G, Zito D. A P2P strategy for QoS discovery and SLA negotiation in Grid environment [J]. Future Generation Computer Systems, 2009, 25(8): 862-875
- [15] Zhou J, Dou W. A QoS-Aware Service Selection Approach on P2P Network for Dynamic Cross-Organizational Workflow Development [C]//Proc. of the International Conference on Web Information Systems and Mining (WISM 2009). Shanghai: Springer-Verlag Berlin, 2009; 289-298
- [16] 郭得科,任彦,陈洪辉,等. 一种 QoS 有保障的 Web 服务分布式发现模型[J]. 软件学报, 2006, 17(11): 2324-2334
- [17] 刘志忠,王怀民,周斌. 一种双层 P2P 结构的语义服务发现模型 [J]. 软件学报, 2007, 18(8): 1922-1932
- [18] 吴万明,吴毅坚,赵文耘. 基于 Chord 网的语义 Web Service 发现 [J]. 电子学报, 2007, 35(B12): 152-155
- [19] and Ubiquitous Computing, 2009, 13(7): 471-477
- [10] He Q, Yan J, Yang Y, et al. Chord4S; a P2P-based decentralised service discovery approach [C]//Proc. of the 2008 IEEE International Conference on Services Computing (SCC 2008). Salt Lake City, 2008; 221-228
- [11] Sioutas S, Sakkopoulos E, Makris C, et al. Dynamic Web Service discovery architecture based on a novel peer based overlay network [J]. Journal of Systems and Software, 2009, 82(5): 809-824
- [12] Skoutas D, Sacharidis D, Kantere V, et al. Efficient Semantic Web Service Discovery in Centralized and P2P Environments [C]//Proc. of the 7th International Semantic Web Conference (ISWC 2008). Karlsruhe; Springer Berlin, 2008; 583-598
- [13] Zhang Y, Huang H, Yang D, et al. Bring QoS to P2P-based semantic service discovery for the Universal Network [J]. Personal

(上接第 82 页)