

基于粒子群优化的 ad hoc 网络最小能耗多播路由算法

朱晓建^{1,2}, 沈军^{1,2}

(1.东南大学 计算机科学与工程学院, 江苏 南京 211189; 2.东南大学 计算机网络和信息集成教育部重点实验室, 江苏 南京 211189)

摘 要: 在无线 ad hoc 网络中, 设备使用电池提供能量, 随着多播应用日益广泛, 如何构造最小能耗多播树是一个重要问题。针对选择不同的中继节点集对构造最小能耗多播树的影响, 提出了一种优化最小能耗多播树构造的离散粒子群算法。为了避免离散粒子群算法早熟收敛, 引入惯性权重策略, 以平衡算法的全局搜索能力和局部搜索能力。模拟实验结果表明改进后的离散粒子群算法具有较强的优化能力, 有效地优化了最小能耗多播树的构造。

关键词: 无线 ad hoc 网络; 多播路由; 最小能耗; 粒子群优化

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2012)03-0052-07

Minimum energy consumption multicast routing in ad hoc networks based on particle swarm optimization

ZHU Xiao-jian^{1,2}, SHEN Jun^{1,2}

(1. School of Computer Science and Engineering, Southeast University, Nanjing 211189, China;

2. Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing 211189, China)

Abstract: In wireless ad hoc networks, because devices are powered by batteries, and multicast applications are constantly emerging, how to construct a multicast tree of the minimum energy consumption is an important problem. For the effect of the different choices of relay nodes to the construction of the minimum energy consumption multicast tree, a discrete particle swarm optimization algorithm to optimize the construction of the minimum energy consumption multicast tree was proposed. In order to avoid the premature convergence of the discrete particle swarm optimization algorithm, an inertia weight strategy was introduced to balance the global search ability and the local searching ability. The results of simulated experiments show that the modified discrete particle swarm optimization algorithm has strong optimization ability, and can effectively optimize the construction of the minimum energy consumption multicast tree.

Key words: wireless ad hoc network; multicast routing; minimum energy consumption; particle swarm optimization

1 引言

无线 ad hoc 网络由一组无线设备组成, 在不需要使用网络基础设施的情况下, 可以实现快速临时组网, 具有广泛的应用。在无线 ad hoc 网络中, 节点使用电池提供能量, 而电池的能量十分有限, 一旦电池的能量耗尽, 节点将不能继续工作, 网络将

不再连通。因此, 在无线 ad hoc 网络中, 节能是一个核心问题, 针对越来越多的多播应用, 如何构造最小能耗多播树是一个重要问题。

文献[1]指出在无线 ad hoc 网络中无论使用全向天线还是使用有向天线, 构造最小能耗多播树问题是一个 NP 难解问题, 因此需要设计有效的启发式算法以求得较好的近似最优解。文献[2]提出了在使用

收稿日期: 2010-09-29; 修回日期: 2011-04-29

基金项目: 国家自然科学基金资助项目(61070159); 国家重点基础研究发展计划(“973”计划)基金资助项目(2009CB320501)

Foundation Items: The National Natural Science Foundation of China (61070159); The National Basic Research Program of China (973 Program) (2009CB320501)

全向天线的情况下构造最小能耗广播树的BIP(broadcast incremental power)算法和构造最小能耗多播树的MIP(multicast incremental power)算法, MIP算法首先利用BIP算法构造一棵广播树, 然后对该广播树进行修剪得到多播树, 未考虑不同的中继节点选择对构造多播树的影响。文献[3]提出了在使用有向天线的情况下构造最小能耗广播树的D-BIP(directional BIP)算法和构造最小能耗多播树的D-MIP(directional MIP)算法, 与MIP算法基于BIP算法类似, D-MIP算法是基于D-BIP算法的。文献[4]提出了一种优化最小能耗广播树的r-shrink算法, 该算法首先基于其他算法构造一棵广播树, 然后以降低树的总能耗为目标, 对树的结构进行调整。文献[5]提出了一种构造最小能耗广播树的模拟退火算法, 利用BIP算法构造初始解, 求解质量显著优于BIP算法, 但仅对中小规模网络的最小能耗广播树问题求解效果较好。文献[6]提出了一个构造最小能耗多播树的蚁群算法, 并在构造过程中使用r-shrink算法优化多播树的构造, 优化效果较好, 但运行时间较长。文献[7]提出了一种构造最小能耗多播树的PSOR算法, 对到达非多播组成员的传输建立惩罚函数, 在多播树的构造过程中基于收缩重叠传输范围改变多播树的结构, 以减少多播能耗, 并给出了理论上的最大近似比, 该算法的时间复杂度较高。这些算法大多是直接针对最小能耗广播树的构造, 很少是直接针对最小能耗多播树的构造, 并且不同算法的适用情况不同。文献[8]提出了一种新型离散粒子群优化算法求解带权无向图的Steiner树问题, 通过搜索最优中继节点来获取Steiner树, 但由于无线传输本身所固有的多播特性使得无线网络的最小能耗多播树问题不同于Steiner树问题。文献[9]首先对最小能耗多播树问题进行整数线性规划, 然后运用一种多相离散粒子群算法分布式地计算该整数线性规划的最优解, 引入共生机制处理约束, 该方法计算复杂度较高。

针对不同的中继节点选择对构造最小能耗多播树的影响, 本文提出了一种改进的离散粒子群算法, 引入惯性权重策略以平衡离散粒子群算法的全局搜索能力和局部搜索能力, 以优化在使用全向天线和有向天线情况下的最小能耗多播树的构造, 并通过相关的模拟实验验证了改进后的离散粒子群算法的优化能力, 以及优化最小能耗多播树构造的有效性。

2 问题模型

考虑静态无线ad hoc网络由若干个无线节点构成, 节点的位置已知。节点配有多个收发器可以同时支持多个多播会话, 并且节点可以动态地调整自身的发送能量, 本文只考虑节点的发送能量, 不考虑接收能量和处理能量。

采用与文献[1~3]相似的天线模型和无线传播模型。在使用全向天线的情况下, 假设均匀传播, 信号能量按照 $r^{-\alpha}$ 衰减, r 是到信号源的距离, α 为通信媒介参数, 其值取决于通信介质, 通常介于 2 和 4 之间, 本文假设 $\alpha=2$ 。当节点 i 的传输能量 $p_i \geq r_{ij}^\alpha$ 时, 节点 j 可以成功接收到节点 i 发送的信号, r_{ij} 为节点 i 和节点 j 之间的距离, β_j 为节点 j 的接收能量门限。假设所有节点的接收能量门限都相同且归一化为 1。节点 i 的传输范围由节点 i 的传输距离 r_i 所决定, 节点 i 的传输能量 $p_i = r_i^\alpha$ 。在使用有向天线的情况下, 假设节点的传输能量均匀分布在天线波束内, 当节点 i 的传输能量 $p_i \geq \beta_j / 360 \times r_{ij}^\alpha$ 时, 位于节点 i 的天线波束内的节点 j 可以成功接收到节点 i 发送的信号, β_i 为节点 i 的波束宽度。节点 i 的传输范围不仅取决于节点 i 的传输距离 r_i , 还取决于节点 i 的波束宽度 β_i , 节点 i 的传输能量 $p_i = \max\{\beta_i, \beta_{\min}\} / 360 \times r_i^\alpha$, 其中 β_{\min} 为最小波束宽度。和全向天线相比, 由于有向天线的波束较窄, 使用有向天线使得在给定传输距离时可以节省传输能量, 或者在给定传输能量时可以延长传输距离。位于节点 i 传输范围内的节点都可以接收到节点 i 所发送的信号, 即无线传输本身具有多播特性(WMA, wireless multicast advantage), WMA 特性使得无线多播不同于有线多播。

假设网络中各个节点的传输能量均确定, 可以将网络模型化为一个有向图 $G=(V, E)$, V 表示网络节点集合, 边集 E 定义为: 对于 V 中的任意节点 $i, j, \langle i, j \rangle \in E$ 当且仅当节点 j 位于节点 i 的传输范围之内。由于节点的传输能量 p 可以在一定的范围内进行调节 ($p \leq p_{\max}$, p_{\max} 为节点的最大传输能量), 在调节节点传输能量的同时, 节点的传输距离发生变化, 相关的网络链路被添加或移除, 网络拓扑结构也相应发生变化。对于给定的网络 G , 以及多播会话指定的发送节点 s 、接收节点集 $D, D=\{d_1, d_2, \dots, d_g\}$, 网络中除了发送节点 s 和目标节点 D 之外的其他节点都视为可参与多播会话的中继节点, 本文研

究的是如何建立基于源端的多播树 $T(V_T, E_T)$, T 的根节点是 s , T 的任意叶子节点 $l \in D$, T 中除了源节点 s 和目标节点 D 之外, 其余节点为中继节点。

在使用全向天线的情况下, T 中任一节点 i 的传输能量 p_i 取决于它的最大链路传输能量, 即 $p_i = \max_{\langle i, j \rangle \in E_T} \{r_{ij}^a\}$; 在使用有向天线的情况下, T 中任一节点 i 的传输能量 p_i 取决于其最大链路传输能量和其波束宽度, 即 $p_i = \max_{\langle i, j \rangle \in E_T} \{r_{ij}^a\} / 360 \times \max_{\langle i, j \rangle \in E_T} \{r_{ij}^a\}$ 。

树 T 的总能耗 $p(T)$ 为各个节点的能耗之和, 即 $p(T) = \sum_{i \in V_T} p_i$ 。

假设网络中各节点的位置相对固定, 每个节点的最大传输能量均为 p_{\max} , p_{\max} 应足够大以使得多播会话得以进行。求解最小能耗多播树问题可表示为: 对于给定的网络 G , 以及多播会话指定的源节点 s 和目标节点集 D , 寻找一棵以源节点 s 为根并且可以到达所有目标节点 D 的树 T , 在满足节点传输能量 $p \leq p_{\max}$ 的条件下为节点分配合适的传输能量, 使得树的总能耗 $p(T)$ 最小。

3 改进的离散粒子群算法

粒子群优化 (PSO, particle swarm optimization)^[10,11] 算法是一种利用粒子群体进行随机搜索的智能优化算法。PSO 由一群粒子组成, 每个粒子代表问题的一个潜在解, 具有一个随机速度, 飞行于解空间执行随机搜索。搜索过程从问题的一个解集开始。PSO 模拟社会行为, 每个粒子记录自身所经历的最好位置, 并且学习群体所经历的最好位置。粒子在飞行过程中追踪自身和群体迄今为止所经历的最好位置, 并保持惯性, 飞行速度具有随机性, 以尽力搜索到全局最优解。PSO 具有简单高效, 并行性好、收敛快等优点。

假设在 m 维解空间中, 执行搜索的粒子群由 n 个粒子组成, 向量 $X_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$ 表示第 i 个粒子的位置, 向量 $V_i = (v_{i1}, v_{i2}, \dots, v_{im})^T$ 表示第 i 个粒子的速度。粒子在解空间执行搜索的过程中, 向量 $P_i = (p_{i1}, p_{i2}, \dots, p_{im})^T$ 表示第 i 个粒子自身所经历的最好位置, P_i 对应的适应度值为个体极值 $pbest_i$ 。每个粒子共享整个群体迄今为止所搜索到的最优解, 全局极值索引 g 为所有粒子中个体极值最优的那个粒子的索引, 则向量 $P_g = (p_{g1}, p_{g2}, \dots, p_{gm})^T$ 表示所有粒

子所经历的最好位置, P_g 对应的适应度值 $pbest_g$ 为全局极值。在连续粒子群算法中粒子的速度和位置按下式更新:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 \times rand_1() \times (p_{ij}(t) - x_{ij}(t)) + c_2 \times rand_2() \times (p_{gj}(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

其中, c_1 、 c_2 表示加速因子, 通常取值为 2; $rand_1()$ 、 $rand_2()$ 为均匀分布在区间 [0,1] 上的随机函数; 粒子速度被限制在某个范围之内, 即 $|v_{ij}| \leq v_{\max}$, v_{\max} 表示粒子运动速度的最大值。

在离散粒子群优化 (DPSO, discrete particle swarm optimization) 算法^[12] 中, 粒子位置 X_i 的每一维 x_{ij} 取值仅限于 0 和 1, 粒子速度 V_i 的每一维 v_{ij} 表示 x_{ij} 取值为 1 的可能性, v_{\max} 通常取值为 6.0。粒子速度仍按式(1)更新, 粒子位置按下式更新:

$$\begin{cases} s(v_{ij}(t+1)) = \frac{1}{1 + e^{-v_{ij}(t+1)}} \\ \text{if } (rand() < s(v_{ij}(t+1))) \\ \text{then } x_{ij}(t+1) = 1 \\ \text{else } x_{ij}(t+1) = 0 \end{cases} \quad (3)$$

其中, $s(v_{ij}(t+1))$ 为 sigmoid 函数, $rand()$ 产生服从在区间 [0,1] 上均匀分布的随机数。

为了提高算法获取最优解的能力, 通常在计算前期进行有效的全局搜索, 以定位最优解的大致位置, 在计算后期进行局部搜索, 以获得精确度较高解^[13~15], 然而 DPSO 算法不能有效地平衡全局搜索能力和局部搜索能力。令 $v = v_{ij}(t+1) - v_{ij}(t) = c_1 rand_1() (p_{ij} - x_{ij}) + c_2 rand_2() (p_{gj} - x_{ij})$, 当 $p_{ij} = p_{gj} = 1$, $x_{ij} = 0$ 时, $v > 0$, v_{ij} 增大, $s(v_{ij})$ 增大, $x_{ij} = 1$ 的概率增大; 当 $p_{ij} = p_{gj} = 0$, $x_{ij} = 1$ 时, $v < 0$, v_{ij} 减小, $s(v_{ij})$ 减小, $x_{ij} = 0$ 的概率增大; 当 $x_{ij} = p_{ij} = p_{gj}$ 时, $v = 0$, v_{ij} 不变, $s(v_{ij})$ 不变, x_{ij} 以原有概率保持不变; 当 $p_{ij} > p_{gj}$ 时, x_{ij} 趋于 p_{ij} 或 p_{gj} 。因此, 在 DPSO 算法中, 粒子追寻极值点的能力较强, 算法的局部搜索能力较强, 但算法容易早熟收敛, 全局搜索能力较差。

在连续粒子群算法中, 通常使用惯性权重 w 来平衡算法的全局搜索能力和局部搜索能力^[13,14], w 通常初值为 0.9, 随着迭代的进行而线性下降至 0.4^[13~15]。然而这种逐渐递减的惯性权重不能直接适用于 DPSO 算法。如果 DPSO 算法使用惯性权重 w , 则 $v_{ij}(t+1) = wv_{ij}(t) + c_1 rand_1() (p_{ij} - x_{ij}) + c_2 rand_2() (p_{gj} - x_{ij})$, $v = v_{ij}(t+1) - v_{ij}(t) = (w - 1)v_{ij}(t) + c_1 rand_1() (p_{ij} - x_{ij}) + c_2 r$

$\text{and}_2() \times (p_{gj} - x_{ij}), c_1 \text{rand}_1() \times (p_{ij} - x_{ij}) + c_2 \text{rand}_2() \times (p_{gj} - x_{ij})$ 使粒子具有局部搜索能力, $(w-1)v_{ij}(t)$ 具有随机性和无记忆性, 使粒子具有扩大搜索空间的趋势、探索新区域的能力, 从而使粒子具有全局搜索能力。当 $0 \leq w \leq 1$ 时, 随着 w 的减小, $|(w-1)v_{ij}(t)|$ 逐渐增大, 全局搜索能力逐渐增强, 反之, w 越大, $|(w-1)v_{ij}(t)|$ 越小, 局部搜索能力越强。因此, 逐渐递减的惯性权重不适用于 DPSO 算法。

为了克服 DPSO 算法易于早熟收敛、全局搜索能力较差的问题, 在 DPSO 算法中引入一种逐渐递增的惯性权重, 以有效地平衡算法的全局搜索能力和局部搜索能力, 提高算法获取全局最优解的能力。在计算前期, 使惯性权重 w 由 $w_a (w_a \in [0,1])$ 逐渐增大至 $w_b (w_b \in [0,1], w_b > w_a)$, 从而使算法具有较强的全局搜索能力; 在计算后期, 使惯性权重 w 保持为 w_b 不变, 从而使算法具有较强的局部搜索能力, 以进行精细的局部搜索。在改进后的 DPSO 算法(MDPSO, modified DPSO)中, 粒子速度按式(4)更新, 惯性权重按式(5)更新:

$$v_{ij}(t+1) = w \times v_{ij}(t) + c_1 \times \text{rand}_1() \times (p_{ij}(t) - x_{ij}(t)) + c_2 \times \text{rand}_2() \times (p_{gj}(t) - x_{ij}(t)) \quad (4)$$

$$w = \begin{cases} w_a + (w_b - w_a) \times t / t_d, & t \leq t_d \\ w_b, & t > t_d \end{cases} \quad (5)$$

其中, t_d 为增强全局搜索能力的时期, w_a 、 w_b 、 t_d 需根据具体情况进行设定。

4 构造最小能耗多播树的粒子群算法

MIP(D-MIP)算法^[2,3]是基于 BIP(D-BIP)算法的, 利用了无线传输的 WMA 特性, 首先由 BIP(D-BIP)算法构造一棵广播树, 然后修剪该广播树来获得多播树, 其步骤如下。

Step1 初始化多播树 $T(V_T, E_T)$ 只包含源节点 s , $V_T = \{s\}$, $E_T = \emptyset$, 初始化 V 为网络中所有节点的集合。

Step2 对于任意一对节点 i 和节点 j , $i \in V_T$, $j \in V - V_T$, 计算将 j 作为 i 的孩子节点后节点 i 的能耗 p_{ij} , 如果 $p_{ij} \leq p_{\max}$, 计算节点 i 所增加的能耗 $\Delta_{ij} = p_{ij} - p_i$, 其中, p_i 为 j 成为 i 的孩子之前节点 i 的能耗。

Step3 如果对于任意一对节点 i 和节点 j , $i \in V_T$, $j \in V - V_T$, $p_{ij} > p_{\max}$, 返回计算失败。

Step4 找到最小的 Δ_{ij} 对应的节点 i 和节点 j ,

将节点 j 加入 V_T , 将边 $\langle i, j \rangle$ 加入 E_T 。

Step5 如果 $V_T = V$, 则转 Step2。

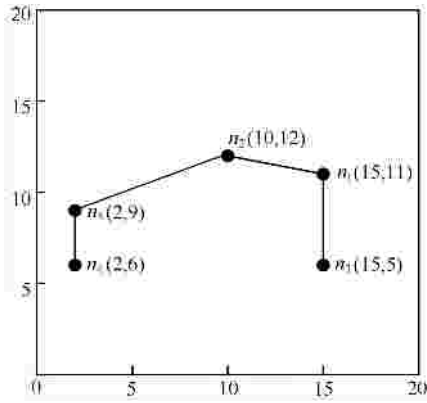
Step6 对 T 进行修剪, 剪除对于到达目标节点所不需要的边, 即如果节点及其下游节点中不包含目标节点, 就将其排除。

Step7 输出多播树 T 及其总能耗, 返回计算成功。

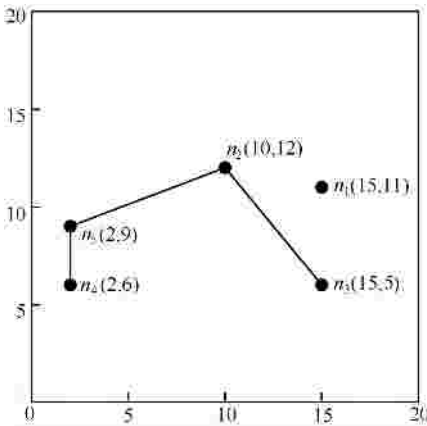
可见, MIP(D-MIP)算法在构造多播树时, 将网络中除了源节点和目标节点之外的其他所有节点都作为中继节点参与多播树的构造, 未考虑对参与多播树构造的中继节点进行筛选, 导致算法结果误差较大。如果节点的最大传输能量 p_{\max} 较小, 会导致网络无法连通, 算法将返回计算失败。

图 1 所示的一个实例演示了 2 种不同的中继节点选择对 MIP 算法计算结果的影响。在该实例中, 网络共有 5 个节点, 各节点的坐标分别为 $n_1(15,11)$ 、 $n_2(10,12)$ 、 $n_3(15,5)$ 、 $n_4(2,6)$ 、 $n_5(2,9)$, 其中, n_2 为源节点, n_3 、 n_4 为目标节点, 节点的最大传输能量无限制即 $p_{\max}=8$ 。如果使用除了源节点和目标节点之外的其他所有节点(包括 n_1 、 n_5)作为中继节点参与多播树的构造, 使用 MIP 算法构造多播树: 初始时 $V_T = \{n_2\}$ 、 $V = \{n_1, n_2, n_3, n_4, n_5\}$, 首先选择距离源节点 n_2 最近的节点 n_1 加入多播树得到 $V_T = \{n_2, n_1\}$, 然后根据最小化能耗增加原则, 依次选择节点 n_3 、 n_5 、 n_4 加入多播树, 结果如图 1(a)所示, 节点 n_2 的能耗 $p_2 = \max\{r_{21}^2, r_{23}^2\} = 73$, 节点 n_5 的能耗 $p_5 = r_{54}^2 = 9$, 节点 n_1 的能耗 $p_1 = r_{13}^2 = 36$, 多播树的总能耗为 $p_2 + p_5 + p_1 = 118$ 。如果仅使用节点 n_5 作为中继节点参与多播树的构造, 使用 MIP 算法构造多播树: 初始时 $V_T = \{n_2\}$ 、 $V = \{n_2, n_3, n_4, n_5\}$, 首先选择距离源节点 n_2 最近的节点, 由于节点 n_1 未参与多播树的构造, 所以选择节点 n_5 加入多播树得到 $V_T = \{n_2, n_5\}$, 然后根据最小化能耗增加原则依次选择节点 n_3 、 n_4 加入多播树, 结果如图 1(b)所示, 节点 n_2 的能耗 $p_2 = \max\{r_{23}^2, r_{25}^2\} = 74$, 节点 n_5 的能耗 $p_5 = r_{54}^2 = 9$, 多播树的总能耗为 $p_2 + p_5 = 83$ 。

由此可见, 选择不同的中继节点集对 MIP 算法的计算结果影响较大, 可以通过比较 MIP 算法对不同中继节点集求得的结果, 来获取一个最优中继节点集, 从而降低 MIP 算法的计算误差。假设集合 $R = \{u_1, u_2, \dots, u_m\}$ 表示网络中除去源节点和目标节点之外其他所有节点的集合, 则 R 的幂集就表示该问题的解空间, 由于 MDPSO 算法是对问题解空间的一次



(a) 选择 n_1 、 n_3 作为中继节点参与最小能耗多播树的构造



(b) 选择 n_3 作为中继节点参与最小能耗多播树的构造

图 1 不同中继节点选择对 MIP 构造最小能耗多播树的影响

全局搜索过程,因而可以使用 MDPSO 算法求解最优中继节点集,从而优化最小能耗多播树的构造。在 MDPSO 算法中,每个粒子的位置代表一个中继节点集,粒子 i 位置 X_i 的第 j 维 $x_{ij}=1$ 表示该维对应的中继节点 $u_j(u_j \in R)$ 参与多播树的构造, $x_{ij}=0$ 表示 u_j 不参与多播树的构造,粒子的维数为网络中除去源节点和目标节点之外其他所有节点的个数,即为 $|R|$ 。在 MDPSO 算法中,在使用全向天线的情况下,使用 MIP 算法计算粒子的适应度值,在使用有向天线的情况下,使用 D-MIP 算法计算粒子的适应度值,即在选择了特定中继节点集的基础上,使用 MIP 或 D-MIP 算法构造多播树,并计算树的总能耗。对于选定的某个中继节点集,由于 p_{\max} 的限制,由中继节点以及多播会话指定的源节点和目标节点构成的子网络不一定能连通,因而该中继节点集不一定是可行解。求解最小能耗多播树的 MDPSO 算法如下。

Step1 随机初始化粒子位置 X_1, X_2, \dots, X_n , $x_{ij} \in \{0,1\}$; 随机初始化粒子速度 V_1, V_2, \dots, V_n , $|v_{ij}| \leq v_{\max}$; 初始化粒子个体极值 $pbest_1? 8$,

$pbest_2? 8, \dots, pbest_n? 8$; 初始化全局极值索引 $g? 1$; 初始化总迭代次数 $duration$; 初始化迭代次数变量 $t? 0$; 初始化粒子标识变量 $i? 1$ 。

Step2 如果 $t=duration$, 则转 Step12。

Step3 按式(5)更新惯性权重 w 。

Step4 如果 $i>n$, 则转 Step11。

Step5 使用 MIP(D-MIP)算法计算第 i 个粒子的适应度值 $f(X_i)$, 如果计算失败, 转 Step8。

Step6 如果 $f(X_i) < pbest_i$, 那么更新个体极值点 $P_i? X_i$, 更新个体极值 $pbest_i? f(X_i)$ 。

Step7 如果 $pbest_i < pbest_g$, 那么更新全局极值索引 $g? i$ 。

Step8 按式(4)更新粒子的速度 V_i 。

Step9 按式(3)更新粒子的位置 X_i 。

Step10 $i? i+1$, 转 Step4。

Step11 $t? t+1, i? 1$, 转 Step2。

Step12 输出全局极值点 P_g , 输出全局极值 $pbest_g$ 。

5 实验结果

为了验证 MDPSO 算法协调全局搜索能力和局部搜索能力的有效性, 采用文献[16]提出的二进制编码的基因型多样性的测度方法来度量粒子群的

多样性, 粒子群多样性 $div = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n h_{ij}}{\frac{1}{2}mn(n-1)}$, 其中, n

为粒子数, m 为粒子的维数, h_{ij} 为粒子 i 到粒子 j 的海明距离。以 MIP 算法计算粒子的适应度, 比较 MDPSO 算法与 DPSO 算法在运行过程中粒子群多样性的变化情况。随机生成一个包含 50 个节点的网络, 节点随机分布在 $1\ 000\text{m} \times 1\ 000\text{m}$ 的平面区域内, 目标节点数为总节点数的 $1/3$, 随机产生源节点 s 和目标节点集 D , 节点最大传输能量 $p_{\max}=8$, 对该网络分别运行 MDPSO 算法与 DPSO 算法。在选取粒子群规模时, 如果粒子数越多, 算法的计算时间越长, 如果粒子数越少, 算法计算结果的精确度越低, 因此应综合考虑算法的计算精确度和计算时间, 通过多次实验测试得出取粒子群规模 $n=30$ 较合适。每个粒子的位置代表一个参与多播树构造的中继节点集, 粒子的维数为除去源节点和目标节点之外其余节点的个数, 即 $m=50-1-50/3=33$, 算法总迭代次数 $duration=100, w_a=0.4, w_b=1.0, t_d=$

0.5duration。每种算法运行 20 次, 统计每种算法在这 20 次运行过程中每次迭代后粒子群的平均多样性, 算法的粒子群多样性变化情况如图 2 所示。在 DPSO 算法运行过程中, 粒子群的多样性很快降低, 算法很快收敛, 全局搜索能力较差。在 MDPSO 算法运行过程中, 在计算前期, 粒子群的多样性较高, 全局搜索能力较强, 在计算后期, 粒子群多样性降低, 局部搜索能力较强。因此, MDPSO 算法有效地平衡了全局搜索能力和局部搜索能力。

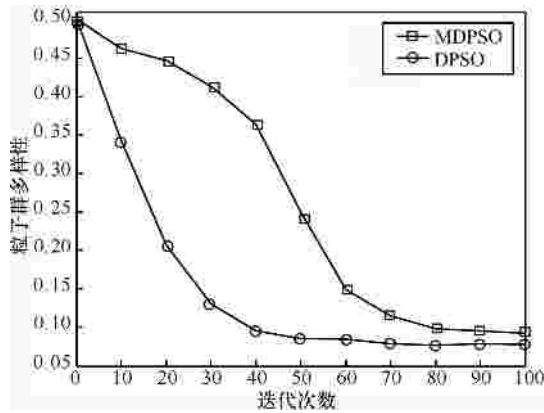


图 2 MDPSO 和 DPSO 粒子群多样性变化情况比较

为了验证 MDPSO 算法的优化能力, 对几个不同规模的网络分别运行 MDPSO 算法和 DPSO 算法。在使用全向天线时使用 MIP 算法计算粒子的适应度, 在使用有向天线时使用 DMIP 算法计算粒子的适应度。每种算法对同一个网络运行 20 次, 统计求得平均解。粒子群规模 $n=30$, 算法总迭代次数 $duration=100$, $w_a=0.4$, $w_b=1.0$, $t_d=0.5duration$ 。网络节点随机分布在 $1\ 000m \times 1\ 000m$ 的平面区域内, 目标节点数为总节点数的 $1/3$, 随机产生源节点 s 和目标节点集 D , 节点最大传输能量 $p_{max}=8$ 。实验结果如表 1 所示, MDPSO 算法求得的平均解普遍优于 DPSO 算法, 表明 MDPSO 算法的优化能力要优于 DPSO 算法。

为了验证构造最小能耗多播树的 MDPSO 算法的性能, 在基于 Java 6.0 的 MyEclipse 8.5 平台上实现了 MDPSO 算法, 并在处理器为 Intel Q6600、内存为 2GB、操作系统为 Microsoft Windows XP 的主机上运行实验程序。为了考虑 MDPSO 算法对不同节点最大传输能量 p_{max} 的适应性, 对多种不同的 p_{max} 分别进行了实验。对于每一种 p_{max} , 与文献[6]和文献[17]类似, 分别对 30 个不同的网络, 每个网络进行 30 次实验, 统计所有计算结果的总平均值。

每次实验中随机产生源节点 s 和目标节点集 D , 同一个网络的 30 次实验中, 每 10 次实验分别采用目标节点数为总节点数的 $1/3$ 、 $1/2$ 、 $2/3$ 。每个网络包含 50 个节点, 随机分布在 $1\ 000m \times 1\ 000m$ 的平面区域内。MDPSO 算法的总迭代次数 $duration$ 取为 30, $w_a=0.6$, $w_b=1.0$, $t_d=0.5duration$, 和目标节点数分别为总节点数的 $1/3$ 、 $1/2$ 、 $2/3$ 相对应, 粒子数分别取为 30、25、20。

表 1 MDPSO 和 DPSO 优化能力比较

| 节点数 | 算法 | 全向天线 | 有向天线 |
|-----|-------|------------|------------|
| 20 | DPSO | 291 999.18 | 74 938.95 |
| | MDPSO | 291 999.18 | 74 938.95 |
| 40 | DPSO | 218 518.27 | 67 467.94 |
| | MDPSO | 217 837.1 | 67 393.49 |
| 60 | DPSO | 395 439.1 | 113 418.64 |
| | MDPSO | 391 425.39 | 112 762.8 |
| 80 | DPSO | 325 292.47 | 86 179.52 |
| | MDPSO | 323 976.18 | 86 024.2 |
| 100 | DPSO | 290 459.28 | 78 805.17 |
| | MDPSO | 290 188.45 | 78 678.13 |

表 2 MDPSO 和 MIP 计算结果比较

| 节点最大传输能量 | MIP 算法 | | MDPSO 算法 | | |
|-------------------|------------|-----------|----------|------------|----------|
| | 平均能耗 | 平均执行时间/ms | 平均降低率 | 平均能耗 | 平均执行时间/s |
| 250 ² | 392 095.16 | 0.45 | 8.87% | 357 327.15 | 0.21 |
| 300 ² | 400 769.09 | 0.3 | 9.42% | 363 011.04 | 0.22 |
| 350 ² | 395 845.72 | 0.38 | 9.27% | 359 141.86 | 0.22 |
| 400 ² | 398 794.62 | 0.42 | 9.24% | 361 928.58 | 0.22 |
| 450 ² | 401 212.07 | 0.41 | 9.08% | 364 780.41 | 0.22 |
| 500 ² | 409 938.03 | 0.53 | 9.18% | 372 291.98 | 0.22 |
| 600 ² | 412 003.33 | 0.31 | 10.09% | 370 427.15 | 0.22 |
| 700 ² | 405 344.76 | 0.44 | 9.38% | 367 317.7 | 0.22 |
| 800 ² | 396 901.74 | 0.4 | 9.17% | 360 518.94 | 0.22 |
| 1000 ² | 403 447.14 | 0.3 | 9.94% | 363 336.21 | 0.22 |
| 8 | 408 597.57 | 0.48 | 9.99% | 367 789.34 | 0.22 |

表 2 显示了在使用全向天线的情况下, 对于多种 p_{max} , MIP 算法和 MDPSO 算法的运行结果。表 3 显示了在使用有向天线的情况下并且最小波束宽度 $\theta_{min}=90^\circ$ 时, 对于多种 p_{max} , D-MIP 算法和 MDPSO 算法的运行结果。实验结果表明, 对于不同的 p_{max} , MDPSO 算法均能有效地优化最小能耗多播树的构造。在使用全向天线的情况下, 当 p_{max} 较小时 MDPSO 算法在计算过程中所遇到的不可行解较多, 随着 p_{max} 的增加不可行解逐渐减少; 而在使用有向天线的情况下, 由于有向天线减小了波束宽度, 延长了通信距离, 即使当 p_{max} 较小时不可行解也很少。与 MIP(D-MIP)算法相比, MDPSO 算法

由于在计算过程中进行了多次迭代，其计算时间相对较长，然而，MDPSO 算法本质上是利用粒子群体进行并行寻优，从而易于设计分布式并行程序来降低算法的执行时间。

表 3 MDP SO 和 D-MIP 计算结果比较

| 节点最大传输能量 | D-MIP 算法 | | MDPSO 算法 | | |
|------------------|------------|-----------|----------|------------|----------|
| | 平均能耗 | 平均执行时间/ms | 平均降低率 | 平均能耗 | 平均执行时间/s |
| 200 ² | 108 684.6 | 4.01 | 7.48% | 100 558.42 | 1.6 |
| 250 ² | 110 694.68 | 3.47 | 7.97% | 101 876.1 | 1.6 |
| 300 ² | 108 565.92 | 3.71 | 8.08% | 99 793.47 | 1.61 |
| 350 ² | 108 959.21 | 3.82 | 7.07% | 101 256.81 | 1.61 |
| 400 ² | 110 649.8 | 3.77 | 7.71% | 102 123.55 | 1.61 |
| 450 ² | 109 466.71 | 3.9 | 8.17% | 100 520.05 | 1.6 |
| 550 ² | 108 920.56 | 3.76 | 7.38% | 100 883.66 | 1.61 |
| 650 ² | 108 466.22 | 3.45 | 7.9% | 99 902.39 | 1.61 |
| 750 ² | 107 687.63 | 3.68 | 8.04% | 99 034.84 | 1.61 |
| 950 ² | 105 838.58 | 3.84 | 7.72% | 97 671.73 | 1.6 |
| 8 | 110 806.09 | 3.97 | 7.72% | 102 250.93 | 1.62 |

6 结束语

在无线 ad hoc 网络中如何构造最小能耗多播路由树是一个重要问题。本文首先分别分析了在使用全向天线和有向天线的情况下该问题的不同数学模型，针对不同的中继节点选择对构造最小能耗多播树的影响，提出了一种改进的离散粒子群算法，以优化最小能耗多播树的构造，最后通过模拟实验验证了改进的离散粒子群算法有效地优化了最小能耗多播树的构造。进一步的研究工作包括将 MDPSO 算法与一些局部优化算法相结合，以及如何更好地设定 MDPSO 算法的控制参数，以获取更优的近似最小能耗多播树。

参考文献：

[1] GUO S, YANG O. Energy-aware multicasting in wireless ad hoc networks: a survey and discussion[J]. Computer Communications, 2007, 30(9):2129-2148.

[2] WIESELTHIER J E, NGUYEN G D, EPHREMIDES A. On the construction of energy-efficient broadcast and multicast trees in wireless networks[A]. Proceedings of IEEE INFOCOM'2000[C]. Tel Aviv, Israel, 2000. 585-594.

[3] 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.

[4] DAS A K, MARKS R J, EL-SHARKAWI M, et al. R-shrink: a heuristic for improving minimum power broadcast trees wireless networks[A]. Proceedings of IEEE GLOBECOM'03[C]. San Francisco, CA, USA, 2003. 523-527.

[5] MONTEMANNI R, GAMBARDILLA L M, DAS A K. The minimum

power broadcast problem in wireless networks: a simulated annealing approach[A]. Proceedings of the 2005 IEEE Wireless Communications and Networking Conference [C]. New Orleans, LA, USA, 2005. 2057-2062.

[6] HERNANDEZ H, BLUM C. Energy-efficient multicasting in wireless ad-hoc networks: an ant colony optimization approach[A]. Proceedings of the 2008 IEEE International Symposium on Wireless Communication Systems[C]. Reykjavik, Iceland, 2008. 667-671.

[7] MIN M, O'BRIEN A F, SHIN S Y. Partitioning-based SOR for minimum energy multicast tree problem in wireless ad hoc networks[A]. Proceedings of the 18th International Conference on Computer Communications and Networks[C]. San Francisco, CA, USA, 2009. 1-6.

[8] ZHONG W L, HUANG J, ZHANG J. A novel particle swarm optimization for the Steiner tree problem in graphs[A]. Proceedings of the 2008 IEEE Congress on Evolutionary Computation[C]. Hong Kong, China, 2008. 2460-2467.

[9] YUAN P, JI C L, ZHANG Y, et al. Optimal multicast routing in wireless ad hoc sensor networks[A]. Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control[C]. 2004. 367-371.

[10] KENNEDY J, EBERHART R. Particle swarm optimization[A]. Proceedings of the 1995 IEEE International Conference on Neural Networks[C]. Perth, Australia, 1995. 1942-1948.

[11] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[A]. Proceedings of the Sixth International Symposium on Micro Machine and Human Science[C]. Nagoya, Japan, 1995. 39-43.

[12] KENNEDY J, EBERHART R. A discrete binary version of the particle swarm algorithm[A]. Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics[C]. Orlando, FL, USA, 1997. 4104-4108.

[13] SHI Y, EBERHART R. A modified particle swarm optimizer[A]. Proceedings of the 1998 IEEE International Conference on Evolutionary Computation[C]. 1998. 69-73.

[14] SHI Y, EBERHART R. Empirical study of particle swarm optimization[A]. Proceedings of the 1999 Congress on Evolutionary Computation[C]. Washington, DC, USA, 1999. 1945-1950.

[15] SHI Y, EBERHART R. Parameter selection in particle swarm optimization[A]. Proceeding of the 1998 Annual Conference on Evolutionary Programming[C]. San Dingo, CA, USA, 1998.591-600.

[16] 武晓今, 朱仲英. 遗传算法多样性测度问题研究[J]. 信息与控制, 2005, 34(4): 416-422.

WU X J, ZHU Z Y. Research on diversity measure of genetic algorithms[J]. Information and Control, 2005, 34(4): 416-422.

[17] AL-SHIHABI S, MERZ P, WOLF S. Nested partitioning for the minimum energy broadcast problem[A]. LIUN 2007 II, Learning and Intelligent Optimization[C]. 2007. 1-11.

作者简介：



朱晓建 (1984-)，男，江苏南通人，东南大学博士生，主要研究方向为无线网络。

沈军 (1963-)，男，江苏南通人，东南大学教授、博士生导师，主要研究方向为面向服务的网络体系结构、网络教学、分布对象计算。