

# 无线自组网中基于离散粒子群优化的睡眠调度感知最小功率广播

朱晓建<sup>1,2</sup>, 沈军<sup>1,2</sup>

(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:** 针对当网络使用睡眠调度并且节点的传输功率连续可调节时的最小功率广播调度问题, 首先给出了一种计算节点内部最优发送调度的递归方法, 然后提出了一种构造最小功率广播调度的离散粒子群算法。该算法搜索最优广播树结构, 并融合基于最小广播功率增量的贪心算法和基于启发式调整广播树结构的局部优化算法以提高收敛速度和求解质量。模拟实验结果表明所提算法能够有效地减少广播功率。

**关键词:** 无线自组网; 睡眠调度; 最小功率; 广播调度; 离散粒子群优化

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2013)06-0016-13

## Sleep scheduling-aware minimum power broadcast in wireless ad hoc networks based on discrete particle swarm optimization

ZHU Xiao-jian<sup>1,2</sup>, SHEN Jun<sup>1,2</sup>

(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:** For the minimum power broadcast scheduling problem where network uses sleep scheduling and each node's transmission power is continuously adjustable, a recursive approach to compute the optimal transmission scheduling of a node was firstly presented, and then a discrete particle swarm optimization algorithm to construct the minimum power broadcast scheduling was proposed. This algorithm searches for the optimal broadcast arborescence, and utilizes the greedy algorithm based on the minimization of the broadcast's power increment and the local optimization algorithm based on the heuristic adjustment of the broadcast arborescence to improve the convergence speed and the result quality. The simulation results show that the proposed algorithm is able to effectively reduce the broadcast power.

**Key words:** wireless ad hoc networks; sleep scheduling; minimum power; broadcast scheduling; discrete particle swarm optimization

### 1 引言

在无线自组网中, 由于无线节点使用能量较有限的电池供能, 节点或网络的生存时间在一定程度上由电池的能量所决定, 合理高效地使用电池能量有利于延长节点或网络的生存时间。广播是某个节点将其报文发送给其他所有节点, 在无线自组网中应用较广, 研究如何减少广播数据所消耗的能量非常必要。

文献[1]讨论了当节点使用全向天线并且传输功率在一定范围内连续可调时的最小功率广播

(MPB, minimum power broadcast)问题, 提出了经典的贪心启发式 BIP(broadcast incremental power)算法, BIP 算法在构造广播树时每次选择一个使得广播功率增加最小的节点加入广播树。BIP 算法的近似比较大, 文献[2]证明了 BIP 算法的近似比介于 13/3~12 之间。文献[3]使用整数规划描述最小功率广播树。文献[4]证明了 MPB 问题的 NP 完全性, 因此寻求性能较好的多项式时间算法是一个重要目标。文献[5]提出了一种基于收缩重叠传输范围的最小功率多播树构造 SOR(shrinking overlapped range)算法, SOR 算法在构造多播树时既考虑树中节点到

收稿日期: 2012-04-19; 修回日期: 2013-01-24

基金项目: 国家重点基础研究发展计划(“973”计划)基金资助项目(2009CB320501)

Foundation Item: The National Basic Research Program of China (973 Program) (2009CB320501)

树外节点的传输,又考虑树中节点到树中节点的传输。不同于最小功率广播树的构造算法,文献[6]提出了一种优化已有广播树的  $r$ -shrink 算法,通过启发式地调整树结构以减少广播功率。文献[7]将求解最小功率广播树问题转化为求解有向 Steiner 树问题,而对于求解有向 Steiner 树已有相关具有性能保证的近似算法。文献[8]综合使用嵌套分区、局部优化(1-shrink 算法)和线性规划求解最小功率广播树。文献[9]运用模拟退火算法求解最小功率广播树,在每次迭代中,随机选择一个传输节点,降低该节点的传输功率,然后按概率以随机的或确定的方式修复受损的广播树。文献[10]提出了一种结合进化算法和局部搜索算法(改进的  $r$ -shrink 算法)求解最小功率广播树的 ELS(evolutionary local search)算法,ELS 算法在每次迭代中,通过变异和局部搜索生成若干新个体,若这些新个体中的最优者优于原有最优解则进化成功。文献[11]提出了一种构造最小功率广播树的蚁群优化算法 ACO(ant colony optimization),与 BIP 算法不同,ACO 算法在构造广播树时不仅考虑启发式信息,而且考虑历史求解信息,并结合基于  $r$ -shrink 算法的 VND(variable neighborhood descent)算法进行局部搜索,求解质量较高。文献[12]结合遗传算法(GA, genetic algorithm)和基于  $r$ -shrink 算法的局部搜索算法求解最小功率广播树,基于节点排列构造广播树,采用概率二进制锦标赛选择、循环交叉、交换变异等遗传算子,并利用基于  $r$ -shrink 算法的局部搜索,计算结果较好。文献[13]结合共生机制处理约束使用一种多相离散粒子群算法分布式地计算最小功率多播树的整数线性规划的最优解,该方法未利用与最小功率多播问题相关的启发式信息会使收敛到最优解的速度较慢,并且在分布式求解整数规划的最优解的过程中,计算节点之间需要进行通信以交换信息,增加了能量消耗,该方法计算代价较大。文献[14]通过使用一种二进制离散粒子群算法优化中继非目标节点的选择来优化最小功率多播树的构造,但最小功率广播树不存在中继非目标节点。文献[15]结合连续粒子群优化算法 PSO(particle swarm optimization)和增强的 VND 算法求解无睡眠调度的最小功率广播问题,其中,粒子位置表示各个节点的传输功率等级,此方法不能用来求解在网络使用睡眠调度的情况下的最小功率广播调度问题,因为节点在使用睡眠调度的情况下会在多个时隙以不同

的功率转发。上述研究在讨论最小功率广播或多播时虽然考虑了节点的传输功率可调节,但未考虑网络使用睡眠调度。本文使用一种离散粒子群优化算法求解睡眠调度感知的最小功率广播调度问题,其中粒子位置表示各个非源节点的父节点。

在无线自组网中,为了减少节点因长期空闲侦听而消耗的能量,节点使用睡眠调度,在活动状态和睡眠状态之间切换<sup>[16-20]</sup>。文献[20]讨论了在使用睡眠调度的网络中的最少传输次数广播问题,其中所有节点的传输范围固定不变且一致,且每个节点仅在网络工作周期的一个时隙处于活动状态,提出了一个集中式的基于集合覆盖的近似 CSCA(centralized set-cover-based approximation)算法,而在某些情况下,各节点的传输范围可能不同,节点的传输范围可以调节变化,节点可在网络工作周期内的多个时隙处于活动状态。文献[21]讨论了睡眠调度感知广播的效率和延迟平衡问题,使用总传输次数表示广播效率。文献[22]讨论了当网络使用睡眠调度时,针对较少量目标节点的最小功率多播路由,提出了一种基于动态规划思想的 oCast(optimal multicast)算法,但 oCast 算法求解最小功率广播问题需要指数级时间。文献[23]研究了睡眠调度感知的最小能量多播问题,其中所有节点具有一致的固定传输功率,提出了具有可保证近似比的多项式时间近似算法。虽然这些研究在讨论最少传输或最小功率广播/多播时考虑了网络使用睡眠调度,但未直接针对当节点传输功率连续可调时的最小功率广播。

本文讨论了当网络使用睡眠调度并且节点的传输功率在一定范围内连续可调时的最小功率广播调度(DCMPBS, duty-cycle-aware minimum power broadcast scheduling)问题,其中,节点可在网络工作周期内的一个或多个时隙处于活动状态。首先对 DCMPBS 问题进行具体阐述,然后讨论节点内部最优发送调度及其求解方法,最后运用一种基于集合的全面学习的离散粒子群优化(S-CLPSO, the discrete version of comprehensive learning particle swarm optimizer based on set-based particle swarm optimization)算法<sup>[24]</sup>求解 DCMPBS 问题,并结合贪心算法和局部优化算法以提高收敛速度和求解质量。

## 2 问题模型

假设无线自组网由  $ns$  个无线节点构成,节点的

位置固定, 令  $AN$  表示网络节点全集。节点配备全向天线, 信号发射功率可以在  $[0, tp_{\max}]$  内进行调节,  $tp_{\max}$  为最大信号发射功率。网络采用睡眠调度的方式工作以节省能耗, 并且使用了相关的协议进行全局范围内的时间同步。假设广播源节点  $s$  已知各节点的位置和睡眠调度, 考虑在源节点  $s$  处集中式地构造最小功率广播调度。本文仅考虑传输能耗。

无线传输的信号功率以  $dist^{-\delta}$  衰减<sup>[1]</sup>,  $dist$  是到信号发送者的距离,  $\delta$  为通信媒介参数, 通常取值为 2~4, 本文取  $\delta=2$ 。假设信号能被正确接收的最低功率为 1。令  $dist_{ij}$  表示节点  $i$  与节点  $j$  之间的距离, 当节点  $i$  的发射功率  $tp_i \geq (dist_{ij})^\delta$  时, 节点  $j$  可以接收到节点  $i$  发送的信号。位于节点  $i$  的发送范围内的节点都可以接收到节点  $i$  所发送的信号, 这使得无线网络的广播路由不同于有线网络的广播路由。令  $lp_{ij}$  表示节点  $i$  到节点  $j$  的链路功率即  $lp_{ij} = (dist_{ij})^\delta$ 。

假设网络的工作时间以  $WP$  为周期, 即每个节点的睡眠调度周期为  $WP$ , 工作周期  $WP$  被等分为  $wpl$  个等长的时隙, 每个时隙足够长以使得可以完整地发送或接收一个报文<sup>[18-20]</sup>。使用  $ts_g (g \geq 0)$  表示网络工作时间的第  $g$  个时隙, 使用  $\tau_h (0 \leq h < wpl)$  表示工作周期  $WP$  的第  $h$  个时隙。如果  $ts_0$  对应第 0 个工作周期的  $\tau_0$ , 则  $ts_g$  对应第  $\lfloor g/wpl \rfloor$  个工作周期的时隙  $\tau_{(g-\lfloor g/wpl \rfloor wpl)}$ , 反之第  $q$  个工作周期的时隙  $\tau_h$  对应时隙  $ts_{(q \cdot wpl + h)}$ 。节点在工作周期  $WP$  内的一个或多个时隙处于活动状态<sup>[19]</sup>, 在其他时隙处于睡眠状态, 令  $AT_i$  表示节点  $i$  的活动时隙集, 假设各节点在工作周期  $WP$  内具有相同的活动时隙数, 令占空比  $dc$  表示工作周期内的活动时隙数占工作周期总时隙数  $wpl$  的比例。由于节点的能量较为有限, 为了有效地延长网络生命期,  $dc$  通常较小。节点只能在活动时隙接收, 但可在任意时隙唤醒自己以发送报文<sup>[18-20]</sup>。对于节点的一次传输, 在此次传输的通信范围内的节点不都处于活动状态, 所以这些节点不都能接收到该次传输所传递的报文, 因此对于单个报文的广播, 一个节点可能需要进行多次传输。当调节节点  $i$  在时隙  $\tau_h$  的传输功率时, 相应地在时隙  $\tau_h$  被节点  $i$  所覆盖的节点集会发生变化。本文所讨论的广播调度建立在已有的睡眠调度的基础上, 即在各节点的活动时隙集已确定的情况下考虑广播调度。

假设在时隙  $ts_f$  节点  $s$  发起一个广播请求, 在网络使用睡眠调度并且节点的传输功率在  $[0, tp_{\max}]$  内连续可调的情况下, 最小功率广播调度(DCMPBS)问题就是要构造一个广播调度  $BS = \{(s, ts_y, tp(s, ts_y)), (j, ts_h, tp(j, ts_h)), \dots, (k, ts_l, tp(k, ts_l))\} (f+1 \leq y \leq h \leq \dots \leq l)$ ,  $BS$  中的任一  $(i, ts_g, tp(i, ts_g))$  表示节点  $i$  在时隙  $ts_g$  以功率  $tp(i, ts_g)$  进行传输, 通过  $BS$  除了源节点  $s$  之外其他每个节点都能接收到报文, 并且  $BS$  的总传输功率  $TP(BS) = \sum_{(i, ts_g, tp(i, ts_g)) \in BS} tp(i, ts_g)$  最小。

DCMPBS 问题是一个 NP 完全问题。当  $dc=1$  时, DCMPBS 问题就是文献[1]中的 MPB 问题, 因此, MPB 问题是 DCMPBS 问题的一个特例。因为 MPB 问题是 NP 完全的<sup>[4]</sup>, 所以 DCMPBS 问题是 NP 完全的。

最小功率广播调度首先是一棵广播树, 该树确定了节点之间的邻接关系(发送接收关系)。其次, 在已知节点的孩子节点(接收节点)集后, 由于各个孩子节点的活动时隙集不尽相同, 在节点内部存在一个最优发送调度使得覆盖所有孩子节点所需的传输功率最小。因此, 先要构造一棵广播树  $T(N_T, E_T)$ ,  $N_T$  为树  $T$  的节点集,  $N_T = AN$ ,  $E_T$  为树  $T$  的边集, 树  $T$  的根节点为广播源节点  $s$ 。在树  $T$  中已知节点  $i$  的孩子节点集  $CH_i$  以及每个孩子节点  $j (j \in CH_i)$  的活动时隙集  $AT_j$ , 需要考虑如何调度节点  $i$  的发送并确定每次发送的传输功率。当确定了节点  $i$  在时隙  $ts_g$  的接收节点集  $RN(i, ts_g)$  后, 节点  $i$  在时隙  $ts_g$  的传输功率  $tp(i, ts_g) = \max_{j \in RN(i, ts_g)} \{(dist_{i,j})^\delta\}$ 。

### 3 节点内部最优发送调度

在给定了发送节点的接收节点集后, 由于各个接收节点的活动时隙不尽一致, 在发送节点内部存在一个最优发送调度使得覆盖所有接收节点所需的传输功率最小。文献[22]讨论了当节点具有固定传输功率时的节点内部最优发送调度, 只要求得最少传输次数即可, 此方法不能用来求当节点的传输功率连续可调时的节点内部最优发送调度; 若使用文献[22]的 OCast 算法计算节点内部最优发送调度, 需转化为求解有向 Steiner 树, 这样过于复杂和繁琐。因此, 针对节点传输功率连续可调, 给出了一种直接计算节点内部最小传输功率发送调度的递归方法。

在图 1 中, 发送节点 0 到其孩子节点 1~5 的

距离依次为  $dist_{0,1}=10$ 、 $dist_{0,2}=8$ 、 $dist_{0,3}=5$ 、 $dist_{0,4}=4$ 、 $dist_{0,5}=3$ ，节点1~5的活动时隙集依次为  $AT_1=\{\tau_1, \tau_3\}$ 、 $AT_2=\{\tau_1, \tau_4\}$ 、 $AT_3=\{\tau_4, \tau_7\}$ 、 $AT_4=\{\tau_3, \tau_5\}$ 、 $AT_5=\{\tau_4, \tau_6\}$ 。由各孩子节点的活动时隙集可知，节点0的最少传输次数为2，即在时隙  $\tau_3$  和  $\tau_4$  传输。当节点0的传输功率连续可调时，最少传输次数发送调度为  $tp(0, \tau_3)=lp_{0,1}$ ， $tp(0, \tau_4)=lp_{0,2}$ ，总传输功率为  $lp_{0,1}+lp_{0,2}=10^2+8^2=164$ ；而最小传输功率发送调度为  $tp(0, \tau_1)=lp_{0,1}$ 、 $tp(0, \tau_4)=lp_{0,3}$ 、 $tp(0, \tau_3)$ (或  $tp(0, \tau_5)=lp_{0,4}$ )，总传输功率为  $lp_{0,1}+lp_{0,3}+lp_{0,4}=10^2+5^2+4^2=141$ 。因此，求当节点具有固定传输功率时节点内部最优发送调度的方法不能用来求当节点具有连续可调的传输功率时的节点内部最优发送调度。

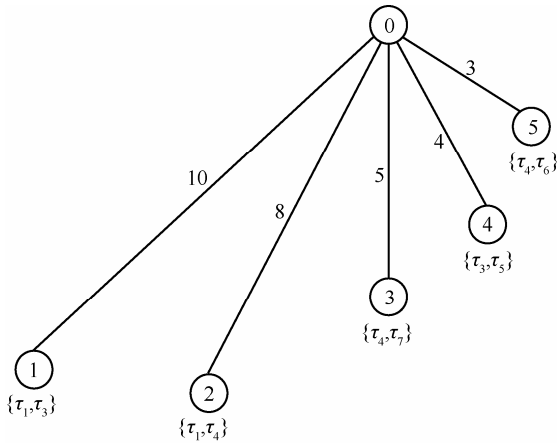


图1 一个发送节点及其接收节点集

在 MPB 问题中，节点的传输功率为到它的最远孩子节点的链路功率，而在 DCMPBS 问题中，在已知了节点的孩子节点集及各孩子节点的活动时隙集时，存在一个最优发送调度使得覆盖所有孩子节点所需的传输功率最小。在图1中，首先，由于节点1距离节点0最远，节点0必须在时隙  $\tau_1$  或  $\tau_3$  以功率  $lp_{0,1}$  传输才能覆盖节点1。考虑节点0在不同时隙覆盖节点1的情况。

1) 若节点0在时隙  $\tau_3$  以功率  $lp_{0,1}$  传输，可以覆盖节点{1,4}。节点0在时隙  $\tau_4$  以功率  $lp_{0,2}$  传输可以覆盖剩余节点{2,3,5}。节点0的总传输功率为  $lp_{0,1}+lp_{0,2}=10^2+8^2=164$ 。

2) 若节点0在时隙  $\tau_1$  以功率  $lp_{0,1}$  传输，可以覆盖节点{1,2}。覆盖剩余节点{3,4,5}的最优发送调度为：节点0在时隙  $\tau_4$  以功率  $lp_{0,3}$  传输可以覆盖节点{3,5}；节点0在时隙  $\tau_3$  或  $\tau_5$  以功率  $lp_{0,4}$  传输给节点4。节点0的总传输功率为  $lp_{0,1}+lp_{0,3}+lp_{0,4}=10^2+5^2+4^2=141$ 。

$$10^2+5^2+4^2=141 < lp_{0,1}+lp_{0,2}=164。$$

通过比较节点在各时隙以不同的功率发送，可以找到一个最优发送调度。在图1所示的实例中，节点0的一个最优发送调度为： $tp(0, \tau_1)=lp_{0,1}$ ， $tp(0, \tau_4)=lp_{0,3}$ ， $tp(0, \tau_3)=lp_{0,4}$ 。

本文假设每个集合都是一个序列，对于任一集合  $S$ ，使用  $|S|$  表示集合  $S$  的大小， $S[q]$  表示集合  $S$  中的第  $q$  个元素。已知节点  $i$  的孩子节点集  $CH_i$  以及各个孩子节点  $j(j \in CH_i)$  的活动时隙集  $AT_j$ ，所有孩子节点的活动时隙构成一个时隙集合  $UAT_i = \bigcup_{j \in CH_i} AT_j$ 。令  $NSS_i = (tp(i, UAT_i[0]), tp(i, UAT_i[1]), \dots, tp(i, UAT_i[|UAT_i|-1]))^T$  表示节点  $i$  在  $UAT_i$  中各时隙的传输功率，即表示节点  $i$  的一个发送调度，其传输功率  $tp(NSS_i) = \sum_{q=0}^{|UAT_i|-1} tp(i, UAT_i[q])$ 。求解节点  $i$  的最

优发送调度问题就是要确定一个  $NSS_i$ ，使得可以覆盖  $CH_i$  且  $tp(NSS_i)$  最小。使用  $CH(i, \tau_g)$  ( $\tau_g \in UAT_i$ ) 表示节点  $i$  的在时隙  $\tau_g$  活动的孩子节点集，即  $CH(i, \tau_g) = \{j | j \in CH_i, \tau_g \in AT_j\}$ 。节点  $i$  在时隙  $\tau_g$  可以以功率  $lp_{ij}$  ( $j \in CH(i, \tau_g)$ ) 或 0 进行传输，相应地可以覆盖  $CH(i, \tau_g)$  的一个子集。当节点  $i$  在  $UAT_i$  中的各个时隙的传输功率都确定，即确定了一个发送调度  $NSS_i$ ，可以计算出被  $NSS_i$  覆盖的节点集  $CN(NSS_i)$ ，若  $CN(NSS_i) = CH_i$ ，则该  $NSS_i$  是有效的，否则是无效的。通过比较所有有效发送调度，可以找到一个总传输功率最小的有效发送调度。

已知节点  $i$  的一个孩子节点集  $Z$ ， $Z \subseteq CH_i$ ，以及一个时隙集合  $Y$ ， $Y \subseteq UAT_i$ ，求节点  $i$  在时隙集  $Y$  覆盖  $Z$  的最优发送调度  $NOSS(i, Y, Z)$ ，使得可以覆盖  $Z$  并且总传输功率最小。使用  $ComputeNOSS(i, Y, Z)$  算法计算  $NOSS(i, Y, Z)$ ，其中， $Y$  用数组表示，若无解则返回 NULL，否则返回一个传输功率数组  $PV$ ，其长度  $|PV|=|Y|$ ， $\forall 0 \leq q \leq |Y|-1$ ， $PV[q]$  为节点  $i$  在时隙  $Y[q]$  的传输功率。则调用  $ComputeNOSS(i, UAT_i, CH_i)$  可以计算出节点  $i$  覆盖其孩子节点  $CH_i$  的最优发送调度。 $ComputeNOSS(i, Y, Z)$  算法如下。

**Step1** 如果  $Y = \Phi$ ，则返回 NULL。如果  $Z$  中存在某个节点  $j$ ，节点  $j$  在  $Y$  中的任一时刻都不活动，则返回 NULL。初始化  $PV$  为一个长度为  $|Y|$  的数组， $\forall 0 \leq q \leq |Y|-1$ ， $PV[q] \leftarrow 0$ 。

**Step2** 如果  $Z = \Phi$ ，则转 Step6。如果  $dc \cdot wpl = 1$ ，则首先  $\forall 0 \leq q \leq |Y|-1$ ， $C \leftarrow CH(i, Y[q]) \cap Z$ ， $PV[q] \leftarrow$

( $C \neq \Phi? \max_{j \in C} \{lp_{i,j}\}:0$ ), 然后转 Step6。如果存在某个  $q, 0 \leq q \leq |Y|-1, CH(i, Y[q]) \supseteq Z$ , 则  $PV[q] \leftarrow \max_{j \in Z} \{lp_{i,j}\}$ , 转 Step6。

**Step3**  $res \leftarrow +\infty$ , 初始化  $SY$  为一个长度为  $|Y|-1$  的数组,  $\forall 0 \leq q \leq |Y|-2, SY[q] \leftarrow Y[q+1]$ 。

**Step4**  $SPV \leftarrow \text{ComputeNOSS}(i, SY, Z)$ 。如果  $SPV \neq \text{NULL}$ , 则  $res \leftarrow \sum_{q=0}^{|SPV|-1} SPV[q]$ ,  $\forall 1 \leq q \leq |PV|-1, PV[q] \leftarrow SPV[q-1]$ 。  $C \leftarrow CH(i, Y[0]) \cap Z$ 。如果  $C = \Phi$ , 则转 Step6。  $q \leftarrow 0$ 。

**Step5**  $j \leftarrow C[q]$ ,  $A \leftarrow \{k | k \in C, lp_{i,k} \leq lp_{i,j}\}$ 。  $SPV \leftarrow \text{ComputeNOSS}(i, SY, Z-A)$ 。如果  $SPV \neq \text{NULL}$  并且  $lp_{i,j} + \sum_{u=0}^{|SPV|-1} SPV[u] < res$ , 则  $res \leftarrow lp_{i,j} + \sum_{u=0}^{|SPV|-1} SPV[u]$ ,  $PV[0] \leftarrow lp_{i,j}$ ,  $\forall 1 \leq u \leq |PV|-1, PV[u] \leftarrow SPV[u-1]$ 。  $q \leftarrow q+1$ 。如果  $q \leq |C|-1$ , 转 Step5。

**Step6** 返回  $PV$ 。

## 4 贪心(DCBIP, duty-cycle-aware BIP)算法和局部优化(DCShrink, duty-cycle-aware shrink)算法

### 4.1 贪心算法

BIP 算法<sup>[1]</sup>的基本思想是从源节点开始每次选择一个使得广播功率增加最小的节点加入广播树。根据 BIP 算法<sup>[1]</sup>的最小化广播功率增量思想求解 DCMPBS 问题。在将节点  $j$  成为节点  $i$  的孩子节点后, 节点  $i$  的传输功率增量  $incr_{i,j} = tp'_{i,j} - tp_i$ ,  $tp_i$  为节点  $i$  原有的最优发送调度的传输功率,  $tp'_{i,j}$  为节点  $i$  在新增节点  $j$  成为其孩子节点之后的最优发送调度的传输功率, 节点的最优发送调度使用 ComputeNOSS 算法进行计算。在构造广播树的过程中, 在每次迭代中, 对于每一对满足  $lp_{i,j} \leq tp_{\max}$  的树中节点  $i$  和树外节点  $j$  计算  $incr_{i,j}$ , 选择最小的  $incr_{i,j}$ , 将其对应的节点  $j$  加入广播树并成为节点  $i$  的孩子节点。

### 4.2 局部优化算法

对于 MPB 问题,  $r$ -shrink 算法<sup>[6]</sup>是对已有的广播树进行优化, 其基本思想是对于  $|CH_i| \geq r$  的节点  $i$ , 移除节点  $i$  的  $r$  个最远的孩子节点, 并将这  $r$  个节点重新加入广播树中, 满足最大传输功率限制和树结构限制, 并且不能再成为节点  $i$  的孩子节点, 使得覆盖这  $r$  个节点所需的传输功率增量最小并且小于节点  $i$  因移除这  $r$  个孩子节点所减少的传输功率,

如果能够实现就按此执行广播树的调整, 从而使得广播树得到优化。 $r$ -shrink 算法<sup>[6]</sup>自底向上依次对每个非叶节点进行处理, 直至广播树无法被优化。

在不使用睡眠调度时, 只有移除了节点  $i$  的最远孩子节点, 节点  $i$  的传输功率才会减少。在使用睡眠调度时, 移除节点  $i$  的最远孩子节点肯定会导致节点  $i$  的传输功率减少, 移除其他孩子节点可能也会导致节点  $i$  的传输功率的减少。在图 1 所示的实例中,  $(tp(0, \tau_1) = lp_{0,1}, tp(0, \tau_4) = lp_{0,3}, tp(0, \tau_3) = lp_{0,4})^T$  是节点 0 的一个最优发送调度, 移除孩子节点  $\{1, 3, 4\}$  中的任意一个肯定会导致节点 0 的传输功率减少。除此之外, 在移除孩子节点 2 后, 节点 0 的最优发送调度变为  $(tp(0, \tau_3) = lp_{0,1}, tp(0, \tau_4) = lp_{0,3})^T$ , 节点 0 的传输功率得到减少。考虑一次性移除  $r (r > 1)$  个孩子节点, 在无睡眠调度的情况下, 只有一种情况即移除  $r$  个最远的孩子节点, 而在使用睡眠调度的情况下, 将有多种组合。为了简化讨论, DCShrink 算法每次仅考虑移除一个孩子节点。DCShrink 算法在处理传输节点  $i$  时, 不仅需要考虑最远孩子节点, 还需要考虑其他的在被移除后会导致节点  $i$  的传输功率减少的孩子节点。

令  $tp''_{i,k}$  表示移除节点  $i$  的孩子节点  $k$  后节点  $i$  的最优发送调度的总传输功率。DCShrink 算法与文献[12]的局部搜索算法一样按随机顺序处理各传输节点, 同时与文献[6]的  $r$ -shrink 算法一样若广播树得到优化就重新开始新的循环。令  $DES_j$  表示节点  $j$  的后代节点集, 基于 1-shrink 算法<sup>[6]</sup>, 给定了一棵广播树  $T$  并已知  $T$  中各个非叶子节点的最优发送调度, 对该广播树  $T$  进行局部优化的 DCShrink 算法如下。

**Step1**  $impr \leftarrow \text{false}$ ,  $I \leftarrow T$  中所有传输节点的集合。

**Step2** 如果  $I \neq \Phi$ , 从  $I$  中随机选择一个节点  $i$ ,  $I \leftarrow I - \{i\}$ , 否则, 转 Step7。

**Step3**  $J \leftarrow CH_i$ 。

**Step4** 如果  $J \neq \Phi$ , 从  $J$  中随机选择一个节点  $j$ ,  $J \leftarrow J - \{j\}$ , 否则, 转 Step6。如果  $tp''_{i,j} = tp_i$ , 则转 Step4。

**Step5**  $FP \leftarrow \{k | k \in AN - DES_j - \{i, j\}, lp_{k,j} \leq tp_{\max}, tp'_{k,j} - tp_k < tp_i - tp''_{i,j}\}$ 。如果  $FP = \Phi$ , 则转 Step4。  $bp \leftarrow \text{argmin}_{k \in FP} \{tp'_{k,j} - tp_k\}$ 。在树  $T$  中, 移除节点  $i$  的孩子节点  $j$  并将节点  $j$  成为节点  $bp$  的孩子节点, 更新节点  $i$  和节点  $bp$  的最优发送调度。  $impr \leftarrow \text{true}$ 。转 Step3。

**Step6** 如果  $impr=true$ , 转 Step1, 否则, 转 Step2.

**Step7** 返回  $T$ .

## 5 构造睡眠调度感知最小功率广播调度的离散粒子群(SCLPSODCMPBS, S-CLPSO for the DCMCBS problem)算法

### 5.1 S-CLPSO 算法简介

粒子群优化(PSO, particle swarm optimization)<sup>[25,26]</sup>模拟社会行为, 粒子具有认知和学习能力, 在多维空间执行随机搜索。粒子具有位置和速度 2 个属性, 位置表示问题的解, 速度表示位移。粒子记录自身所经历的最好位置即个体极值点, 群体记录所有粒子所经历的最好位置即全局极值点。粒子在搜索过程中追踪个体极值和群体极值, 结合已有的惯性, 进行有偏向的随机搜索。使用  $b$  表示解空间的维数, 向量  $V_m=(v_m^0, v_m^1, \dots, v_m^{b-1})^T$  表示粒子  $m$  的速度, 向量  $X_m=(x_m^0, x_m^1, \dots, x_m^{b-1})^T$  表示粒子  $m$  的位置,  $fit(X_m)$  表示粒子  $m$  的适应度值。向量  $PBX_m=(pbx_m^0, pbx_m^1, \dots, pbx_m^{b-1})^T$  表示粒子  $m$  的个体极值点, 索引  $gbest$  指向个体极值最优的粒子。在 PSO<sup>[25,26]</sup>中, 粒子的速度按式(1)进行更新, 粒子的位置按式(2)更新。

$$v_m^d(t+1) = wv_m^d(t) + ac_1 \cdot rand_1() (pbx_m^d(t) - x_m^d(t)) + ac_2 \cdot rand_2() (pbx_{gbest}^d(t) - x_m^d(t)) \quad (1)$$

$$x_m^d(t+1) = x_m^d(t) + v_m^d(t+1) \quad (2)$$

其中,  $w$  为惯性权重<sup>[27]</sup>,  $ac_1$  和  $ac_2$  为加速因子,  $rand_1()$  和  $rand_2()$  为在区间 $[0,1]$ 上均匀分布的随机函数。

为了解决组合优化问题, 文献[24]提出了一种基于集合的粒子群优化(S-PSO, set-based PSO), S-PSO 重新定义了粒子的速度和位置两者的含义及更新算子。文献[28]使用 S-PSO<sup>[24]</sup>求解带时间窗的车辆路由问题, 获得了较好的计算结果, 显示了 S-PSO<sup>[24]</sup>具有较好的性能。组合优化问题的解由若干离散元素构成, 假设元素全集为  $E$ ,  $E_d$  表示与维度  $d$  相关的元素集。在 S-PSO<sup>[24]</sup>中, 粒子位置的每一维  $x_m^d$  表示构成位置  $X_m$  所表示的解的元素中与维度  $d$  相关的元素集合,  $x_m^d \subseteq E_d$ 。粒子速度的每一维  $v_m^d$  表示与维度  $d$  相关的元素概率集合,  $v_m^d = \{e/p_m^d(e) | e \in E_d\}$ , 概率  $p_m^d(e)$  反映了粒子  $m$  利用元素  $e$  构造解的第  $d$  维的可能性。S-PSO<sup>[24]</sup>的速度更新中的算子定义如下。

1) 系数  $c$  乘以速度  $v_m^d$  ( $v_m^d = \{e/p_m^d(e) | e \in E_d\}$ ) 定义为  $c v_m^d = \{e/\min\{1, c p_m^d(e)\} | e \in E_d\}$ 。

2) 位置  $x_m^d$  减去位置  $x_n^d$  定义为  $\Delta x^d = x_m^d - x_n^d = \{e | e \in x_m^d, e \notin x_n^d\}$ 。

3) 系数  $c$  乘以位置之差  $\Delta x^d$  定义为  $c \cdot \Delta x^d = \{e/p(e) | e \in E_d\}$ ,  $p(e) = \begin{cases} \min\{1, c\}, & e \in \Delta x^d \\ 0, & e \notin \Delta x^d \end{cases}$ 。

4) 速度  $v_m^d = \{e/p_m^d(e) | e \in E_d\}$  加上速度  $v_n^d = \{e/p_n^d(e) | e \in E_d\}$  定义为

$$v_m^d + v_n^d = \{e/\max\{p_m^d(e), p_n^d(e)\} | e \in E_d\}$$

在 S-PSO<sup>[24]</sup>中, 粒子位置更新采用构造式的方法。令  $\alpha$  表示一个  $[0,1]$  上的随机数。按照某种顺序依次处理每一维, 对于  $x_m^d(t+1)$ , 首先从  $cut_\alpha(v_m^d(t+1)) = \{e | e/p_m^d(e) \in v_m^d(t+1), p_m^d(e) \geq \alpha\}$  中选择满足约束条件的元素构造  $x_m^d(t+1)$ , 若未能构造完  $x_m^d(t+1)$ , 再从先前位置  $x_m^d(t)$  中选择满足约束条件的元素构造  $x_m^d(t+1)$ , 若仍未能构造完成再在剩余元素即  $E_d - cut_\alpha(v_m^d(t+1)) - x_m^d(t)$  中选择满足约束条件的元素进行构造。元素选择算子以及对各维的处理顺序需要根据具体求解的问题进行确定, 使用启发式选择算子有利于加快收敛速度。

在全面学习的粒子群优化(CLPSO, comprehensive learning particle swarm optimizer)<sup>[29]</sup>中, 粒子的每一维独立地选择学习榜样, 可以增强群体多样性从而避免早熟收敛。文献[24]的研究结果表明基于 S-PSO 的 CLPSO<sup>[29]</sup>(S-CLPSO)具有较优的性能, 在 S-CLPSO 中, 粒子速度和位置的含义及更新算子与 S-PSO 相同, 粒子速度的更新规则与 CLPSO<sup>[29]</sup>相同。CLPSO<sup>[29]</sup>采用式(3)更新粒子速度。

$$v_m^d(t+1) = w v_m^d(t) + ac \cdot rand() (pbx_{ep_m^d}^d(t) - x_m^d(t)) \quad (3)$$

其中,  $ac$  为加速因子,  $rand()$  生成在  $[0,1]$  上均匀分布的随机数,  $ep_m^d$  表示被粒子  $m$  的第  $d$  维所学习的粒子。向量  $EP_m=(ep_m^0, ep_m^1, \dots, ep_m^{b-1})^T$  就表示了粒子  $m$  的各维的学习榜样。在选择学习榜样  $ep_m^d$  时, 以概率  $1-pc_m$  选择自身, 以概率  $pc_m$  按锦标赛选择方式选择一个其他粒子, 即从随机选择的 2 个其他粒子中选择个体极值较优者<sup>[29]</sup>。  $pc_m$  表示粒子  $m$  的

$$\text{学习概率, } pc_m = 0.05 + 0.45 \times \frac{\exp(\frac{10m}{pss-1}) - 1}{\exp(10) - 1} \quad (m \in [0,$$

$pss-1]$ )<sup>[29]</sup>, 其中,  $pss$  为粒子群规模。为了使粒子能够追踪较好的学习榜样, 如果粒子的个体极值连续  $CG$  代都没有进化, 就重新选择粒子各维的学习榜样  $EP_m$ <sup>[29]</sup>,  $CG$  称为与更新粒子学习榜样相关的粒子个体极值连续停止进化代数阈值。

### 5.2 使用 S-CLPSO 算法求解 DCOMPBS 问题

DCBIP 算法近似比较大, DCShrink 算法只能对给定的树进行优化, 对不同的树优化结果不尽相同。为了获得更优的广播调度, 采用 S-CLPSO 算法<sup>[24]</sup> 搜寻最优广播树, 并结合 DCBIP 算法和 DCShrink 算法以提高收敛速度和求解质量。在初始化阶段, 以一定的概率使用 DCBIP 算法构造粒子的初始位置; 在更新粒子位置时, 按照 DCBIP 算法的最小化广播功率增量的思想依次处理各维和选择父节点。使用 DCShrink 算法局部优化粒子的位置。

#### 5.2.1 粒子含义

粒子位置  $X_m$  表示一棵广播树, 假设维度  $d$  对应网络节点  $wn(d)$ ,  $x_m^d$  表示节点  $wn(d)$  的父节点。由于广播树以源节点  $s$  为根, 源节点  $s$  无父节点, 因此粒子位置  $X_m$  无需包含源节点  $s$  的父节点, 若网络节点数为  $ns$ , 则粒子维数  $b=ns-1$ ,  $\{wn(d)|0 \leq d < ns-1\} = AN - \{s\}$ 。由于节点数为  $ns$  的广播树由  $ns-1$  条有向边构成, 因而构成解的基本元素就为有向边, 有向边全集  $E = \{<i, j> | i \in AN, j \in AN, i \neq j, lp_{ij} \leq tp_{max}\}$ 。  $E_d$  表示和节点  $wn(d)$  相关的有向边集,  $x_m^d$  表示节点  $wn(d)$  的父节点,  $E_d$  就由可以传输到达节点  $wn(d)$  的节点构成, 即  $E_d = \{i | i \in AN, i \neq wn(d), lp_{i,wn(d)} \leq tp_{max}\}$ 。相应地, 粒子速度的每一维  $v_m^d$  表示可以传输到达节点  $wn(d)$  的节点及其概率集合。

#### 5.2.2 粒子初始化

使用贪心算法初始化粒子位置有利于加快收敛速度<sup>[24,28]</sup>, 在初始化粒子位置  $X_m$  时, 以概率  $\varphi$  使用 DCBIP 算法构造  $X_m$ , 以概率  $1-\varphi$  随机初始化  $X_m$ 。

在随机初始化粒子位置  $X_m$  时, 每次先从除了源节点之外其他所有的无父节点的节点中随机选择一个节点  $i$ , 再从所有的既是节点  $i$  的非后代又可以传输到达节点  $i$  的节点中随机选择一个节点  $j(lp_{ji} \leq tp_{max}, j \neq i)$  作为节点  $i$  的父节点, 重复执行上述步骤直至除了源节点之外其他每个节点都具有父节点。如果对于某个无父节点的节点  $i$ , 不存在既是节点  $i$  的非后代又可以传输到达节点  $i$  的节点, 则重新对粒子位置  $X_m$  进行随机初始化, 直至随机初始化成功。

随机初始化粒子速度  $V_m$ , 在初始化每一维  $v_m^d$  ( $v_m^d = \{el | p_m^d(e) | e \in E_d\}$ ) 时, 对于  $v_m^d$  中的每个元素  $el | p_m^d(e)$ ,  $p_m^d(e) = \begin{cases} rand_2(0), & \text{若 } rand_1(0) \leq 0.5 \\ 0 & \text{, 其他} \end{cases}$ 。

#### 5.2.3 粒子位置更新

在更新粒子位置  $X_m$  时, 按照最小化广播功率增量原则<sup>[1]</sup> 依次处理各维, 每次在除了源节点之外的所有无父节点的节点中选择一个节点, 覆盖该节点所需的广播功率增量最小, 确定其父节点, 直至新粒子位置的每维对应的节点都具有父节点。

在计算覆盖无父节点的节点  $wn(d)$  所需的最小广播功率增量时, 首先计算节点  $wn(d)$  在新位置  $X_m(t+1)$  所表示的中间树  $(X_m(t+1))$  的一些维度对应的节点尚未确定父节点) 中的后代节点集  $DES(X_m(t+1), wn(d))$ , 然后按优先级由高到低依次从  $cut_\alpha(v_m^d(t+1))$ 、 $\{x_m^d(t)\}$ 、 $E_d - cut_\alpha(v_m^d(t+1)) - \{x_m^d(t)\}$  中选择一个节点  $i$ <sup>[24]</sup>, 满足  $i \notin DES(X_m(t+1), wn(d))$  且  $ntp(X_m(t+1), i, wn(d)) - otp(X_m(t+1), i)$  最小, 若节点  $i$  存在, 则覆盖节点  $wn(d)$  所需的最小广播功率增量就为  $ntp(X_m(t+1), i, wn(d)) - otp(X_m(t+1), i)$ , 对应的传输节点就为  $i$ 。其中,  $ntp(X_m(t+1), i, wn(d))$  表示在  $X_m(t+1)$  所表示的中间树中假设将节点  $wn(d)$  成为节点  $i$  的孩子节点后节点  $i$  的最优发送调度的传输功率,  $otp(X_m(t+1), i)$  表示在  $X_m(t+1)$  所表示的中间树中节点  $i$  的最优发送调度的传输功率。

更新粒子位置  $X_m$  的算法为 UpdateX, 子算法 UpdatePa 启发式地选择父节点, 其中,  $INCR[d]$  为覆盖节点  $wn(d)$  所需的最小传输功率增量,  $PA[d]$  为  $INCR[d]$  对应的传输节点。在选择父节点时, 如果对于某一维  $d$ , 不存在满足约束条件的节点可以作为节点  $wn(d)$  的父节点, 则说明  $X_m(t+1)$  表示了一个不可行解, UpdatePa 返回计算失败, 相应地 UpdateX 返回计算失败。更新粒子位置  $X_m$  的 UpdateX 算法如下。

**Step1** 产生一个  $[0,1]$  上的随机数  $\alpha$ 。初始化  $PA$  为一个长度为  $ns-1$  的数组。初始化  $INCR$  为一个长度为  $ns-1$  的数组。  $\forall 0 \leq d \leq ns-2, x_m^d(t+1) \leftarrow -1$ 。

**Step2** 使用 UpdatePa 算法更新  $PA$  和  $INCR$ 。如果 UpdatePa 计算失败, 则返回 false 表示计算失败。  $D \leftarrow \{d | 0 \leq d \leq ns-2, x_m^d(t+1) = -1\}$ 。  $u \leftarrow \operatorname{argmin}_{d \in D} \{INCR[d]\}$ ,  $x_m^u(t+1) \leftarrow PA[u]$ 。如果  $|D| > 1$ , 则转 Step2。

**Step3** 返回新粒子位置  $X_m(t+1)$  以及 true 表示计算成功。

启发式选择父节点的 UpdatePa 算法如下。

**Step1**  $d \leftarrow 0$ 。

**Step2** 如果  $x_m^d(t+1) \neq -1$ , 则转 Step5。

**Step3**  $FP \leftarrow \{j | j \in cut_\alpha(v_m^d(t+1)), j \notin DES(X_m(t+1), wn(d))\}$ 。如果  $FP \neq \Phi$ , 则转 Step4。如果  $x_m^d(t) \notin DES(X_m(t+1), wn(d))$ , 则  $FP \leftarrow \{x_m^d(t)\}$ , 转 Step4。 $FP \leftarrow \{j | j \in E_d - cut_\alpha(v_m^d(t+1)) - \{x_m^d(t)\}, j \notin DES(X_m(t+1), wn(d))\}$ 。如果  $FP \neq \Phi$ , 则转 Step4。返回 false 表示计算失败。

**Step4**  $i \leftarrow \operatorname{argmin}_{j \in FP} \{ntp(X_m(t+1), j, wn(d)) - otp(X_m(t+1), j)\}$ 。 $PA[d] \leftarrow i$ ,  $INCR[d] \leftarrow ntp(X_m(t+1), i, wn(d)) - otp(X_m(t+1), i)$ 。

**Step5**  $d \leftarrow d+1$ 。如果  $d < ns-1$ , 则转 Step2。

**Step6** 返回 PA、INCR 以及 true 表示计算成功。

在 UpdateX 算法的计算过程中, 在确定了节点  $i$  的父节点为节点  $j$  后, 需为节点  $i$  在  $X_i(t+1)$  所表示的中间树中的那个未确定父节点且不是源节点的祖先节点  $k$  重新选择父节点, 因为在节点  $i$  成为节点  $j$  的孩子节点后, 节点  $k$  的后代节点集发生了变化; 此外, 如果节点  $j$  在新增孩子节点  $i$  后的最优发送调度的传输功率有所增加, 还需为那些未确定父节点且可以使用节点  $j$  作为父节点的节点重新选择父节点; 对于其他每个未确定父节点的节点  $wn(d)$ , 无需为  $wn(d)$  重新选择父节点, 因为  $E_d - DES(X_m(t+1), wn(d))$  不变且  $E_d - DES(X_m(t+1), wn(d))$  中的每个节点的最优发送调度不变, 故而  $PA[d]$  和  $INCR[d]$  不会发生改变。这样可以减少 UpdatePa 算法的计算量。

#### 5.2.4 粒子适应度计算

在计算粒子  $m$  的适应度值  $fit(X_m)$  时, 首先应得到每个节点在粒子位置  $X_m$  所表示的广播树中的孩子节点集, 然后使用 ComputeNOSS 算法计算每个传输节点的最优发送调度, 粒子  $m$  的适应度值  $fit(X_m)$  为所有节点的传输功率之和。

#### 5.2.5 SCLPSODCMPBS 算法

在 S-CLPSO<sup>[24]</sup> 的计算过程中, 在每次速度更新中,  $w v_m^d(t)$  ( $w \in [0, 1]$ ) 部分使得  $v_m^d$  中的每个元素概率  $p_m^d(e)$  ( $e | p_m^d(e) \in v_m^d$ ) 减小,  $c \cdot rand() (pbx_{ep_m^d}^d(t) - x_m^d(t))$  部分使得  $v_m^d$  中的某些元素概率增大。随着迭代的进行, 粒子速度中的元素概率  $p_m^d(e)$  呈指数下

降, 经过一定次数的迭代后  $p_m^d(e)$  趋于 0,  $cut_\alpha(v_m^d)$  趋于空集, 粒子位置易于保持不变, 会导致早熟收敛。为此, 如果粒子的个体极值连续  $SG$  代都没有进化, 就对粒子速度进行随机扰动, 以避免粒子陷入停滞状态,  $SG$  称为与随机扰动粒子速度相关的粒子个体极值连续停止进化代数阈值。与文献[15]的速度变异类似, 对粒子速度  $V_m$  按如下方式进行随机扰动, 对于每一维  $v_m^d$  ( $v_m^d = \{e | p_m^d(e) | e \in E_d\}$ ), 对于  $v_m^d$  中的每个  $e | p_m^d(e)$ ,  $p_m^d(e) = \begin{cases} rand_2(0), & rand_1(0) \leq 0.5 \\ p_m^d(e), & \text{其他} \end{cases}$ 。若参数

$SG$  较大, 则虽然粒子的局部开发能力较好, 但是粒子易于陷入停滞状态, 全局探索能力较差; 若  $SG$  较小, 则粒子虽然全局探索能力较好但是局部开发能力较差。

在粒子位置得到更新后, 使用 DCShrink 算法优化粒子位置, 以提高收敛速度和求解质量。基于 S-CLPSO<sup>[24]</sup> 求解 DCMCBS 问题的 SCLPSODCMPBS 算法如下。

**Step1**  $\forall 0 \leq m \leq pss-1$ , 如果  $rand() \leq \varphi$ , 则使用 DCBIP 算法初始化粒子位置  $X_m$ , 否则随机初始化  $X_m$ 。 $\forall 0 \leq m \leq pss-1$ , 随机初始化粒子速度  $V_m$ ,  $PBX_m \leftarrow X_m$ ,  $rg_m \leftarrow 0$ ,  $dg_m \leftarrow 0$ 。 $\forall 0 \leq m \leq pss-1$ , 初始化粒子学习榜样  $EP_m$ 。 $gbest \leftarrow \operatorname{argmin}_{0 \leq m \leq pss-1} \{fit(PBX_m)\}$ 。 $t \leftarrow 0$ 。

**Step2**  $m \leftarrow 0$ , 更新惯性权重  $w$ 。

**Step3** 如果  $rg_m \geq CG$ , 则更新粒子  $m$  每一维的学习榜样  $ep_m^d$ ,  $rg_m \leftarrow 0$ 。如果  $dg_m \geq SG$ , 则随机扰动粒子速度  $V_m$ ,  $dg_m \leftarrow 0$ 。

**Step4** 按式(3)计算新粒子速度  $V_m(t+1)$ 。使用 UpdateX 算法计算新粒子位置  $X_m(t+1)$ 。如果 UpdateX 计算失败, 则  $X_m(t+1) \leftarrow X_m(t)$ , 转 Step5。如果  $X_m(t+1) = X_m(t)$ , 则转 Step5。使用 DCShrink 算法局部优化  $X_m(t+1)$ 。计算粒子  $m$  的适应度值  $fit(X_m(t+1))$ 。如果  $fit(X_m(t+1)) < fit(PBX_m)$ , 则转 Step6。

**Step5**  $rg_m \leftarrow rg_m+1$ ,  $dg_m \leftarrow dg_m+1$ , 转 Step7。

**Step6**  $PBX_m \leftarrow X_m(t+1)$ ,  $rg_m \leftarrow 0$ ,  $dg_m \leftarrow 0$ 。如果  $fit(PBX_m) < fit(PBX_{gbest})$ , 则  $gbest \leftarrow m$ 。

**Step7**  $m \leftarrow m+1$ 。如果  $m < pss$ , 则转 Step3。

**Step8**  $t \leftarrow t+1$ 。如果不满足终止条件, 则转 Step2。

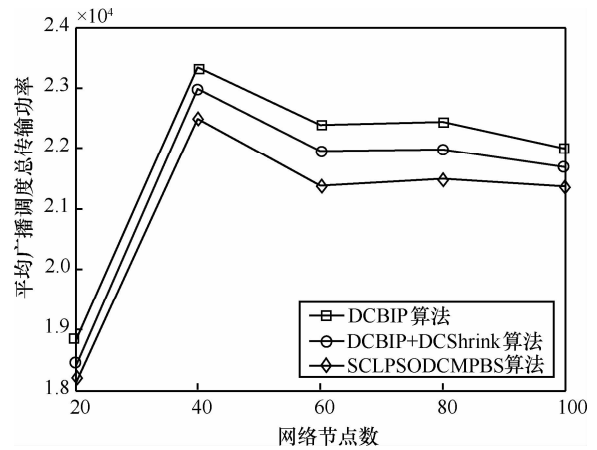
**Step9** 输出广播调度总功率  $fit(PBX_{gbest})$  及其对应的广播调度。

### 6 仿真实验结果

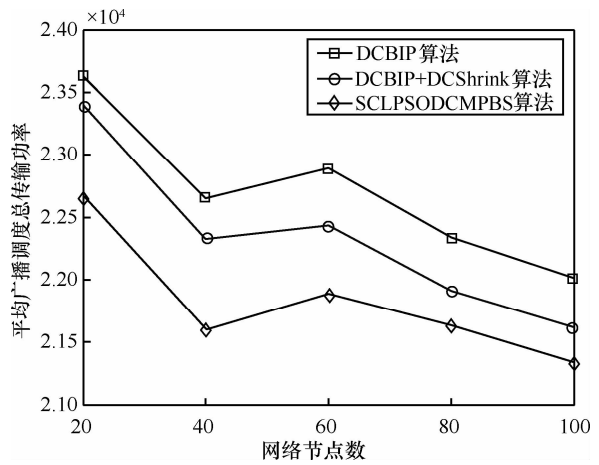
在配置了 2 GB 内存、Intel Q6600 处理器、Microsoft Windows XP 操作系统的个人计算机上开发并运行实验程序，实验程序的开发工具为基于 Java 6.0 的 MyEclipse 8.5。通过多次实验得出 SCLPSODCMPBS 算法控制参数的合适取值。

所有网络节点分布在一个  $200 \times 200$  的平面矩形区域内，工作周期(WP)的大小  $wpl=10$ 。在  $dc=0.1$ 、 $0.2$ 、 $0.3$ ，以及  $tp_{max}=50^2$ 、 $100^2$ 、 $150^2$  的情况下，对几种不同规模的网络进行实验。对于同一组( $dc$ ， $tp_{max}$ ， $ns$ )，随机生成 20 个网络，统计 DCBIP、DCBIP+DCShrink、SCLPSODCMPBS 这 3 种算法各自对这 20 个网络的平均计算结果。对于每个网络，随机产生各节点的位置，随机产生各节点的活动时隙，随机产生一个广播源节点  $s$ ，并确保在  $tp_{max}$  的限制下网络是连通的。DCBIP+DCShrink 表示先由 DCBIP 算法进行计算，然后由 DCShrink 算法对 DCBIP 算法的计算结果进行局部优化。在 SCLPSODCMPBS 算法中，加速因子  $ac=1$ ，惯性权重  $w=0.8$ ，贪心初始化概率  $\varphi=0.5$ ，与更新粒子学习榜样相关的粒子个体极值连续停止进化代数阈值  $CG=5$ ，与随机扰动粒子速度相关的粒子个体极值连续停止进化代数阈值  $SG=10$ ，粒子群规模  $pss=10$ ，终止条件为迭代 100 次。实验结果如图 2~图 4 所示。

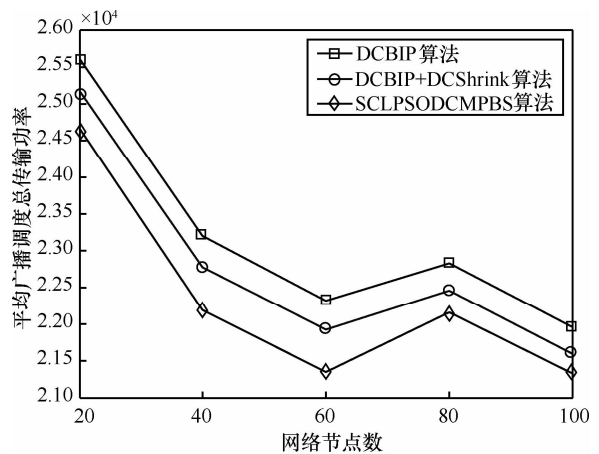
图 2~图 4 的实验结果表明，SCLPSODCMPBS 算法的计算结果优于 DCBIP 算法和 DCBIP+DCShrink 算法，这是因为 SCLPSODCMPBS 算法能够搜索到较优的广播树结构。例如，当  $dc=0.1$ 、 $tp_{max}=50^2$  时，SCLPSODCMPBS 算法的平均计算结果比 DCBIP 算法降低了 2.84%~4.54%，比 DCBIP+DCShrink 算法降低了 1.46%~2.65%；当  $dc=0.2$ 、 $tp_{max}=50^2$  时，SCLPSODCMPBS 算法的平均计算结果比 DCBIP 算法降低了 4.5%~8.34%，比 DCBIP+DCShrink 算法降低了 2.86%~4.99%；当  $dc=0.3$ 、 $tp_{max}=50^2$  时，SCLPSODCMPBS 算法的平均计算结果比 DCBIP 算法降低了 6.09%~10.81%，比 DCBIP+DCShrink 算法降低了 2.22%~6.15%。但 SCLPSODCMPBS 算法的计算时间比 DCBIP 算法和 DCBIP+DCShrink 算法要长，这是因为 SCLPSODCMPBS 算法使用多个粒子进行搜索并且进行多次迭代。



(a)  $tp_{max}=2500$



(b)  $tp_{max}=10000$



(c)  $tp_{max}=22500$

图 2 当  $dc=0.1$  时，3 种算法的平均广播调度总传输功率比较

图 2~图 4 的实验结果表明，SCLPSODCMPBS 算法对于几种不同的  $dc$  都获得了较好的优化效果。3 种算法各自的平均广播调度总传输功率在  $dc=0.2$  时比在  $dc=0.1$  时小，在  $dc=0.3$  时比在  $dc=0.2$  时小。这是因为  $dc$  越大，节点的活动时隙越

多，平均每个时隙的活动节点数越多，平均节点内部最优发送调度的传输功率越小，从而平均广播调度总传输功率越小。在  $dc$  较小的情况下， $dc$  越大，平均节点内部最优发送调度的计算时间越长，平均 SCLPSODCMPBS 算法的计算时间也就越长。

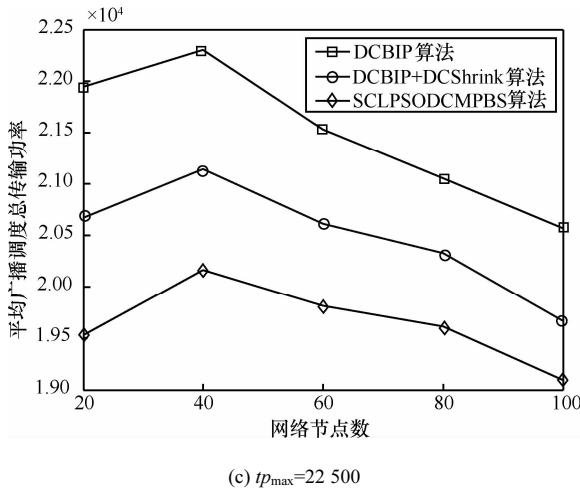
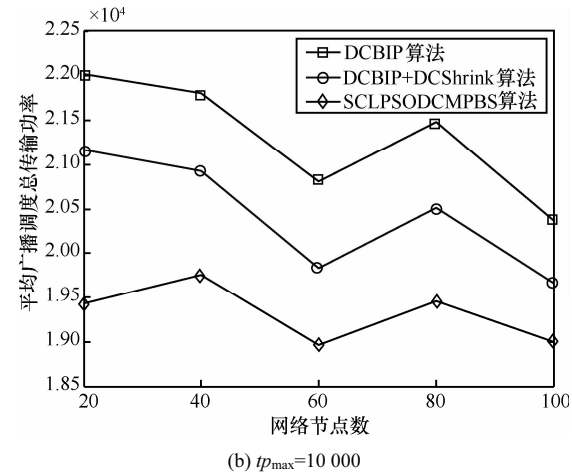
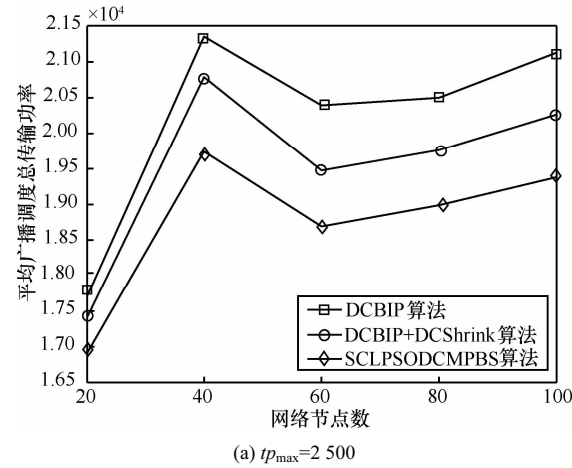


图 3 当  $dc=0.2$  时，3 种算法的平均广播调度总传输功率比较

图 2~图 4 的实验结果表明，SCLPSODCMPBS 算法对于几种不同的  $tp_{max}$  均取得了较好的计算结果。 $tp_{max}$  越大， $|E_d|$  越大，粒子位置更新失败次数越少，从而 SCLPSODCMPBS 算法的计算时间越长。

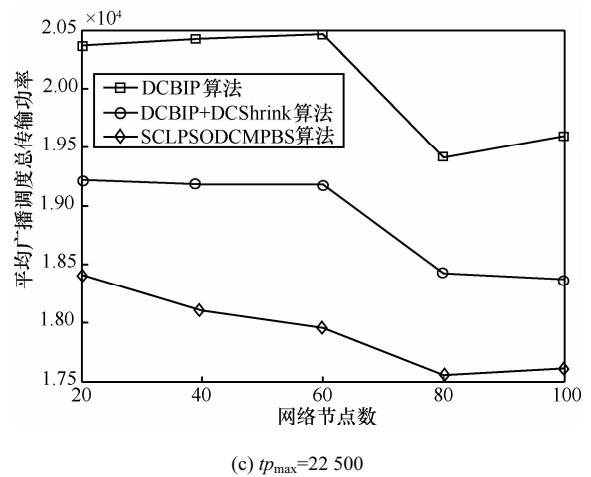
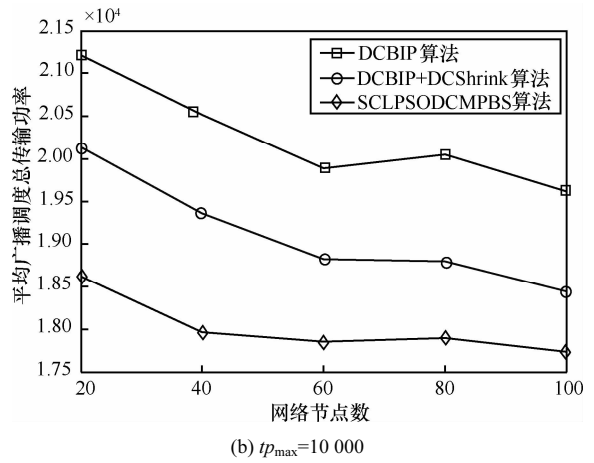
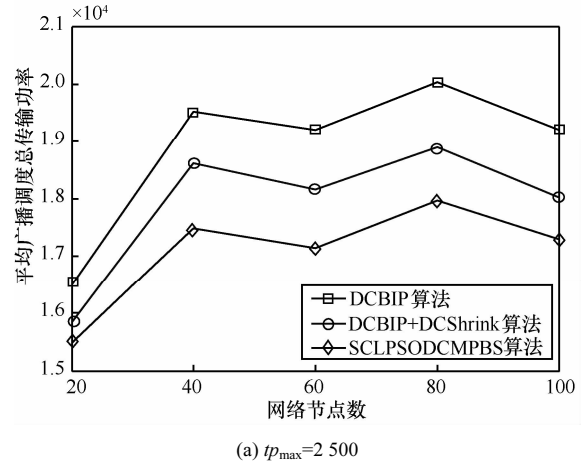


图 4 当  $dc=0.3$  时，3 种算法的平均广播调度总传输功率比较

当  $dc=1$  时，DCMPBS 算法的问题就是 MPB 问题。SCLPSODCMPBS 算法亦可以用来求解 MPB

问题。令  $LS_{GA}(q)$  表示文献[12]的局部搜索算法。 $LS_{GA}(q)$  按  $r$  从 1 递增到  $q$  的顺序进行处理, 每轮按随机顺序使用  $r$ -shrink 处理各个非叶节点, 在每轮处理结束后若广播树得到优化, 则重新从  $r=1$  开始。SCLPSODCMPBS 算法在求解 MPB 问题时使用  $LS_{GA}(q)$  算法<sup>[12]</sup>替代 DCShrink 算法对粒子位置进行局部优化, 使用 SCLPSODCMPBS- $qs$  算法表示利用  $LS_{GA}(q)$ <sup>[12]</sup>进行局部优化的 SCLPSODCMPBS 算法。为了衡量 SCLPSODCMPBS- $qs$  算法求解 MPB 问题的性能, 对文献[8]提供的基准测试实例(可从 <http://dag.informatik.uni-kl.de/research/meb/> 获得)进行计算, 对每个实例计算 30 次, 对这 30 次计算的结果统计平均超出最优解的百分比、计算出最优解的次数。同时统计 SCLPSODCMPBS- $qs$  算法的平均运行时间, 与文献[15]类似, 对于每次计算的运行时间, 若找到最优解则为找到最优解所需的时间, 否则为运行至满足终止条件所需的时间。对于节点数为 20 的基准测试实例, 使用 SCLPSODCMPBS-2s 算法进行计算, 其中,  $ac=1$ ,  $w=0.8$ ,  $\varphi=0.5$ ,  $CG=5$ ,  $SG=6$ ,  $pss=30$ , 运行终止条件为全局极值连续 50 代未进化。SCLPSODCMPBS-2s 每次都能计算出最优解, 总平均运行时间为 0.03 s。对于节点数为 50 的基准测试实例, 使用 SCLPSODCMPBS-6s 进行计算, 其中,  $ac=1$ ,  $w=0.8$ ,  $\varphi=0.5$ ,  $CG=5$ ,  $SG=6$ ,  $pss=30$ , 运行终止条件为全局极值连续 300 代未进化, 计算结果如表 1 所示。表 1 中列出的 ELS(with LS(2))、GA-2s、ACO、PSO 的数据分别取自文献[10]、文献[12]、文献[11]、文献[15]。就总平均运行时间而言, ELS(with LS(2))为 46 s<sup>[10]</sup>, GA-2s 为 13.71 s<sup>[12]</sup>, ACO 为 1.51 s<sup>[11]</sup>, PSO 为 1.45 s<sup>[15]</sup>, SCLPSODCMPBS-6s 为 9.72 s。实验结果显示 SCLPSODCMPBS-6s 算法虽然运行时间较长, 但是能够获得较优的计算结果, 表明 SCLPSODCMPBS 算法在求解 MPB 问题时具有一定的竞争力。SCLPSODCMPBS 采用的是离散型的 S-CLPSO<sup>[24]</sup>, 从而能够很好地针对 MPB 问题的离散特性, 在 S-CLPSO<sup>[24]</sup>中粒子的每一维独立地选择学习榜样, 使得 SCLPSODCMPBS 能够有效地避免早熟收敛。对于 ELS<sup>[10]</sup>、GA<sup>[12]</sup>、ACO<sup>[11]</sup>、PSO<sup>[15]</sup>、SCLPSODCMPBS 算法, 就时间复杂度而言, 每种算法的时间复杂度均主要取决于迭代次数  $iter$ 、种群规模(或后代数量) $ps$ 、局部搜索的计算时间  $tc_b(ns)$ , 因此都可以表示为  $O(iter \cdot ps \cdot tc_b(ns))$ , 每

种算法的局部搜索都是基于  $r$ -shrink 算法<sup>[6]</sup>的, 其计算时间取决于  $r$ -shrink(或改进的  $r$ -shrink)的计算时间以及对多个  $r$ -shrink 的执行策略。

## 7 结束语

本文分析了在无线自组网中当网络使用睡眠调度并且节点的传输功率连续可调时的最小功率广播调度问题。由于节点只能在活动时隙接收, 发送节点的一次传输不一定能覆盖它的所有接收节点, 因此发送节点可能需要进行多次传输, 在发送节点内部存在一个最优发送调度使得覆盖所有接收节点所需的总传输功率最小。本文提出了一个构造最小功率广播调度的 SCLPSODCMPBS 算法, 其运用一种基于集合的全面学习的离散粒子群优化算法 S-CLPSO<sup>[24]</sup>来搜索最优广播树, 并结合贪心算法和局部优化算法以提高收敛速度和求解质量。模拟实验结果表明, SCLPSODCMPBS 算法能够有效地减少广播调度的总功率。

## 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[A]. Proceedings of the IEEE INFOCOM'2000[C]. Tel Aviv, Israel, 2000.585-594.
- [2] WAN P J, CALINESCU G, LI X Y, *et al.* Minimum-energy broadcast routing in static ad hoc wireless networks[A]. Proceedings of the IEEE INFOCOM'2001[C]. Anchorage, Alaska, USA, 2001.1162-1171.
- [3] DAS A K, MARKS R J, EL-SHARKAWI M, *et al.* Minimum power broadcast trees for wireless networks: integer programming formulations[A]. Proceedings of the IEEE INFOCOM'2003[C]. San Francisco, CA, USA, 2003.1001-1010.
- [4] CAGALJ M, HUBAUX J P, ENZ C. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues[A]. Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking[C]. Atlanta, Georgia, USA, 2002. 172-182.
- [5] MIN M, PARDALOS P M. Total energy optimal multicasting in wireless ad hoc networks[J]. Journal of Combinatorial Optimization, 2007, 13(4):365-378.
- [6] DAS A K, MARKS R J, EL-SHARKAWI M, *et al.* R-shrink: a heuristic for improving minimum power broadcast trees in wireless networks[A]. Proceedings of the 2003 IEEE Global Telecommunications Conference[C]. San Francisco, CA, USA, 2003. 523-527.
- [7] LIANG W. Constructing minimum-energy broadcast trees in wireless ad hoc networks[A]. Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing[C]. Lausanne, Switzerland, 2002.112-122.
- [8] AL-SHIHABI S, MERZ P, WOLF S. Nested partitioning for the minimum energy broadcast problem[A]. Proceedings of the Second International Conference on Learning and Intelligent Optimization

表 1 对于节点数为 50 的基准测试实例各算法计算结果

实例	最优解	算法									
		ELS(with LS(2)) <sup>[10]</sup>		GA-2s <sup>[12]</sup>		ACO <sup>[11]</sup>		PSO <sup>[15]</sup>		SCLPSODCMPBS-6s	
		平均 超出率/%	计算出最优 解的次数	平均 超出率/%	计算出最优 解的次数	平均 超出率/%	计算出最优 解的次数	平均 超出率/%	计算出最优 解的次数	平均 超出率/%	计算出最优 解的次数
0	399 074.64	0.41	15/30	0.88	6/30	0	30/30	0	30/30	0	30/30
1	≤373 565.15	0.16	5/30	0.36	9/30	0	30/30	0	30/30	0	30/30
2	393 641.09	0.28	13/30	0	30/30	0	30/30	0	30/30	0	30/30
3	316 801.09	1.71	11/30	0	30/30	0	30/30	0	30/30	0	30/30
4	≤325 774.22	0.3	25/30	0.4	8/30	0	30/30	0	30/30	0	30/30
5	382 235.90	0.83	16/30	0	30/30	0	30/30	0	30/30	0	30/30
6	≤384 438.46	0	30/30	0	30/30	0	30/30	0	30/30	0	30/30
7	≤401 836.85	0.54	24/30	1.46	1/30	0	30/30	0	30/30	0	30/30
8	334 418.45	0	30/30	0	30/30	0	30/30	0	30/30	0	30/30
9	≤346 732.05	3.29	0/30	1.24	17/30	0	30/30	0	30/30	0	30/30
10	416 783.45	1.16	13/30	0	30/30	0	30/30	0	30/30	0	30/30
11	≤369 869.41	2.87	1/30	0.32	25/30	0	30/30	0	30/30	0	30/30
12	≤392 326.01	0.57	7/30	0.9	20/30	0	30/30	0	30/30	0	30/30
13	≤400 563.83	0.04	29/30	0.09	11/30	0	30/30	0	30/30	0	30/30
14	388 714.91	0.34	3/30	0	30/30	0.01	29/30	0.47	28/30	0	30/30
15	371 694.65	0.2	5/30	0	30/30	0	30/30	0	30/30	0	30/30
16	≤414 587.42	0.3	26/30	1.24	1/30	0	30/30	0	30/30	0	30/30
17	355 937.07	1.88	17/30	0.03	28/30	0	30/30	0	30/30	0	30/30
18	≤376 617.33	0.24	8/30	0	24/30	0	30/30	0	29/30	0	30/30
19	335 059.72	0	30/30	0	30/30	0	30/30	0	30/30	0	30/30
20	≤414 768.96	0.15	0/30	0.13	4/30	0.04	21/30	0	30/30	0.01	28/30
21	≤361 354.27	0	30/30	0	30/30	0	30/30	0	30/30	0	30/30
22	329 043.51	0	30/30	0	30/30	0	30/30	0	30/30	0	30/30
23	383 321.04	0	30/30	0	30/30	0	30/30	0	30/30	0	30/30
24	404 855.92	0.07	17/30	0	30/30	0	30/30	0	30/30	0	30/30
25	363 200.32	0	30/30	0	30/30	0	30/30	0	30/30	0	30/30
26	406 631.51	2.17	2/30	0.30	27/30	0	30/30	0	30/30	0	30/30
27	≤451 059.62	0.18	22/30	0.01	29/30	0	30/30	0	30/30	0	30/30
28	≤415 832.44	0.47	23/30	0.16	27/30	0	30/30	0	30/30	0	30/30
29	380 492.77	0.08	27/30	0	30/30	0	30/30	0	30/30	0	30/30
总平均		0.61	17.3/30	0.25	22.9/30	0	29.67/30	0.02	29.9/30	0	29.93/30

(LION 2007 II)[C]. Trento, Italy, 2007. 1-11.

[9] MONTEMANNI R, GAMBARELLA 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.

[10] WOLF S, MERZ P. Evolutionary local search for the minimum energy broadcast problem[A]. Proceedings of the 8th European Conference on

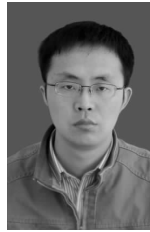
Evolutionary Computation in Combinatorial Optimization (EvoCOP 2008)[C]. Naples, Italy, 2008. 61-72.

[11] HERNANDEZ H, BLUM C. Ant colony optimization for multicasting in static wireless ad-hoc networks[J]. Swarm Intelligence, 2009, 3(2): 125-148.

[12] SINGH A, BHUKYA W N. A hybrid genetic algorithm for the minimum energy broadcast problem in wireless ad hoc networks[J]. Applied Soft Computing, 2011, 11(1):667-674.

- [13] YUAN P, JI C, ZHANG Y, *et al.* Optimal multicast routing in wireless ad hoc sensor networks[A]. Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control[C]. 2004. 367-371.
- [14] 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.  
ZHU X J, SHEN J. Minimum energy consumption multicast routing in ad hoc networks based on particle swarm optimization[J]. Journal on Communications, 2012, 33(3):52-58.
- [15] HSIAO P C, CHIANG T C, FU L C. Particle swarm optimization for the minimum energy broadcast problem in wireless ad-hoc networks[A]. Proceedings of the 2012 IEEE Congress on Evolutionary Computation[C]. Brisbane, Australia, 2012.1-8.
- [16] PALCHAUDHURI S, JOHNSON D B. Power mode scheduling for ad hoc networks[A]. Proceedings of the 10th IEEE International Conference on Network Protocols[C]. Paris, France, 2002. 192-193.
- [17] BUETTNER M, YEE G V, ANDERSON E, *et al.* X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks[A]. Proceedings of the 4th International Conference on Embedded Networked Sensor Systems[C]. Boulder, Colorado, USA, 2006. 307-320.
- [18] GU Y, HE T. Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links[A]. Proceedings of the 5th International Conference on Embedded Networked Sensor Systems[C]. Sydney, Australia, 2007. 321-334.
- [19] GUO S, GU Y, JIANG B, *et al.* Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links[A]. Proceedings of the 15th Annual International Conference on Mobile Computing and Networking[C]. Beijing, China, 2009. 133-144.
- [20] HONG J, CAO J, LI W, *et al.* Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks[J]. IEEE Transactions on Vehicular Technology, 2010, 59(1):307-318.
- [21] WANG F, LIU J. Duty-cycle-aware broadcast in wireless sensor networks[A]. Proceedings of the IEEE INFOCOM'2009[C]. Rio de Janeiro, Brazil, 2009. 468-476.
- [22] SU L, DING B, YANG Y, *et al.* OCast: optimal multicast routing protocol for wireless sensor networks[A]. Proceedings of the 17th IEEE International Conference on Network Protocols[C]. Princeton, New Jersey, USA, 2009. 151-160.
- [23] HAN K, LIU Y, LUO J. Duty-cycle-aware minimum-energy multicasting in wireless sensor networks[J]. IEEE/ACM Transactions on Networking, 2013,21(3):910-923.
- [24] CHEN W N, ZHANG J, CHUNG H S H, *et al.* A novel set-based particle swarm optimization method for discrete optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2010, 14(2):278-300.
- [25] KENNEDY J, EBERHART R. Particle swarm optimization[A]. Proceedings of the 1995 IEEE International Conference on Neural Networks[C]. Perth, Australia, 1995.1942-1948.
- [26] 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.
- [27] SHI Y, EBERHART R. A modified particle swarm optimizer[A]. Proceedings of the 1998 IEEE International Conference on Evolutionary Computation[C]. Anchorage, AK, USA, 1998. 69-73.
- [28] GONG Y J, ZHANG J, LIU O, *et al.* Optimizing the vehicle routing problem with time windows: a discrete particle swarm optimization approach[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2012, 42(2):254-267.
- [29] LIANG J J, QIN A K, SUGANTHAN P N, *et al.* Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3):281-295.

#### 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、分布对象计算、网络教学等。