

可充电无线传感器网络中的最大流算法研究

李天旭, 肖 硕

(中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116)

摘 要: 通过在传感器网络中一些节点附近部署静态辅助充电器(ACs)的方法,能提升网络中从 source 节点流向 sink 节点的最大流量. 为此,构建了该问题的混合整数线性规划模型(MILP),并证明该问题为 NP-hard 问题,提出首先使用 BottleNeck 算法为遗传算法生成初始种群,该算法以路径为单位,采用能量最低的节点优先的原则部署 ACs,然后使用改进的自适应的遗传算法(IAGA)模拟自然进化过程,搜索部署 ACs 的最优位置,使到达 sink 节点的流量达到最大. 仿真实验结果表明,与其他的几种布属 ACs 的方法相比,IAGA 可以有效提高到达 sink 节点的最大流量.

关键词: 可充电无线传感器网络;辅助充电器;最大流;遗传算法

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2018)10-0116-05

Research on Maximum flow Algorithm in Rechargeable Wireless Sensor Networks

LI Tian-xu¹, XIAO Shuo¹

(1 School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

Abstract: A method of deploying static auxiliary chargers (ACs) next to some sensor nodes is used to improve the maximum flow from sources to sinks in the network. So, the research formulates a mixed integer linear program (MILP) for the problem and proves that the problem is NP-hard. Firstly, it proposes to use BottleNeck algorithm which uses path-by-unit and deploys ACs using the lowest energy node-first principle to generate initial population for genetic algorithm. Then the Improved Adaptive Genetic Algorithm (IAGA) is used to simulate the natural evolutionary process and search for the optimal location for deployment of ACs to maximize the flow rate to sinks. The simulation results show that IAGA can effectively increase the maximum flow arriving at the sinks compared with some other algorithms of distributing ACs.

Key words: rechargeable wireless sensor networks; auxiliary chargers; maximum flow; genetic algorithm

1 引言

目前无线传感器网络中无线充电主要使用磁耦合谐振技术. 磁耦合谐振技术^[1]是一种新兴的无线充电技术,它具有充电效率高、可阻挡、远距离、全方向等特点.

文献[2]通过在瓶颈节点周围部署带有无线充电能力的 rovers,从而提高网络整体的数据传输速率,提出启发式算法 Path,用来选择合适的节点部

署 rovers. Path 首先为传感器网络分别设置了一个虚拟 source 节点和虚拟 sink 节点,然后选择这两个节点之间距离最短的路径上的节点进行部署 rovers. 这种方法虽然提高了最短路径上的数据传输速率,但在随机的网络环境中,数据并不总是通过最短路径传输,瓶颈节点依旧会对数据传输造成较大影响.

为了改善这种状况,本文首先使用 BottleNeck 算法为遗传算法生成初始种群. BottleNeck 选择在

虚拟 source 节点和虚拟 sink 节点之间的每条路径上能量最低的节点(bottleneck),进行部署静态辅助充电器(ACs),然后通过改进的自适应的遗传算法(AGA)对 BottleNeck 生成的种群进行优化。

2 网络模型与 ACs 部署问题

2.1 网络数据传输能耗模型

首先,将无线传感器网络建模为有向图 $G = (V, E)$, $\langle i, j \rangle \in E$ 表示从节点 i 到节点 j 的有向边, V 是图中所有节点的集合, E 是有向边 $\langle i, j \rangle$ 的集合, $i, j \in V$. S 表示所有 source 节点的集合, T 表示所有 sink 节点的集合. $T_G^-(i) \subset V$ 表示向节点 i 发送数据的节点集合, $T_G^+(i) \subset V$ 表示接收节点 i 发送的数据的节点集合. δ_i 表示节点 i 的度, 且 $\delta_i = T_G^-(i) + T_G^+(i)$.

节点 i 从自然界中采集能量的功率为 α_i , 若在节点 i 周围部署 ACs, ACs 对节点进行充电的功率为 β , ACs 的充电效率为 λ , 那么节点 i 通过这两种充电方式获得的总的充电功率为 $\alpha_i + \lambda\beta$. 使用 f_{ij} 表示数据在有向边 $\langle i, j \rangle$ 上的传输速率, e_{ij} 表示节点 i 接收 1 bit 数据, 并将其经过有向边 $\langle i, j \rangle$ 传输给节点 j 所消耗的能量,

$$e_{ij} = \tau_{ij} + \rho, \quad (1)$$

式中, τ_{ij} 表示将 1 bit 数据, 从节点 i 传输到节点 j 所消耗的能量; ρ 表示节点 i 接收 1 bit 数据消耗的能量. 使用简单无线链路模型^[3] 可对 τ_{ij} 进行如下表示:

$$\tau_{ij} = \tau_{dec} + \tau_{amp} d_{ij}^v, \quad (2)$$

式中, τ_{dec} 表示节点发射电路每发送 1 bit 数据消耗的能量; $\tau_{amp} d_{ij}^v$ 表示信号放大器为了达到信号接收节点可接受的信噪比, 每发送 1 bit 数据需要消耗的能量; τ_{amp} 表示信号放大电路的能耗系数; d_{ij} 表示节点 i 和节点 j 之间的物理距离; v 用于计算路径损耗.

使用 D 表示所有部署了 ACs 的节点的集合, D_i 表示节点 i 周围是否有 ACs 部署, 且 $D_i \in \{0, 1\}$. 若 $D_i = 1$, 则表示节点 i 周围有 1 个 ACs 部署.

2.2 ACs 部署问题模型

假设共有 N 个 ACs 可供部署, 本文主要研究通过在有向图 $G = (V, E)$ 中选择 N 个传感器节点, 在每个节点周围部署 ACs, 使得流向 sink 节点的流量达到最大. 该问题可使用混合整数线性规划(MILP)模型, 其中包括 4 个限制: 能量限制、流量保护限制、链路容量限制、ACs 数量限制.

能量限制保证了节点传输数据时消耗的能量小

于该节点从自然界或 ACs 中获得的能量. 假设传感器节点只有在数据传输时消耗能量^[4], 则对于任意的节点 $i \in \{V - T\}$, 满足:

$$\sum_{j \in T_G^-(i)} f_{ij} e_{ij} \leq D_i \lambda \beta + \alpha_i \quad (3)$$

式中, 左端为节点 i 传输数据时总的能量消耗, 右端为节点 i 可使用的能量. 当有 1 个 ACs 部署在节点 i 附近为其充电, 则节点 i 获得的总的能量为 $D_i \lambda \beta + \alpha_i$.

对节点进行流量保护, 节点必须将接收到的数据全部进行转发, 对于任意的节点 $i \in \{V - T - S\}$, 满足:

$$\sum_{j \in T_G^-(i)} f_{ji} = \sum_{j \in T_G^+(i)} f_{ij} \quad (4)$$

链路容量 C , 同样会对节点间数据传输造成限制, 对于任意有向边 $\langle i, j \rangle \in E$, 满足:

$$f_{ij} \leq C \quad (5)$$

在网络中可进行部署的 ACs 数量共有 N 个, 则有:

$$\sum_{i \in V-T} D_i = N \quad (6)$$

目标函数为求流向所有 sink 节点的流量和的最大值, 可以得到 MILP:

$$\begin{aligned} \max & \sum_{i \in T} \sum_{j \in T_G^-(i)} f_{ij}, \\ \text{s. t.} & (3)(4)(5)(6) \end{aligned}$$

该问题的决策变量包括 f_{ij} , e_{ij} , D_i , 解决该问题的关键在于在网络中搜索合适的节点, 进行 ACs 的部署. 当网络中的搜索空间变大时, 很难通过 MILP 的求解获得结果. 背包问题为经典的 NP-hard 问题, 可以通过该问题和背包问题的对比, 证明该问题为 NP-hard 问题.

定义 1 ACs 部署问题为 NP-hard 问题.

证明 背包问题试图寻找从给定的物品集合 P 中寻找一些物品组成集合 Q 放进背包, 背包能够承受的重量上限为 W , 物品 i 的价值为 v_i , 重量为 w_i , 如何选择物品组成 Q , 最大化 $\sum_{i \in Q} v_i$, 并且 $\sum_{i \in Q} w_i \leq W$. 首先为 $G = (V, E)$ 添加虚拟的 source 节点 s' 和 sink 节点 t' , s' 连接所有 source 节点, 起点为 s' , 终点为 S 中的所有节点的有向边, 组成集合 E_s . 所有的 sink 节点连接 t' , 起点为 T 中的所有节点, 终点为 t' 的有向边, 组成集合 E_T . E_s, E_T 中的所有有向边具有无限的链路容量. $V' = V \cup s' \cup t'$, $E' = E \cup E_s \cup E_T$, 构成 $G'(V', E')$ ^[5]. 假设从 s' 到 t' 共 k 条互不

相交的路径 $p_1, p_2, p_3, \dots, p_k$, 对于任意 $1 \leq i \leq k$, 在 p_i 上部署的 ACs 的数量为 $D(p_i)$, 到达该路径上 sink 节点的流量为 $f(p_i)$, 如何选择节点部署 ACs, 最大化 $\sum_{i=1}^k f(p_i)$, 并且 $\sum_{i=1}^k D(p_i) \leq |D|$. 可以看出该问题等价于背包问题, 因此该问题为 NP-hard 问题.

3 ACs 部署问题解决方案

本文提出的解决方法为首先使用启发式算法 BottleNeck 生成初始种群, 然后使用 IAGA 对初始种群进行优化.

3.1 BottleNeck 算法

从 source 节点到 sink 节点的每条路径上的瓶颈节点, 是阻碍无线传感器网络数据传输的主要原因. 为此本文提出 BottleNeck 算法, 为遗传算法生成较为优质的初始种群.

BottleNeck 算法实现细节如算法 1 所示: 输入为 $G'(V', E')$. 1—2 行, 首先定义变量 k 用来表示路径的编号, 变量 AC 表示 ACs 的数量, AC 的初始值为 N. 第 3 行, 使用 $\text{CalculateAllPaths}(G'(V', E'), \&n)$ 函数求解从虚拟节点 s' 到 t' 之间的所有路径, 存入向量 P 中, 并将路径的数量写入变量 n. 4—17 行, 开始进行 ACs 的部署. 5—7 行, 首先判断是否已经完成所有路径上最小能量节点 bottlenode 的搜索, 若已完成, 则将 k 赋值为 1, 重新搜索. 第 8 行, $\text{searchBottleNode}(P[k])$ 在满足 (1)、(2)、(3) 条件下, 搜索路径 P[k] 上的 bottlenode. 9—16 行, 将找到的 bottlenode, 放入到集合 D 中. 首先需要判断 $\text{searchBottleNode}(P[k])$ 找到的 bottlenode 是否已经存在于集合 D 中, 若已经存在, 说明该节点已经部署过 ACs, 那么继续寻找下一条路径上的 bottlenode. 否则, 将该节点放入集合 D 中, 同时将该节点的能量增加 β , 并调整 ACs 的数量. k 的值需要增加 1, 以便于寻找下一条路径上的 bottlenode. 循环结束的条件为所有 ACs 已部署完毕, 所有部署了 ACs 的节点组成集合 D. 第 18 行, $\text{InitGroup}(G'(V', E'), D)$ 使用 D 生成初始种群 G, 实现方法如下: 设 $G = \{g_1, g_2, g_3, \dots, g_{|G|}\}$, $g_k (1 \leq k \leq |G|)$ 表示第 k 个个体, 即网络中所有部署 ACs 的节点的集合, 随机从 $D, \{V - T - D\}$ 中分别选择节点 i, j , 则 $g_k = \{D - i\} \cup j$.

算法 1 BottleNeck

INPUT: $G'(V', E')$

OUTPUT: G and D

1 $k = 1$;

2 $AC = N$;

3 $P = \text{CalculateAllPaths}(G'(V', E'), \&n)$;

4 While $AC > 0$

5 If $k > n$ then

6 $k = 1$;

7 END

8 $\text{bottlenode} = \text{searchBottleNode}(P[k])$;

9 If $\text{bottlenode} \in D$ then

10 $k++$;

11 Else

12 $D = D \cup \text{bottlenode}$;

13 $E(\text{bottlenode}) = E(\text{bottlenode}) + \beta$;

14 $AC--$;

15 $k++$;

16 END

17 END

18 $G = \text{InitGroup}(G'(V', E'), D)$;

3.2 IAGA 算法

遗传算法^[6] 是使用计算机模拟生物的遗传和进化, 进行全局最优搜索的智能算法. 传统的遗传算法由于采用固定的交叉概率 P_c 、变异概率 P_m , 存在较为严重的早熟现象, 即陷入局部最优解. 自适应的调节 P_c, P_m 的大小, 可以有效避免早熟现象的发生. 研究中采用如下的规则^[7] 自适应的改变 P_c, P_m 的大小:

$$P_c = \begin{cases} k_1 (f_{\max} - f') / (f_{\max} - f_{\text{ave}}) & f' \geq f_{\text{ave}} \\ f' < f_{\text{ave}} \end{cases} \quad (7)$$

$$P_m = \begin{cases} k_3 (f_{\max} - f) / (f_{\max} - f_{\text{ave}}) & f \geq f_{\text{ave}} \\ k_4 & f < f_{\text{ave}} \end{cases} \quad (8)$$

式中, f' 为两个交叉个体中适应度较大的个体的适应度; f 为变异个体的适应度, $0 < k_1, k_2, k_3, k_4 \leq 1$, $f_{\max}$ 为当前种群中的最大适应度, f_{ave} 为当前种群的平均适应度. 从上式可以看到: 当交叉或变异个体的适应度大于种群的平均适应度时, 就自适应的调整 P_c 和 P_m 的大小. 反之, 则给予 P_c 和 P_m 一个较大的值 (k_2, k_4).

IAGA 的实现细节如算法 2 所示: 输入为 BottleNeck 的输出结果 G 和进化代数 GENS. 1—2 行, 初始化进化的代数, $\text{CalculateFitness}(G)$ 在 (1)、(2)、(3) 限制下使用 Ford-Fulkerson 算法^[8] 计算出 G 中每一个个体的最大流, 并返回所有最大流中的

最大值,赋值给 $f_{\max}$. 3—11 行,开始进行进化操作. 第 4 行 Crossover(&G),使用轮盘赌算法选出两个待交叉的个体,在(4)的限制下进行单点交叉,根据(5)修改 P_c 的值. 第 5 行, Mutate(&G) 在(4)的限制下进行变异操作,并根据(6)修改 P_m 的值. 6—9 行,对 $f_{\max}$ 的值进行更新. 第 10 行,修改变量 N 的值,进入下一次迭代. 循环结束的条件为已完成所有迭代,输出为到达虚拟节点 t' 的最大流 $f_{\max}$.

```
算法 2 IAGA
INPUT:  $G$  and  $GENS$ 
OUTPUT:  $Max\ Flow\ f_{max}$ 
1  $N = GENS$ ;
 $f_{max} = CalculateFitness(G)$ ;
3 While  $N > 0$ 
4   Crossover(&G);
5   Mutate(&G);
6    $Temp = CalculateFitness(G)$ ;
7 IF  $Temp > f_{max}$  then
 $f_{max} = Temp$ ;
9 END
10  $N \leftarrow$ ;
11 END
```

4 仿真实验结果

仿真实验平台为 Matlab. ACs 的充电效率 λ 为 60%. 节点 i 发送 1 bit 数据消耗的能量为 $\tau_{ij} = 200$ nJ/bit + $d_{ij}^2 \cdot 0.01$ nJ/bit/m²,为了避免传感器节点之间的距离对实验结果的影响,假设节点均匀的分布在网络中,将 d_{ij} 设置为 50 m,则 $\tau_{ij} = 225$ nJ/bit. 具体试验参数如表 1 所示.

表 1 仿真参数	
参数	值
链路容量 C	250 kbps
从自然界中进行能量采集的功率 α_i	0~15 mW
ACs 充电的功率 β	50 W
ACs 的充电效率 λ	60%
节点 i 发送 1 bit 数据消耗的能量 τ_{ij}	225 nJ/bit
节点 i 接收 1 bit 数据消耗的能量 ρ	220 nJ/bit
k_1, k_2	1.0
k_3, k_4	0.5
种群数量 $ G $	100
染色体长度	$ V-T $
迭代次数 GENS	200

实验中 BottleNeck 算法部署 ACs 后得到的结

果 D,可以通过 Ford - Fulkerson 算法求得到达 s' 的最大流,用来和 IAGA 的结果进行对比.

除了同 Path 进行比较外,实验中设计了其他三种部署 ACs 的算法同 BottleNeck 和 IAGA 进行对比:

- (1)Random:将 $|D|$ 个 ACs 随机部署在无线传感器网络中;
- (2)Lower:网络中不部署 ACs;
- (3)Upper:网络中每一个节点附近均部署 ACs.

仿真实验主要研究网络中的节点个数 $|V|$, ACs 的数量 $|D|$ 、节点的度 δ 以及 source 节点的数量 $|S|$ 对最大流的影响. 每次实验只改变其中一个变量,其他三个变量保持不变. 每次实验运行每个算法 100 遍,每次从集合 S 中随机选择 source 节点发送数据^[9,10].

4.1 节点个数 $|V|$ 对最大流的影响

将 $|V|$ 依次设置为 100、120、140、160、180, ACs 的数量 $|D|$ 设置为 15,节点的度 δ 设置为 5,source 节点的数量 $|S|$ 为 10.

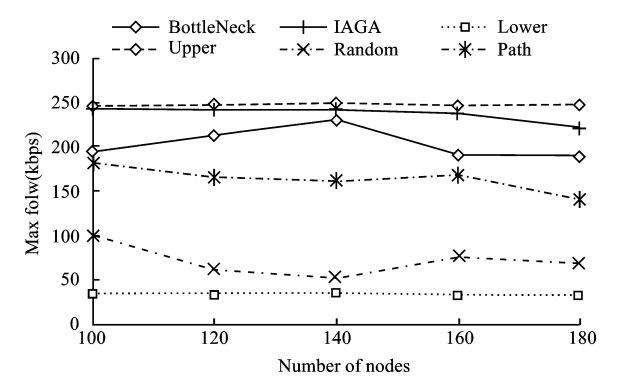


图 1 Nodes 对最大流的影响

从图 1 中可以看出,当 $|V|$ 从 100 增长到 180 时,IAGA 从 242.5 kbps 下降到 220.8 kbps,下降了 8.9%. 当 $|V|$ 从 100 增加到 120 时,BottleNeck 增长了 18.6%,但很快随着 $|V|$ 的增加,开始出现下降趋势,当 $|V| = 180$ 时,相对于 $|V| = 120$ 时,下降了 17.9%,相对于 $|V| = 100$ 时,下降了 2.7%. Path 下降了 22.3%. 出现这种下降现象的原因是由于随着节点数量的增加,部署有 ACs 的节点的比重开始下降,意味着会出现更多的瓶颈节点没有足够的能量传输数据.

4.2 ACs 的数量 $|D|$ 对最大流的影响

依次将 $|D|$ 取值为 5、10、15、20、25,节点的数量 $|V|$ 取值为 160,节点的度 δ 设置为 5,source 节点的数量 $|S|$ 为 10.

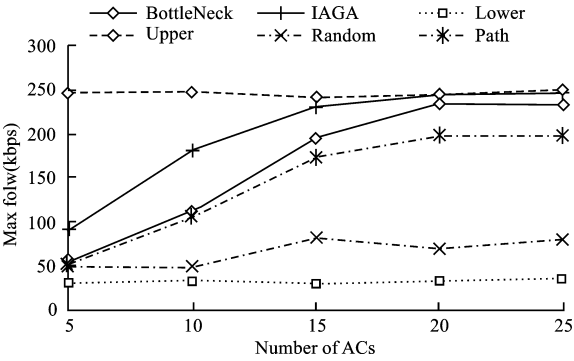


图 2 ACs 对最大流的影响

从图 2 可以看出,当 ACs 的数量从 5 增加到 25 时,BottleNeck 增长了 334.8%,IAGA 增长了 176.3%,同时可以看到,当 ACs 的数量从 10 增加到 20 时,IAGA 和 BottleNeck 的间隔缩小了 528.6%.同时,Path 和 BottleNeck 之间的间隔不断增大.ACs 的增多,导致每条数据传输路径上的瓶颈节点数量开始减少,流向 sink 节点的流量开始增加.因此相对于没有 ACs 部署的 Lower,在网络中进行随机部署的 Random 的最大流量也会得到提高.

4.3 节点的度 δ 对最大流的影响

将 δ 依次设置为 3、4、5、6、7, ACs 的数量 $|D|$ 取值为 10,节点的数量 $|V|$ 取值为 160,source 节点的数量 $|S|$ 为 10.

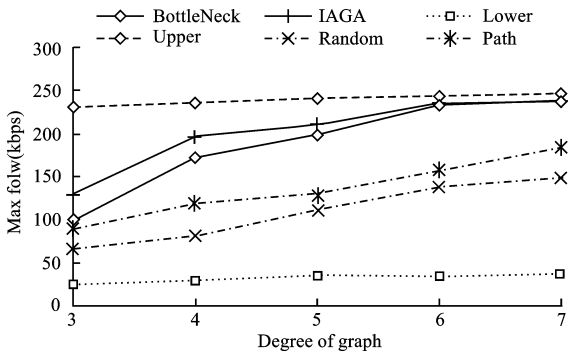


图 3 Degree 对最大流的影响

从图 3 中可以看到,随着网络中节点的度 δ 的增加,BottleNeck, IAGA 的最大流量分别增加了 141.5%、83.6%,Random 增大了 2.27 倍.对于这种现象的产生,是由于当 δ 较小时,每个节点周围的邻居节点较少,导致邻居节点中存在部署了 ACs 的节点的概率较低,同时瓶颈节点在邻居节点中存在的概率较高.而当 δ 增加时,邻居节点增多,邻居节点中出现部署了 ACs 的节点概率增加.由限制(1)可知,节点的能量决定了其数据传输速

率,因此到达 sink 节点的最大流随着 δ 的增加,不断增大.

4.4 source 节点的数量 $|S|$ 对最大流的影响

将 $|S|$ 依次设置为 10、15、20、25、30,节点的度 δ 设置为 5, ACs 的数量 $|D|$ 取值为 10,将节点的数量 $|V|$ 取值为 160.

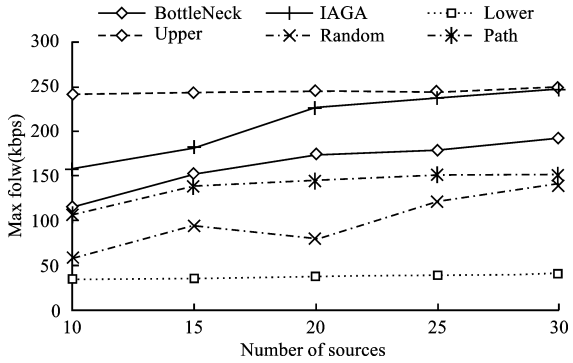


图 4 Sources 对最大流的影响

从图 4 中可以看到,随着 source 节点数量的增加,IAGA, BottleNeck, Path, Random 最大流量分别增加了 57.8%、69.1%、43.18、148.6%.这种现象出现的原因可以解释为,source 节点和 sink 节点由于路径中的某个瓶颈节点的制约而导致数据传输速率较低,当 source 节点数量不断增加,可能在该路径瓶颈节点和 sink 节点之间出现某个新的 source 节点,这时瓶颈节点对这个 source 节点的数据传输就不会产生影响.

5 结束语

新一代可充电无线传感器网络,可以很好地解决传感器网络中的瓶颈节点问题.本文通过在节点周围部署 ACs 来提高从 source 节点流向 sink 节点的最大流量.在 MILP 模型的基础上,提出了使用 BottleNeck 算法生成初始种群,然后利用 IAGA 对种群进行优化的方法.仿真结果显示,IAGA 有效提高了到达 sink 节点的最大流量,同时可以看出,不同的网络拓扑环境下,算法的性能会有较大差异,需要根据应用场景综合考虑,这些工作还有待于进一步完善.

参考文献:

[1] Kurs A, Karalis A, Moffatt R, et al. Wireless Power Transfer via Strongly Coupled Magnetic Resonances [J]. Science, 2007, 317,83-86.

- of Grid Computing, 2013, 11(4):633-651.
- [3] Arabnejad V, Bubendorfer K. Cost Effective and Deadline Constrained Scientific Workflow Scheduling for Commercial Clouds[C]. In International Symposium on Network Computing and Applications. IEEE, 2016:106-113.
- [4] Arabnejad V, Bubendorfer K, Ng b, et. al, A deadline constrained critical path heuristic for cost-effectively scheduling workflows[C]. In 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing, IEEE, 2015: 242-250.
- [5] Zeng L, Veeravalli B, Li X. Scalestar: Budget conscious scheduling precedence-constrained many-task workflow applications in cloud[C]. In 2012 IEEE 26th International Conference on Advanced Information Networking and Applications, 2012:534 - 541.
- [6] Lin X, Wu C Q. On Scientific Workflow Scheduling in Clouds under Budget Constraint [C]. International Conference on Parallel Processing. IEEE, 2013: 90-99.
- [7] Wu C Q, Lin X, Yu D, et al. End-to-End Delay Minimization for Scientific Workflows in Clouds under Budget Constraint[J]. IEEE Transactions on Cloud Computing, 2015, 3(2):169-181.
- [8] Mao M, Humphrey M. Scaling and Scheduling to Maximize Application Performance within Budget Constraints in Cloud Workflows[C]. In International Symposium on Parallel & Distributed Processing. IEEE, 2013:67-78.
- [9] Juve G, Chervenak A, Deelman E, et al. Characterizing and profiling scientific workflows [J]. Future Generation Computer Systems, 2013, 29(3):682-692.

作者简介:

吴文兵 男, (1968—), 男, 博士, 副教授. 研究方向为云计算调度及其应用. E-mail: wenbing808@163. com.

(上接第 120 页)

- [2] He T, Chin K W, Soh S. On using Wireless Power Transfer to increase the max flow of Rechargeable Wireless Sensor Networks[C]// IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing. IEEE, 2015:1-6.
- [3] Rappaport T S. Wireless Communications: Principles and Practice[M]. Prentice Hall PTR, 1996.
- [4] Xiao Y, Chen H, Li F H. Handbook on Sensor Networks[M]. World Scientific, 2010.
- [5] Ahuja R K, Magnanti T L, Orlin J B. Network flows: theory, algorithms, and applications[J]. Journal of the Operational Research Society, 1993, 45 (11): 791-796.
- [6] Goldberg D E. Genetic Algorithm in Search, Optimization, and Machine Learning[J]. 1989, xiii(7):2104 - 2116.
- [7] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms [J]. IEEE Transactions on Systems Man & Cybernetics, 2002, 24(4):656-667.
- [8] Ford L R J, Fulkerson D R. A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem[J]. Canadian Journal of Mathematics, 1955, 9:210-218.
- [9] Zhu C, Liu K, Yu C, et al. Simulation and experimental analysis on wireless energy transfer based on magnetic resonances[C]// Vehicle Power and Propulsion Conference. IEEE, 2008:1-4.
- [10] Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks [C]// Hawaii International Conference on System Sciences. IEEE Computer Society, 2000:4-7.

作者简介:

李天旭 男, (1995-), 硕士研究生. 研究方向为无线传感器网络、无线充电. E-mail: tianxuli@cumt. edu. cn.

肖 硕 男, (1981-), 博士, 副教授. 研究方向为无线传感器网络.