

传感器网络中多移动sink节点的路径规划算法

俸皓^{1,2}, 罗蕾¹, 董荣胜², 王勇²

(1. 电子科技大学计算机科学与工程学院 成都 611731; 2. 桂林电子科技大学广西自动检测技术与仪器重点实验室 广西 桂林 541004)

【摘要】考虑多移动sink且路径端点在圆周边界上的情形, 将此抽象为一个混合优化问题, 该优化问题具有维数高和搜索空间大的特点, 经典的算法(如 k -splitour算法)无法针对其连续分量进行优化, 为此该文首先以 k -splitour算法获得 k 条子路径并设计了消除子路径交叉的方法, 以获得对离散分量的局部寻优, 再通过设计对连续分量的局部优化方法以确定每个通信圆盘上访问点的位置, 从而可以高效地获取多个sink移动节点的规划路径解。给出了算法结果的上界及其理论证明。最后通过实验验证了所设计的模型及其求解算法能有效地解决数据集中的路径规划问题。

关键词 近似算法; k -TSPN; 多移动sink; 无线传感器网络

中图分类号 TP393 **文献标志码** A **doi**:10.3969/j.issn.1001-0548.2016.02.017

Tour Planning in Wireless Sensor Networks for Multi-Mobile Sinks

FENG Hao^{1,2}, LUO Lei¹, DONG Rong-sheng², and WANG Yong²

(1. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731;

2. Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology Guilin Guangxi 541004)

Abstract This paper considers the situation where multi-mobile sinks and path endpoints are located along the edge of the circumference, and abstracts it as a hybrid optimization problem characterized in high dimensionality and large searching space. Classic algorithms like the k -splitour algorithm cannot optimize its continuous variables. This paper first obtains k sub-paths by adopting k -splitour algorithm and designs the method to eliminate the crossing of sub-paths to acquire local optimum for discrete variables. Then the algorithm acquires multi-mobile sinks path planning results efficiently by designing local optimization methods for continuous variables to decide on the location of access points on each communication disk. The upper bound of the algorithm and its theoretical proof are presented. The experiments show the effectiveness of both the designed model and its algorithm in solving the path planning problem in data collection.

Key words approximate algorithm; k -TSPN; multi-mobile sinks; wireless sensor networks

数据采集是无线传感器网络的重要技术之一。传统的数据采集模式首先依靠节点的自组织特性构建出完全连通的通信网络, 收集到的感知数据依据路由协议从感知区域传输到处理区域, 从而完成感知数据的传输^[1]。在实际应用中, 经常会存在如下因素影响感知数据的采集: 1) 对于大范围的目标监控应用, 采用传感器节点随机抛洒的方式进行传感器节点的部署, 有可能形成不能完全连通的传感器网络^[2]; 2) 由于地理位置或物理条件的限制、节点的能量耗尽、突发灾难性事件等原因, 传感器网络往往会主动或被动的演化为若干个互不连通的子网络, 从而影响数据采集和传输任务^[3]; 3) 各传感器节点因为能量受限和频繁的数据转发, 会形成感知

区域的覆盖空洞, 从而降低整个传感网络的可用性^[4]。

为了提高无线传感器网络的可用性、减少节点的通信能量消耗及平衡节点的负载, 近年来研究者引入了具备可控移动能力的sink节点代替传统传感器网络中的固定sink节点实施数据收集任务^[5]。在各种基于移动sink的数据采集方案中, 单跳数据采集的方式由于能够最大限度地减少传感器节点的通信能量消耗和平衡节点的负载, 且通信链路质量能够得到更好的保证, 从而得到了广泛的关注和应用^[6]。如何获取移动sink的最短路径, 以降低网络的数据采集延迟及移动sink的能耗, 一直是研究中的关键问题^[5]。在考虑了传感器节点的通信范围之后, 问题可建模为带有圆形邻域且邻域可相互重叠的带邻

收稿日期: 2015-07-14; 修回日期: 2016-02-26

基金项目: 国家科技重大专项(2014ZX03002001); 国家自然科学基金(61363070, 61163058); 广西自然科学基金(2014GXNSFAA118370)

作者简介: 俸皓(1978-), 男, 博士生, 主要从事无线传感器网络、嵌入式实时软件等方面的研究。

域的旅行商问题(traveling salesman problem with neighborhoods, TSPN), 该问题已经被证明是APX-Hard的^[7]。在实际应用之中, 由于节点的数目往往比较多, 低时间复杂度的近似算法更加受到重视。近似算法可分为旅行商问题(traveling salesman problem, TSP)后置型和TSP前置型两类。TSP后置算法^[8-10]首先获取各个近邻区域的访问点, 然后再计算这些访问点的TSP路径。对邻域不重叠的特殊情况, 可取得比较好的效果, 但当存在重叠邻域时, 由于很难评价每次选择的访问点对最终路径的影响程度, 所以此时TSP后置型算法的效果并不理想。在随机部署的无线传感器网络中, 通信区域的重叠往往不可避免, 因此这类算法在随机部署的传感器网络中应用较少。TSP前置算法先使用通信区域的中心构建一条TSP路径, 再以此为基础进行路径优化, 这种方法的优点是便于采用渐进式的优化算法且解的效果便于评价, 但没有讨论访问点在圆盘上的情形^[11-12]。文献[11-12]将路径优化问题建模为“标签覆盖”的问题, 采用动态规划算法在 $O(n^3)$ 的时间复杂度内得到TSP路径中圆心之间的捷径, 但该方法只考虑了子路径上的端点是某个通信区域的中心的情形, 没有考虑端点可以在圆周边界的情况, 这限制了该算法性能的提升, 而且这样对模型的简单化处理是以增加移动sink的延时为代价的。文献[13]提出了一种 $O(n^2)$ 时间复杂度的近似算法, 该算法首先应用TSP路径不自交的特性, 构建一条赛道, 随后在赛道上采用内圈启发式、弯道启发式搜索策略得到近似最短路径, 但是该算法只支持单个移动sink的情形。在文献[14]提出的CSS算法之中, 以TSP路径为基础, 首先采用确定性的最小圆覆盖算法减少路径中访问区域的数目, 再使用路径覆盖和二分搜索进一步对路径进行优化, 时间复杂度为 $O(n^2(\log n + \log(1/\delta)))$, δ 是一个足够小的常数, 用于在二分搜索算法中判断两个点是否足够相近。但该算法也是只对多移动sink提供支持。从目前能够查到的文献来看还没有考虑多个移动sink节点且路径端点在圆周边界上的模型及其近似求解算法的讨论。

本文将数据采集问题中基于多移动sink且路径端点在圆周边界上的节点路径规划问题抽象为一个复杂的离散与连续混合优化问题, 该优化问题具有维数高和搜索空间大难于求解的特点, 经典的 k -splitour算法无法针对连续分量进行优化, 而且会产生交叉的不可行子路径解, 为此本文首先通过 k -splitour算法获得 k 条子路径, 并设计了顺序交换的

局部寻优方法, 再通过贪婪路径覆盖与近似梯度下降算法确定每个通信圆盘上访问点的位置以达到对连续分量的局部寻优, 从而可以高效地获取近似最优的多个sink移动节点的规划路径解。

1 网络模型和问题描述

假设二维平面上部署有 n 个传感器节点, 记为 $\{(v_0, r_0), (v_1, r_1), \dots, (v_i, r_i), \dots, (v_n, r_n)\}$, v_i 表示第 i 个传感器的位置, 其通信范围建模为圆形区域, r_i 表示对应的通信半径, 其中 v_0 为基站位置, $r_0=0$ 。假设给定有 k 个移动sink节点分别由基站出发, 每个移动sink可以沿着各自事先规划好的子路径依次经过部分传感器节点的通信范围, 完成数据收集后返回基站, 记第 i 个移动sink需要访问的传感器节点数目为 m_i 。将第 i 条子路径第一次进入到路径上第 j 个传感器节点的通信范围的位置称为节点 j 上的访问点, 记作 $p_{\pi_j}^i$ 。在各种可能的子路径规划方案中, 让各个子路径长度中最大的取值最小, 减少移动sink的遍历时间, 从而降低数据采集的延时。由此多移动sink路径规划问题可表示为:

$$\min_{i=1,2,\dots,k} (\max (d(v_0, p_{\pi_1}^i) + \sum_{j=1}^{m_i-1} d(p_{\pi_j}^i, p_{\pi_{j+1}}^i) + d(p_{\pi_{m_i}}^i, v_0))) \quad (1)$$

式中, $d(p, q)$ 表示平面上两点之间的距离; 访问点 $p_{\pi_j}^i$ 总是位于以第 i 个传感器节点为圆心、以 r_i 为半径的通信圆盘上。传感器节点 v_{ji} 的访问点 $p_{\pi_{j_i}}^i$ 可以表示为 $\langle x_{ji} + r_{ji} \cos \theta_{ji}, y_{ji} + r_{ji} \sin \theta_{ji} \rangle$, 其中 $\langle x_{ji}, y_{ji} \rangle$ 表示第 i 个传感器节点 v_{ji} 的位置坐标, 则式(1)要优化的变量就是 $v_{j1}, v_{j2}, \dots, v_{jn}$ 和 $\theta_{j1}, \theta_{j2}, \dots, \theta_{jn}$ 。对离散变量 $v_{j1}, v_{j2}, \dots, v_{jn}$ 的优化是将其分成 k 组, 每组中的整数都取一个排列, 相当于节点指派后的一个排列; 对 $\theta_{j1}, \theta_{j2}, \dots, \theta_{jn}$ 的优化是要在值域 $[0, 2\pi)$ 范围取合适的一个方位角。

2 多移动sink路径规划问题的近似算法

上节式(1)中的子路径如何指派以及各个访问点的位置如何确定均需要优化。为此本文将问题(1)分解为两个阶段解决: 首先基于 k -splitour解决多个移动sink的路径指派问题, 并设计了顺序交换的局部寻优方法, 然后提出了贪婪近似梯度下降搜索算法解决访问点的位置优化问题, 具体的算法设计步骤与分析如下。

2.1 k 个移动sink路径指派问题的求解

文献[15]提出的 k -splitour算法可以在 $O(n)$ 的时

间复杂度之内获得 k 条TSP路径, 算法的近似比为 $\alpha+1-1/k$, α 是所采用的1-TSP算法的近似比。通过对 k -splitour算法的分析可发现, 每条子路径的起始边和结束边, 可能与该子路径之前的边产生相交。依据三角不等式, 最优TSP路径中是不存在自相交的边的, 因此设计了顺序交换的方法对该子路径去除自相交, 进一步缩短子路径的长度达到局部寻优的效果。具体设计的算法如下:

算法1: OX- k -splitour算法

- 1) Find a 1-TSP tour $T_{TSP}=\langle v_0, v_1, \dots, v_n, v_0 \rangle$ with $L=|T_{TSP}|$;
- 2) for all $i(i=1,2,\dots,k-1)$ do
- 3) find the last node $v_{p(i)}$ such that the length of the path from v_0 to $v_{p(i)}$ along T_{TSP} is not greater than $(i/k)(L-2c_{\max})+c_{\max}$, where $c_{\max}=\max_j d(v_0, v_j)$;
- 4) end for
- 5) form k subtours as $ST_1=\langle v_0, v_1, \dots, v_{p(1)}, v_0 \rangle$, $ST_2=\langle v_0, v_{p(1)+1}, \dots, v_{p(2)}, v_0 \rangle, \dots, ST_k=\langle v_0, v_{p(k-1)+1}, \dots, v_n, v_0 \rangle$;
- 6) for all $i(i=1,2,\dots,k)$ do
- 7) $E_{\text{end}}=e(v_{p(i)}, v_0)$;
- 8) $E_{\text{start}}=e(v_0, v_1)$;
- 9) for all nodes j in ST_i
- 10) if $e(v_j, v_{j+1})$ intersected with E_{end}
- 11) $v_{j+1} \leftrightarrow v_{p(i)}$;
- 12) reverse visit sequence of $\langle v_{j+1}, v_{j+2}, \dots, v_{p(i)} \rangle$;
- 13) end if
- 14) end for
- 15) for all nodes j in ST_i
- 16) if $e(v_j, v_{j+1})$ intersected with E_{start}
- 17) $v_{j+1} \leftrightarrow v_1$;
- 18) reverse visit sequence of $\langle v_1, v_2, \dots, v_{j+1} \rangle$;
- 19) end if
- 20) end for
- 21) end for
- 22) return $ST=\{ST_1, ST_2, \dots, ST_k\}$

算法1首先使用现有TSP问题求解算法或工具(如concorde^[16])找到一条TSP回路, 第2)~5)行采用 k -splitour算法得到 k 条TSP子路径, 第9)~14)行以及第15)~20)行分别用于检测及消除各个子路径中起始边及结束边所产生的自相交, 时间复杂度均为 $O(n)$, 因此算法1的时间复杂度为 $C_{TSP}+O(n)$, C_{TSP} 是所用TSP近似算法的时间复杂度。由于消除了 k -splitour中的自相交边, 所以算法的近似比 $APP \leq$

$\alpha+1-1/k$ 。

2.2 访问点位置的优化

由算法1可以得到各条可行子路径解, 接下来需要确定子路径上在圆周边界上的访问点的位置。由于式(1)中的访问点要求在传感器通信圆盘上^[4], 传感器节点 v_{j_i} 的访问点 $p_{\pi_{j_i}}^i$ 可以表示为 $\langle x_{j_i}+r_{j_i}\cos\theta_{j_i}, y_{j_i}+r_{j_i}\sin\theta_{j_i} \rangle$, 其中 $\langle x_{j_i}, y_{j_i} \rangle, \theta_{j_i} \in [0, 2\pi]$ 。此时子路径长度是 $\theta=\langle \theta_{v_{j_0}}, \theta_{v_{j_1}}, \dots, \theta_{v_{j_n}} \rangle$ 的函数, 记作 $f(\theta)$, 由式(1)可知 $f(\theta)$ 是一个非凸非线性复杂的连续函数。

在优化路径上, 每一个节点的访问点对应的 θ 的取值范围, 可以由 $[0, 2\pi]$ 缩小到由该节点通信圆盘到下一个节点通信圆盘的两条外切线所包围的区间^[4]。文献[4]利用这一特性压缩了访问点取值的搜索范围, 从而提高了求解效率。由于没有考虑前后节点的位置对当前访问点的影响, 搜索范围仍然比较大, 为此本文进行如下设计进一步压缩搜索范围, 提高求解效率。如图1所示考虑两个静态节点的情形, 移动sink从基站 S 出发, 依次访问节点 A 、 B 后回到基站。节点 A 上访问点的可行取值区域位于弧 $\widehat{a_1 a_2}$ 与弧 $\widehat{a_3 a_4}$ 的交集即弧 $\widehat{a_3 a_2}$ 之间, 同理, B 上的可行区域在弧 $\widehat{b_3 b_1}$ 之间。弧段 $\widehat{b_2 b_1}$ 的取值范围为 $[\pi/2+\alpha-\beta, 3\pi/2-\alpha-\beta]$, 其中, $\alpha=\arcsin\{(r_B-r_A)/D_{AB}\}$, $\beta=\arcsin\{(y_A-y_B)/D_{AB}\}$, D_{AB} 为节点 A 、 B 间的距离。对于多个静态节点的情形, 每个节点对应弧段的起始范围可用类似的方法计算得到。

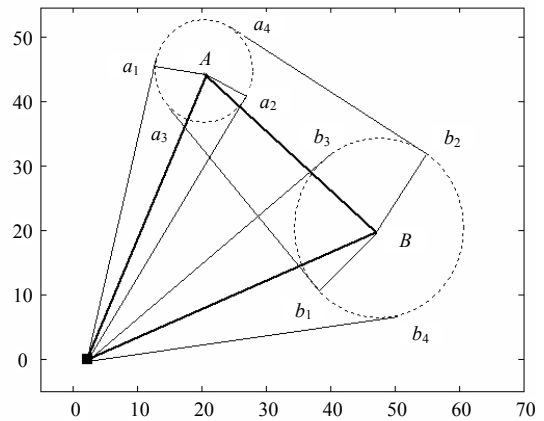


图1 移动sink由 S 出发依次访问节点 A 、 B 的通信区域

对 θ 的取值范围作了裁剪限定之后, 除了减少了搜索空间的范围, 解空间中的极值点也减少了。图2给出了示例图的解空间分布的情况。如图2a所示, 在 $[0, 2\pi]$ 的整个定义域上, 路径长度存在多个极值点, 而在压缩了 θ 的取值范围之后, 图2b所示的解空间呈现出单一最值的波谷, 为优化解的搜索提供了极

大的便利。

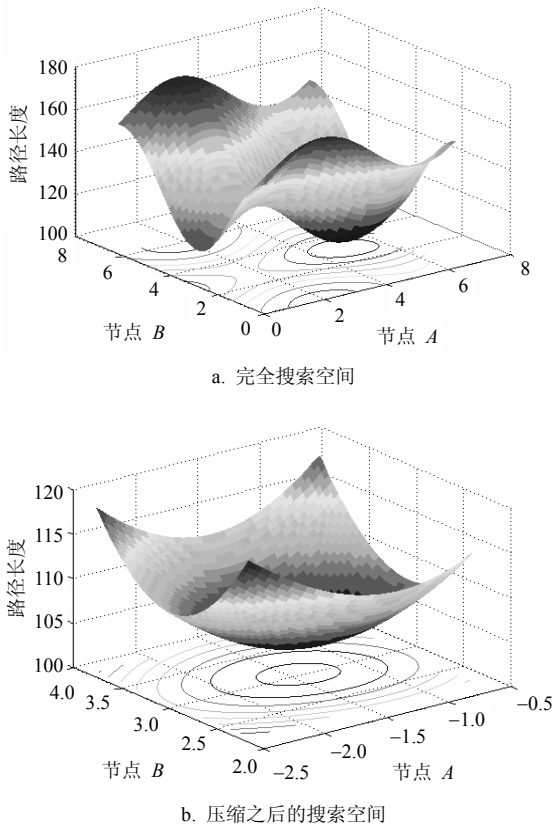


图2 示例图1中的完全搜索空间和压缩之后的搜索空间比较

受裁剪搜索空间之后解结构空间变化的启发,本文提出了一种基于贪婪路径覆盖和近似梯度下降的搜索策略,用于获取路径规划问题的近似最优解。算法的主要思想是从当前节点 v_i 的访问点位置 p_i 开始,以路径覆盖的方式,前向查找到第一个未能构成路径覆盖的节点 v_j ,随后在 v_i 和 v_j 构成的节点集合之上,使用梯度下降算法搜索每个节点之上的访问点位置对应的 θ 。对参数 θ_{v_i} 的偏导数可由下式计算得到:

$$\frac{\partial f}{\partial \theta_{v_i}} \approx \frac{f(\theta') - f(\theta)}{\Delta \theta}$$

$$f(\theta') = f(\theta_{v_1}, \theta_{v_2}, \dots, \theta_{v_i} + \Delta \theta, \dots, \theta_{v_n}) \quad (2)$$

目标函数的梯度可表示为:

$$\nabla f(\theta) = \left(\frac{\partial f}{\partial \theta_{v_1}}, \frac{\partial f}{\partial \theta_{v_2}}, \dots, \frac{\partial f}{\partial \theta_{v_n}} \right)^T \quad (3)$$

所用迭代公式为:

$$\theta_{v_i}(k+1) = \theta_{v_i}(k) - \rho \frac{\nabla f(\theta_{v_i})}{\|\nabla f(\theta_{v_i})\|} \quad (4)$$

由以上分析可得设计的算法如下:

算法2: 贪婪近似梯度下降搜索算法(Greedy approximate gradient based search algorithm, GAGBS)

Input: TSP tour for $V(v_0)$ presents base station which location is p_0 : i.e. $T_{TSP} = \langle v_0, v_1, v_2, \dots, v_n, v_0 \rangle$

- 1) $T_{GAGBS} \leftarrow p_0$;
- 2) calculate θ_i 's upper and lower bound for all nodes in T_{TSP}
- 3) for all v_i in T_{TSP}
- 4) find the largest v_j that the distance of all the sensor nodes(v_i, v_{i+1}, \dots, v_j) to the tour less than r_i
- 5) form the subtour $ST = \langle v_i, v_{i+1}, \dots, v_j, v_{j+1} \rangle$
- 6) apply approximate gradient based local search on ST, obtain access points $\langle p_{i+1}, p_{i+2}, \dots, p_{j+1} \rangle$
- 7) append $\langle p_{i+1}, p_{i+2}, \dots, p_j \rangle$ to T_{GAGBS}
- 8) replace $\langle v_i, v_{i+1}, \dots, v_j \rangle$ with v_j' (location is p_j) in T_{TSP}
- 9) end for
- 10) combine turn point for T_{GAGBS}
- 11) return T_{GAGBS}

在迭代中 θ 的初值设为TSP路径与各个节点通信圆盘的第一个交点的弧度,若 θ 超出压缩后解空间的范围,取对应的上界或下界的值,当两次迭代得到的路径长度小于阈值时迭代终止。

2.3 算法时间复杂度及近似比分析

该算法采用分段路径上的梯度信息代替全局梯度信息,可避免梯度下降迭代过程中由于节点数目过大导致的每次计算目标函数时的用时过高,且容易过早陷入全局极值的缺陷。贪婪梯度搜索过程中,每个节点迭代次数的上界是 $\lfloor 2\pi/\rho \rfloor$,算法的时间复杂度为 $O(\lfloor 2\pi/\rho \rfloor mn)$,其中 ρ 为梯度下降的步长, m 是每次贪婪路径覆盖的节点数目的平均期望值, $m \leq n$ 。由于路径上过多的转折点的数目会对移动sink的移动性产生影响,所以在算法的第10)行,对已获得的访问点再使用路径覆盖的思想进行一次时间复杂度为 $O(n)$ 的转折点合并操作,达到减少转折点数目并进一步缩短路径长度的目的。综合以上分析,算法的时间复杂度为 $C_{TSP} + O\left(\left\lfloor \frac{2\pi}{\rho} \right\rfloor mn\right)$ 。

定理 1 算法所得路径长度的上界为 $|T_{app}| \leq (\alpha+1-1/k) \left(|T_{opt}| + 2 \sum_{i=1}^m r_i \right)$,其中 $|T_{app}|$ 是本文提出的算法所得路径长度, T_{opt} 是最长的子路径上的最优路径, m 为此路径上的传感器节点数目。

证明: 整个算法由算法1和算法2两阶段组成。

设此子路径所对应的最优TSP路径为 T_{TSP} ,由2.1节对算法1的分析可知, $|T_{ox}| / |T_{TSP}| \leq (\alpha+1-1/k)$.在算法2每次迭代中, 都有 $|T_{k+1}| \leq |T_k|$, 因此算法产生的是一个非增序列。由于子路径初始长度为 $|T_{ox}|$, 所以 $|T_{GAGBS}| \leq |T_{ox}|$. 因为 T_{opt} 到路径上第 i 个节点的距离一定小于 r_i , 所以移动sink沿着 T_{opt} 到每个节点的通信区域时, 访问该节点一次之后再返回到 T_{opt} 继续前进所形成的路径也是TSP问题的近似解, 所以有: $|T_{TSP}| \leq (|T_{opt}| + 2 \sum_{i=1}^m r_i)$. 综合以上分析, 定理得证。

3 算法仿真及比较

为了验证本文所提出的路径规划模型及其求解算法的有效性, 进行了如下仿真实验。所用硬件平台为 Intel(R) Core(TM) i7-3632QM 2.2GHz, 4 GB 内存, 软件平台为 Matlab2012b。文献[11,13]指出延时性能与路径长度呈正相关关系, 进而以路径长度衡量延时性能, 本文采取同样的方法与其他算法进行比较。仿真实验在500 m×500 m的监控区域内随机部署传感器节点。为观察算法所用参数对算法性能的影响, 在随机分布有100个通信范围为20 m的节点的场景下进行了实验。实验随机生成了1 000个样本, 统计结果见表1, 其中, 计算时间为各次计算的均值。可以看出, 当 $\rho=\pi/20$ 时可以有较低的计算时间取得较好的计算结果。

表1 100个节点时不同的 ρ 对算法的影响

ρ	平均长度/m	计算时间/s
$\pi/8$	1 839.4	3.211 7
$\pi/10$	1 771.6	4.197 2
$\pi/16$	1 702.7	6.172 8
$\pi/20$	1 689.4	7.390 3
$\pi/30$	1 687.5	10.722 6
$\pi/40$	1 684.3	13.569 4

表2 改进后的 k -splitour与 k -splitour的性能对比

k	改进前/m	改进后/m	性能提升/%
2	2 378.7	2 304.7	3.110 9
3	1 865.6	1 758.4	5.746 1
4	1 510.9	1 378.5	8.762 0
5	1 317.9	1 168.4	11.343 8

为了验证改进后的 k -splitour算法的性能, 本文在与以上实验相同的场景下, 设置不同的移动节点数目进行了测试, 考察1 000次随机实验的均值, 得到的结果如表2所示。从表2中可以发现, k 取不同值时, 改进后的 k -splitour算法的均较原有算法的性能有所提升。同时从表2中也可以看出, 随着移动sink

数目 k 的增加, 改进算法所获得的性能提升越明显, 这是因为子路径数目的增加会导致更多的路径自相交。

为了测试算法的在不同节点规模问题下的性能, 设计了如下的场景进行测试: 节点数目分别从初始值50开始, 以10为增量逐步增加到100个, 对于每种节点数目的场景, 分别生成100个测试样本。统计结果是针对该场景下所有测试样本的均值。近似梯度下降所用参数 $\rho=\pi/20$, 迭代终止阈值为0.5。为验证GAGBS算法的有效性, 首先比较了使用单移动sink时不同算法的表现。本文采用与文献[14]一致的参数, 即对于所有的节点, 通信半径均取值为20 m。与LC算法^[12]和CSS算法^[14]对比的结果如图3所示。可以看出, GAGBS算法的性能较LC算法有较大的提升, 比CSS算法亦有5%左右的改进。

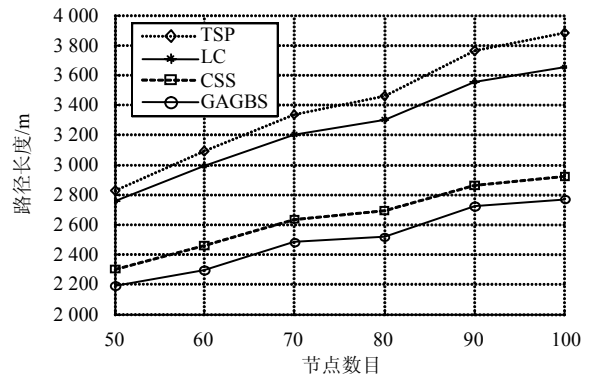


图3 不同节点数目下路径长度的比较图

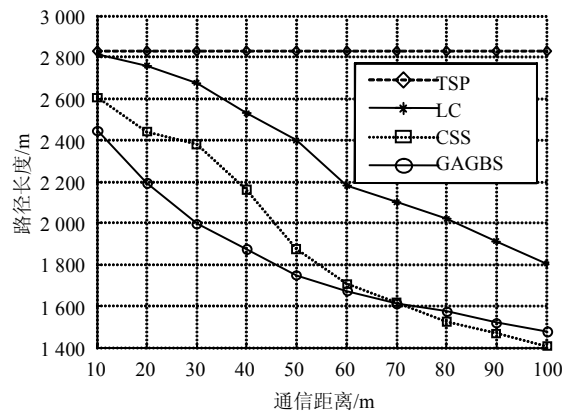


图4 不同通信范围时路径长度的比较图

图4显示了50个节点的情况下, 路径长度随着通信半径增长的变化情况, 可以看出当节点通信范围超过70 m之后, GAGBS算法的性能开始下降, 这是由于梯度下降所用步长设置过大导致的, 该参数可以根据实际应用进一步优化。

图5显示了一个使用4个移动sink, 节点通信范围

服从 $20\sim 50\text{ m}$ 之间的均匀分布(表示为 $U(20,50)$)场景下的路径实例。图6的实验显示了使用100个传感器节点,通信范围服从 $U(20,50)$ 分布时,不同数目的移动sink对 k -splitour、 k -LC和 k -GAGBS算法产生的路径影响,可以看出本文所提出的算法大大缩短了子路径的长度,取得了较好的效果。

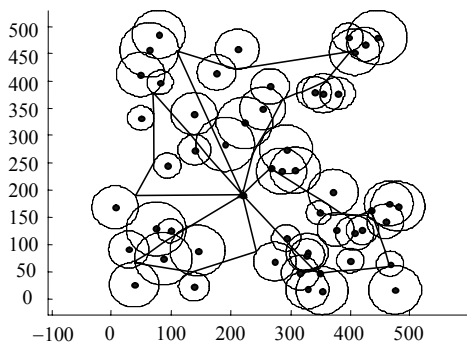


图5 使用4个移动sink时的路径实例

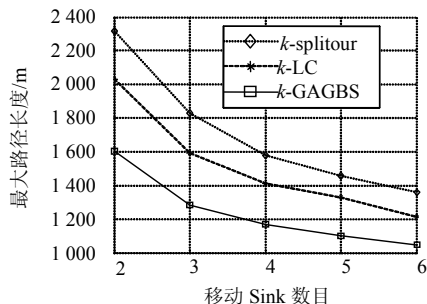


图6 不同移动sink数目最大路径长度的比较图

4 结论

本文将无线传感器网络中多移动sink数据收集中的路径规划问题建模为路径端点在通信圆盘之上的混合优化问题,并设计了求解的近似算法。给出并在理论上证明了算法的上界,最后通过仿真实验及与经典的CSS、 k -LC算法的对比,验证了该模型及其求解算法的有效性。

本文得到广西自动检测技术与仪器重点实验室基金(YQ16205)、广西高校云计算与复杂系统重点实验室资助项目(2015209)资助,在此表示感谢。

参考文献

[1] XU X, LUO J, ZHANG Q. Delay tolerant event collection in sensor networks with mobile sink[C]//Proceedings of the IEEE INFOCOM. San Diego, USA: IEEE, 2010: 1-9.
 [2] GUPTA P, KUMAR P. Critical power for asymptotic connectivity in wireless networks[M]//WILLIAM M M, GEORGE Y, ZHANG Qing. Stochastic Analysis, Control,

Optimization and Applications. New York: Springer, 1998: 547-566.

- [3] WU Fang-jing, TSENG Yu chee. Energy-Conserving data gathering by mobile mules in a spatially separated wireless sensor network[J]. Wireless Communications and Mobile Computing, 2013, 13(15): 1369-1385.
 [4] YUAN Bo, ORLOWSKA M, SADIQ S. On the optimal robot routing problem in wireless sensor networks[J]. IEEE Trans on Knowledge and Data Engineering, 2007, 19(9): 1252-1261.
 [5] MARIO D F, SAJAL K D, GIUSEPPE A. Data collection in wireless sensor networks with mobile elements: a survey[J]. ACM Trans on Sensor Networks, 2011, 8(1): 108-142.
 [6] MA Ming, YANG Yuan-yuan, ZHAO Miao. Tour planning for mobile data-gathering mechanisms in wireless sensor networks[J]. IEEE Trans on Vehicular Technology, 2013, 62(4): 1472-1483.
 [7] DE BERG M, GUDMUNDSSON J, KATZ M J, et al. TSP with neighborhoods of varying size[J]. Journal of Algorithms, 2005, 57: 22-36.
 [8] DUMITRESCU A, MITCHELL J S B. Approximation algorithms for TSP with neighborhoods in the plane[J]. Journal of Algorithms, 2003, 48(1): 135-159.
 [9] ELBASSIONI K, FISHKIN A, MUSTAFA N, et al. Approximation algorithms for euclidean group TSP[C]//32nd International Colloquium on Automata, Languages and Programming. Lisbon: Springer Verlag, 2005: 1115-1126.
 [10] MITCHELL J. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane[C]//Proceedings of SoCG. New York, NY, USA: ACM, 2010.
 [11] SUGIHARA R, GUPTA R. Improving the data delivery latency in sensor networks with controlled mobility[C]//Proceedings of DCOSS. Santorini Island, Greece: [s.n.], 2008: 386-399.
 [12] SUGIHARA R, RAJESH K G. Path planning of data mules in sensor networks[J]. ACM Transactions on Sensor Networks, 2011, 8(1): 1-27.
 [13] 袁远, 彭宇行, 李姗姗, 等. 高效的移动Sink路由问题的启发式算法[J]. 通信学报, 2011, 32(10): 107-117.
 YUAN Yuan, PENG Yu-xing, LI Shan-shan, et al. Efficient heuristic algorithm for the mobile sink routing problem[J]. Journal on Communications, 2011, 32(10): 107-117.
 [14] HE Liang, PAN Jian-ping, XU Jing-dong. A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements[J]. IEEE Trans on Mobile Computing, 2013, 12(7): 1308-1320.
 [15] FREDERICKSON G N, HECHT M S, KIM C E. Approximation algorithms for some routing problems[C]//Symposium of foundations of Computer Science. Houston, TX, USA: IEEE, 1976: 216-227.
 [16] COOK W. Concorde TSP solver[EB/OL]. [2014-06-02]. <http://www.math.uwaterloo.ca/tsp/concorde.html>.

编辑 蒋晓