

# 基于网络覆盖和多目标离散群集蜘蛛算法的多移动 agent 规划

刘洲洲<sup>1,2</sup>, 李士宁<sup>2</sup>

(1. 西安航空学院电子工程学院, 陕西 西安 710077; 2. 西北工业大学计算机学院, 陕西 西安 710072)

**摘要:** 以 agent 负载能耗均衡度和网络总能耗为指标构建多移动 agent 协作规划模型, 为了尽可能延长网络生存周期, 给出基于网络覆盖率的节点休眠机制, 在满足 WSN 网络覆盖率要求的同时, 采用较少节点处于工作状态。根据多移动 agent 协作规划技术特点, 设计融合 Pareto 最优解多目标离散群集蜘蛛算法 (MDSSO), 重新定义插值学习和变异交换粒子更新策略, 并动态调整最优解集规模, 以提高 MDSSO 算法多目标求解精度。实验仿真结果表明, 该方法能够快速合理给出 WSN 多移动 agent 规划路径, 而且与其他传统算法相比, 网络总能耗降低了约 15%, 生存期提高了约 23%。

**关键词:** 无线传感器网络; 移动代理; 网络覆盖; 群集蜘蛛优化算法; 协作规划

中图分类号: TP393

文献标识码: A

## Multi mobile agent itinerary planning based on network coverage and multi-objective discrete social spider optimization algorithm

LIU Zhou-zhou<sup>1,2</sup>, LI Shi-ning<sup>2</sup>

(1. School of Electronic Engineering, Xi'an Aeronautical University, Xi'an 710077, China;

2. School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** The multi mobile agent collaboration planning model was constructed based on the mobile agent load balancing and total network energy consumption index. In order to prolong the network lifetime, the network node dormancy mechanism based on WSN network coverage was put forward, using fewer worked nodes to meet the requirements of network coverage. According to the multi mobile agent collaborative planning technical features, the multi-objective discrete social spider optimization algorithm (MDSSO) with Pareto optimal solutions was designed. The interpolation learning and exchange variations particle updating strategy was redefined, and the optimal set size was adjusted dynamically, which helps to improve the accuracy of MDSSO. Simulation results show that the proposed algorithm can quickly give the WSN multi mobile agent path planning scheme, and compared with other schemes, the network total energy consumption has reduced by 15%, and the network lifetime has increased by 23%.

**Key words:** wireless sensor network, mobile agent, network coverage, social spider optimization algorithm, itinerary planning

### 1 引言

无线传感器网络 (WSN, wireless sensor network) 作为 21 世纪最主要应用技术之一<sup>[1]</sup>, 已被成功地应用于环境监测、物流配送、追踪定位、战场

情报收集等领域<sup>[2]</sup>。由于受传感器节点硬件设备、计算能力和功耗限制, 如何在满足业务需求前提下, 尽可能降低网络能耗和延长生存周期<sup>[3]</sup>, 已成为当前 WSN 研究热点之一。

随着 WSN 应用领域的不断扩展, 这就要求网

收稿日期: 2016-09-09; 修回日期: 2017-04-18

基金项目: 国家自然科学基金资助项目 (No.61601365); 陕西省教育厅科研计划基金资助项目 (No.16JK1395)

**Foundation Items:** The National Natural Science Foundation of China (No. 61601365), The Scientific Research Program Funded by Shaanxi Provincial Education Department (No. 16JK1395)

络具备远程监控、自动配置以及自我调节等功能<sup>[4]</sup>, 移动 agent (MA, mobile agent) 技术的使用, 改变了传统簇式通信结构<sup>[5]</sup>, 融合 MA 技术的无线传感器网络 (MAWSN, mobile agent wireless sensor network) 采用 MA 技术收集传感器节点数据, 有效地提高了网络数据通信效率<sup>[6]</sup>。对于大规模传感器网络, 合理规划移动 agent 路径对提高网络整体性能具有重要意义。常见的多代理路径协作规划 (MIP, multi agent itinerary planning) <sup>[7]</sup> 算法有 GA-MIP<sup>[8]</sup>、TBID<sup>[9]</sup> 和 BST/MST<sup>[10]</sup> 等, 这些算法在一定程度上缓解了网络能量消耗过快和数据延迟问题, 但是对于传感器节点部署不规则网络, 会导致部分节点能耗过大, 少数移动 agent 负载过高, 从而影响了网络生存时间。为此, 学者们开展了系列研究, 史霄波等<sup>[6]</sup> 针对移动 agent 负载不均衡问题, 提出了一种离散多目标优化粒子群 (MIP) 算法, 有效解决了单个节点能耗过大问题, 但是没有考虑后期节点“死亡”情况。文献[11]提出一种基于 ACA-SAA 的移动 agent 路径规划方案, 仿真实验验证了该方案的有效性, 但是并不适用于拓扑结构发生变化的网络。由于 WSN 常常部署在恶劣环境中, 负载均衡、网络能耗、网络延迟以及动态 MA 协作规划等仍是 MIP 研究亟需解决的难题。

针对上述问题, 本文提出一种基于网络覆盖和多目标离散群集蜘蛛算法的多移动 agent 规划方法 (MDSSO-MIP)。通过设计基于网络覆盖率的节点休眠机制, 以实现动态多移动 agent 协作规划。提出符合 MIP 技术要求的多目标离散群集蜘蛛算法 (MDSSO), MDSSO 以 agent 负载能耗均衡度和网络总能耗为多目标优化函数, 经过循环迭代进化, 从而实现多移动 agent 规划问题求解。最后, 通过实验仿真验证该方法的有效性。

## 2 问题描述

构建如图 1 所示的 MAWSN 模型, 在监控区域内部署一个汇聚节点 (sink),  $W$  个传感器节点  $SN_i \in \tilde{V}$  ( $i=1,2,\dots,W$ ,  $\tilde{V}$  为传感器节点集合)。某时刻, sink 根据任务需求指派多个 MA 执行数据采集任务, MA 采用无线多跳的方式逐个访问预定路径中的监测传感器节点, 并将数据带回汇聚节点。sink 节点对汇总数据进行运算处理后, 通过无线网络、无人机等方式完成与监控中心的交互通信。

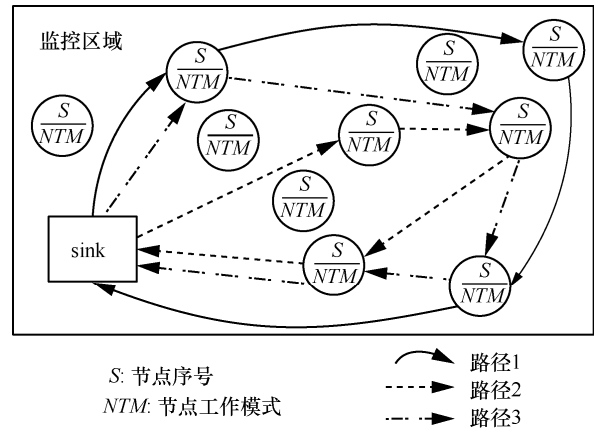


图 1 融合 MA 技术的无线传感器网络模型

**假设 1** 传感器节点与汇聚节点数量以及位置信息已知, sink 能量、处理能力不受限制,  $SN_i$  位置固定且各节点指标规格相同。

**假设 2** 设定传感器节点工作模式 NTM (node task mode) 为工作状态、休眠状态和死亡状态, 并且只有处于工作状态的节点才能执行监控任务。

**假设 3** 传感器节点部署相对密集, 且只有部分节点处于监测工作状态; MA 能够自主唤醒处于休眠状态的节点, 并引导其完成数据中继等任务。

**定义 1**  $SN_i$  编号  $S_i$  唯一, 且  $S_i = i, i=1,2,\dots,W$ 。

**定义 2** MAWSN 抽象为无向图  $G(\tilde{V}, E)$ , 将传感器节点集合  $\tilde{V}$  划分为  $M$  个子集  $V_k$  ( $k=1,2,\dots,M$ ), 其中,  $M$  表示 sink 派出的 MA 数量。

**定义 3** MA 完成对第  $i$  个监测节点数据采集与压缩融合处理后, 其数据  $D_{MA}^i$  大小推导过程为

$$\begin{aligned}
 D_{MA}^1 &= D_{MA}^0 + D'_{SN} \\
 D_{MA}^2 &= D_{MA}^1 + (1-\rho)D'_{SN} = D_{MA}^0 + (1-\rho)D'_{SN} + D'_{SN} \\
 D_{MA}^3 &= D_{MA}^2 + (1-\rho)D'_{SN} = D_{MA}^1 + 2(1-\rho)D'_{SN} \\
 &= D_{MA}^0 + 2(1-\rho)D'_{SN} + D'_{SN} \\
 &\vdots \\
 D_{MA}^i &= D_{MA}^{i-1} + (1-\rho)D'_{SN} = D_{MA}^1 + (i-1)(1-\rho)D'_{SN} \\
 &= D_{MA}^0 + [1+(i-1)(1-\rho)]D'_{SN} \\
 D_{MA}^i D'_{SN} &= (1-r)D_{SN} D_{MA}^0 + \\
 &\quad [1+(i-1)(1-\rho)](1-r)D_{SN} \quad (1)
 \end{aligned}$$

其中,  $D_{MA}^0$ 、 $D_{SN}$  分别为 MA 自身数据和监测节点感知数据大小;  $D'_{SN}$  为监测节点数据压缩后的大小;  $\rho$ 、 $r$  分别为数据融合率和数据压缩率。

**定义 4** MA 完成对路径中所有监测节点数据收集处理任务后, 其能耗为

$$E_{MA} = H(SkN_j \odot SN_{MA}^1) e(D_{MA}^0, D_{MA}^0) + \sum_{l=2}^{q_j} E_{l-1}^l (D_{MA}^{l-1}) + m_p D_{SN} + e(0, D_{MA}^{q_j}) + H(SN_{MA}^{q_j} \odot SkN_j) e(D_{MA}^{q_j}, D_{MA}^{q_j}) \quad (2)$$

$$H(SN_{MA}^{i-1} \odot SN_{MA}^i) = \frac{\|SN_{MA}^{i-1}, SN_{MA}^i\|}{R} \quad (3)$$

$$e(D_{RX}, D_{TX}) = m_{RX} D_{RX} + m_{TX} D_{TX} + e_{cut} \quad (4)$$

$$E_{l-1}^l (D_{MA}^{l-1}) = m_p D_{SN} + e(0, D_{MA}^{l-1}) + H(SN_{MA}^{l-1} \odot SN_{MA}^l) e(D_{MA}^{l-1}, D_{MA}^{l-1}) + e(D_{MA}^{l-1}, 0) \quad (5)$$

其中,  $SkN_j$  表示  $k$  子集内的节点编号,  $SN_{MA}^l$  ( $l=1, 2, \dots, q_j$ ) 表示 MA 路径中监测节点;  $R$  为节点最大传输距离;  $m_{TX}$ 、 $m_{RX}$  分别表示发送和接收单位数据能耗;  $e(D_{RX}, D_{TX})$  表示融合接收和发送节点数据能耗;  $e(D_{MA}^0, D_{MA}^0)$  表示融合节点自身数据和感知数据能耗;  $e(0, D_{MA}^{q_j})$  表示 MA 路径中的监测节点能耗;  $m_p$  为节点感知、压缩单位数据能耗;  $e_{cut}$  为通信能耗控制阈值;  $H(SN_{MA}^{l-1} \odot SN_{MA}^l)$  表示相邻监测节点间多级跳数估计数,  $\odot$  表示计数, 设定 MA 在相邻监测节点多跳过程中数据大小不发生变化。从式 (5) 可以看出, MA 能耗主要包括数据收发、感知、压缩以及融合等。

**定义 5** 网络 MA 负载能耗均衡度为

$$B_{load} = \frac{\sum_{i=1}^M \left| \max\{E_i^1, E_i^2, \dots, E_i^{q_i}\} - \sum_{k=1}^{q_i} E_i^k \right|}{\sum_{i=1}^M \sum_{k=1}^{q_i} E_i^k} \quad (6)$$

其中,  $E_i^k$  为第  $i$  个 MA 访问路径中第  $k$  个监测节点能耗。从式 (6) 可以看出,  $B_{load}$  取值越小, 表明监测节点能耗更加均衡。

**定义 6** MAWSN 移动 agent 协作规划评价函数为

$$f(G) \begin{cases} f_1 = \min \left( \sum_{i=1}^M E_{MA}^i \right) \\ f_2 = \min(B_{load}) \end{cases} \quad (7)$$

从式(7)可以看出, 评价函数以负载能耗均衡度

和网络总能耗为优化目标, 在尽可能降低网络能耗的同时, 确保各个节点能耗更加均衡。

### 3 基于网络覆盖率的节点休眠机制

对于节点部署相对密集的网络, 合理切换节点工作模式 (NTM) 可以在保障满足监控任务要求的前提下, 使用较少节点处于监测工作模式, 有效延长了网络生存周期<sup>[12]</sup>。为此, 提出基于网络覆盖率的节点休眠机制。

对于节点  $SN_i$  ( $i=1, 2, \dots, W$ ), 其感知域节点集合为  $Per_i = \{SN_k \mid \|SN_k - SN_i\| \leq R_k + R_i, k \in \tilde{V} \wedge k \neq i\}$ , 其中,  $R_k$  为节点感知半径。若节点  $SN_i$  感知域可以被  $Per_i$  内其他节点感知域覆盖, 则称  $SN_i$  为可替代节点。例如, 可以使节点  $SN_a$  的感知域  $O_{R_a}$  完全被节点  $SN_b$ 、 $SN_c$  及  $SN_d$  覆盖, 此时关闭  $SN_a$ , 使其处于休眠状态。

**定义 7** 对于  $SN_i$ , 其感知域与  $Per_i$  内节点感知域重叠面积和自身感知域的比值定义为节点重叠率为

$$REP_i = \frac{\sum_{SN_k \in Per_i} A(SN_k \cap SN_i)}{A(SN_i)} \quad (8)$$

其中,  $A(SN_i)$  为  $SN_i$  自身感知域面积,  $A(SN_k \cap SN_i)$  为  $SN_i$  与  $Per_i$  内节点感知域重叠面积。

**定理 1** 对于  $SN_i$ , 其节点重叠率期望值为

$$E(REP_i) = 1 - \left( \frac{3}{4} \right)^{|Per_i|} \quad (9)$$

其中,  $|Per_i|$  为  $SN_i$  感知域节点集合内节点数量。

**证明** 对于节点  $SN_a$ , 取节点  $SN_b \in Per_a$ , 即  $0 < \|SN_a - SN_b\| \leq R_a + R_b$ 。在  $SN_a$  感知域  $O_{R_a}$  内任意选取点  $c$ , 若  $c$  能够被  $SN_b$  感知到, 则点  $c$  与  $SN_b$  满足

$$\|c - SN_b\| \leq R_b \quad (10)$$

由于节点随机部署在监控区域内, 因此, 节点处于监控区域的概率为  $P(SN) = \frac{1}{A(T)}$ , 显然,  $c$  被  $SN_b$  监控到的概率为

$$P(c \in O_{R_b}) = \frac{A(O_{R_b})P(SN)}{A(O_{R_a+R_b})P(SN)} = \frac{R_b^2}{(R_a + R_b)^2} \quad (11)$$

其中,  $A(T)$  为整个网络监控区域面积;  $A(O_{R_a+R_b})$  为半径为  $R_a + R_b$  的感知域面积。因此,  $SN_a$  感知域内任意点被  $Per_a$  内节点感知到的概率为

$$P_a = 1 - \frac{\left(1 - P(c \in O_{R_b})\right) \left(1 - P(c \in O_{R_b})\right) \cdots \left(1 - P(c \in O_{R_b})\right)}{|Per_i|}$$

$$= 1 - \left(1 - P(c \in O_{R_b})\right)^{|Per_i|} = \left(1 - \frac{R_b^2}{(R_a + R_b)^2}\right)^{|Per_i|} \quad (12)$$

当  $R_a = R_b$  时, 节点重叠率期望值为

$$E(REP_i) = \frac{P_a A(SN_a)}{A(SN_a)} = 1 - \left(\frac{3}{4}\right)^{|Per_i|} \quad (13)$$

证毕。

节点休眠机制。给定网络覆盖率最低值  $P_{\min}$ , 若  $E(REP_i) \geq P_{\min}$ , 表明节点  $SN_i$  可以进入休眠状态。为进一步平衡网络节点能耗, 规定当节点能量

$$e(SN_a) \text{ 满足 } e(SN_a) \leq \eta \left[ \frac{\sum_{k=1}^W e(SN_k)}{W} \right]$$

且关闭该节点不影响网络覆盖要求时, 则强制该节点进入休眠状态。图 2 给出了节点休眠机制工作流程。

## 4 MDSSO-MIP 算法

### 4.1 MDSSO 改进

2013 年, Cuevas 等<sup>[13]</sup>提出了群集蜘蛛优化 (SSO, social spider optimization) 算法, 并通过经典函数测试验证了算法收敛性能。SSO 正逐渐成为智能启发计算技术研究新的热点, 并在连续优化问题领域取得了一定研究成果<sup>[14,15]</sup>, 然而关于离散及多目标 SSO 算法的研究则很少。本文结合 MIP 技术特点, 创新地给出多目标离散群集蜘蛛算法 (MDSSO) 的具体实现流程 (以雌性蜘蛛个体为例, SSO 基本原理参考有关文献, 不再赘述)。

**定义 8** MDSSO 算法中, 每个粒子代表 MIP 的一个解, 而 MIP 的目的就是找到多个 MA 最优移动路径。因此, 对于具有  $W$  个传感器节点、 $M$  个移动 agent 的 MAWSN, 定义蜘蛛个体  $F_i$  编码方式为

$$F_i \leftarrow \begin{cases} Z(F_i) = W \\ F_{ij} = F_{ij}(S_k, MA_g), g \in [1, 2, \dots, M] \end{cases}$$

s.t.  $\forall a \in [1, 2, \dots, W-1], b \in [2, \dots, W] \cap a < b$

$$\begin{cases} F_{ia}[1] \neq F_{ib}[1] \\ F_{ia}[2] \leq F_{ib}[2], F_{iw}[2] = M \\ \{F_{i1}[2], F_{i2}[2], \dots, F_{iw}[2]\} \cap \{1, 2, \dots, M\} = \{1, 2, \dots, M\} \end{cases} \quad (14)$$

其中,  $Z(F_i)$  为粒子维度,  $F_{ij}$  为第  $j$  维编码位;  $S_k$ 、 $MA_g$  分别为传感器节点编号和 MA 编号。采用式(15)编码方式, 准确表达出了所有 MA 路径节点信息, 例如, 有个体  $F((3,1), (5,1), (1,1), (2,2), (4,2), (7,3), (6,3), (8,3))$ , 则  $W=8$ 、 $M=3$ , 且  $MA_1$  路径为  $\{3, 5, 1\}$ ,  $MA_2$  路径为  $\{2, 4\}$ ,  $MA_3$  路径为  $\{7, 6, 8\}$ 。

**定义 9** 对于个体  $F_i$  和  $F_j$ , 如果满足

$$f_1(F_i) \leq f_1(F_j) \wedge f_2(F_i) \leq f_2(F_j) \quad (15)$$

称  $F_i$  支配  $F_j$ , 记为  $F_i \succ F_j$ 。

**定义 10** 当个体  $F^*$  满足  $\neg \exists F_i: F_i \succ F^*$  时, 称  $F^*$  为 Pareto 最优解; Pareto 最优解集合则称为 Pareto 最优解集, 记为  $P^* \triangleq \{F^* | \neg \exists F_i: F_i \succ F^*\}$ 。

最优解集规模控制。MDSSO 每次迭代结束都会得到新的 Pareto 最优解  $F_{\text{new}}^*(t)$ , 当  $F_{\text{new}}^*(t)$  不受当前  $P^*$  中任意粒子支配或支配  $P^*$  中所有粒子时, 将  $F_{\text{new}}^*(t)$  加入  $P^*$  中。随着迭代次数增加,  $P^*$  规模不断扩大, 定义  $P^*$  规模控制阈值  $\partial_{\max}$ , 当  $P^*$  规模大于  $\partial_{\max}$  时, 逐个去除适应度较差个体, 直到满足规模控制阈值为止。

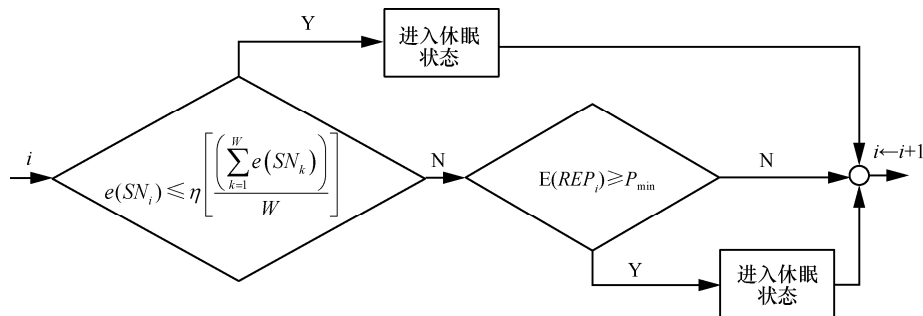


图 2 节点休眠机制工作流程

插值学习更新机制。当 MDSSO 采用定义 8 的编码方式时，如果仍按照传统 SSO 更新策略进行迭代进化，会产生大量不符合要求的解，因此，需要重新设计粒子更新机制。插值学习更新具体实现流程如下所示。

**Step1** 路径重组。设  $|MA_g|$  表示个体  $F_i$  第  $g$  个 MA 路径中节点个数，选取  $F_a$  且  $F_a$  适应度值  $f(F_a)$  优于  $f(F_i)$ ，令  $|MA_g|_{i,new} = |MA_g|_a$  ( $g=1,2,\dots,M$ )，此时  $F_i$  编码位  $F_{ij}[2]$  按照  $|MA_g|_{i,new}$  依次进行更新，并得到新的个体  $F_i^{new}$ 。

**Step2** 插值学习。按照 MA 编号大小，依次对  $F_{ij}^{new}[1]$  ( $j=1,2,\dots,W$ ) 进行更新：分别取  $F_i^{new}$  和  $F_a$  的  $MA_1$  路径节点集合  $MA_1^{i,new} \{F_{i_1}^{new}[1], F_{i_2}^{new}[1], \dots, F_{i_{|MA_1|}}^{new}[1]\}$  及  $MA_1^a \{F_{a_1}[1], F_{a_2}[1], \dots, F_{a_{|MA_1|}}[1]\}$ ，如果  $MA_1^{i,new} \cap MA_1^a \neq \emptyset$ ，令集合  $\mathfrak{R}_a = MA_1^a - MA_1^{i,new} \cap MA_1^a$ 、 $\mathfrak{R}_i^{new} = MA_1^{i,new} - MA_1^{i,new} \cap MA_1^a$ ，其中， $F_{ij}^{new}[1]$  表示更新后的  $j$  维编码位，new 表示对编码操作， $\mathfrak{R}_a$  表示编码位元素的集合， $\mathfrak{R}_i^{new}$  表示更新后的编码位元素集合，随机选取  $\mathfrak{R}_a$  内元素  $\mathfrak{R}_a[k]$  ( $k \in [1, 2, \dots, |\mathfrak{R}_a|]$ ) 替换  $\mathfrak{R}_i^{new}$  内元素  $\mathfrak{R}_i^{new}[l]$  ( $l \in [1, 2, \dots, |\mathfrak{R}_i^{new}|]$ ) (当  $\mathfrak{R}_a = \emptyset$  或  $\mathfrak{R}_i^{new} = \emptyset$  时，不执行替换操作)，此时，还需要将  $F_{ij}^{new}[1] = \mathfrak{R}_a[k]$  的编码位重新赋值为  $F_{ij}^{new}[1] = \mathfrak{R}_i^{new}[l]$ ；如果  $MA_1^{i,new} \cap MA_1^a = \emptyset$ ，则随机  $MA_1^a$  内元素替换  $MA_1^{i,new}$  内元素，同时，对  $F_i^{new}$  内相应编码位进行重新赋值。当  $MA_1$  完成路径节点更新后，同理，对剩余  $M-2$  个 MA 路径依次进行路径节点变换 ( $MA_M$  不进行插值学习处理操作)，最终得到新的个体  $F_i^{NEW}$ 。(NEW 表示对个体进行操作)。

**Step3** 重新排序。个体编码位不仅对应节点编号，而且决定了节点在路径中的先后访问顺序。当得到  $F_i^{NEW}$  后，参考  $F_a$  依次对  $MA_g$  内节点进行调整，使  $F_i^{NEW}$  与  $F_a$  同一  $MA_g$  内相同节点具有相同的访问顺序。

**Step4** 适应度值判定。如果  $f(F_i^{NEW})$  优于  $f(F_i)$ ，则用  $F_i^{NEW}$  替代  $F_i$ ，否则，随机生成新的个体进行替代。

**定义 11** 取插值学习更新算子“ $\otimes$ ”，定义插值学习更新机制为

$$F_i^{NEW} = \begin{cases} F_i \otimes F_a, & F_i \otimes F_a < F_i \\ rand(F_i), & \text{其他} \end{cases} \quad (16)$$

变异交换更新机制。同其他智能算法一样，在算法迭代后期，随着种群样本多样性降低，算法容易产生进化停滞现象。为了进一步提高 MDSSO 寻优能力，对种群部分个体执行变异交换更新策略：对于个体  $F_i$ ，随机选取 2 个相邻节点  $SN_w$ 、 $SN_{w+1}$  ( $w \in \{1, 2, \dots, W-1\}$ )，节点编号分别为  $S_w = w$ 、 $S_{w+1} = w+1$ ，找到 2 个节点在  $F_i$  编码中的位置，即确定  $x$ ， $y$  使  $F_{ix}[1] = w$ 、 $F_{iy}[1] = w+1$ 。当  $F_{ix}[2] \neq F_{iy}[2]$  时，将  $SN_w$  加入  $SN_{w+1}$  所在 MA 路径中，得到新的个体  $F_{i,new}$ 。

**定义 12** 取变异交换更新算子“ $\leftrightarrow$ ”，定义变异交换更新机制为

$$F_i^{NEW} = F_i(x \leftrightarrow y) \quad (17)$$

如果  $f(F_i^{NEW})$  优于  $f(F_i)$ ，则用  $F_i^{NEW}$  替代  $F_i$ ，否则  $F_i$  保持不变。

算法计算复杂度。MDSSO 种群初始化计算复杂度为  $O(F)$  ( $F$  为种群个数)，插值学习更新机制计算复杂度为  $O(FlbW)$ ，变异交换更新机制进化计算复杂度为  $O(FlbW)$ ，因此，算法每迭代一次算法计算复杂度为  $O(F) + 2O(FlbW)$ ，当种群完成  $T_{max}$  次迭代后，MDSSO 计算复杂度为  $T_{max}(O(F) + 2O(FlbW))$ 。

## 4.2 MDSSO-MIP 算法实现

MDSSO 在节点休眠机制的基础上，以评价函数  $f(G)$  为多目标优化函数，通过迭代更新实现对 MIP 问题求解。在粒子更新过程中，蜘蛛个体随机选取  $P^*$  集合内最优解作为学习对象，并按照插值学习更新机制进行更新；同时，选取  $P^*$  内少部分最优解并根据变异交换更新机制进行更新。

MDSSO-MIP 算法实现流程如下所示。

//算法初始化

1) 初始化 MDSSO-MIP 有关参数，并设置算法终止条件。

2) 根据定义 9 和定义 10 对 Pareto 最优解集初始化。

//网络工作节点确定

3) 接收 sink 指令，根据节点休眠机制，确定网络中工作节点相关信息。

While ( $t \leq T_{max}$ ) do

{

//蜘蛛种群初始化

4) 根据网络工作节点相关信息和蜘蛛编码方式对蜘蛛种群初始化。

//蜘蛛个体更新

5) 随机选取  $P^*$  集合内最优解作为蜘蛛个体学习对象。

6) 根据式(16)分别对雌性和雄性蜘蛛个体进行更新。

7) 雌性和雄性蜘蛛执行基本 SSO 算法分配操作。

// Pareto 最优解集更新

8) 更新 Pareto 最优解  $F_{new}^*(t)$ , 并判定  $F_{new}^*(t)$  能否进入  $P^*$ 。

9) 选取  $P^*$  内少部分最优解, 并根据式(17)进行更新。

10) 根据 Pareto 最优解集规模控制策略动态调整  $P^*$  规模。

11)  $t \leftarrow t + 1$

}

12) 程序结束, 选取 Pareto 最优解集中能耗最小解作为 MIP 最终方案。

13) sink 派出 MA 执行监控数据采集任务, 并根据网络覆盖要求和节点能耗确定是否发出更换工作节点指令。

## 5 实验仿真

在  $45\text{ m} \times 30\text{ m}$  监控区域内随机部署  $W = 150$  个传感器, 网络设置一个 sink 节点, sink 派出  $M = 4$  个 MA 访问监控节点。MDSSO-MIP 相关参数设置如下:  $\rho = 0.7$ 、 $r = 0.8$ 、 $P_{\min} = 85\%$ 、 $\partial_{\max} = 50$ 、 $R_{\max} = 5\text{ m}$ , MAWSN 及传感器节点指标参考文献[6]。参考定义 3 和定义 4, 采用网络总能耗  $E_{\text{total}}$ 、网络生存时间  $T_{\text{net}}$ 、任务延迟时间  $t_{\text{re}}$  以及延迟能耗积  $Et_{\text{net}}$  作为算法性能评价指标。

$$t_{\text{re}} = \max(t_{MA_i}), i = 1, 2, \dots, M \quad (18)$$

$$Et_{\text{net}} = E_{\text{total}} t_{\text{re}} \quad (19)$$

其中,  $t_{MA_i}$  为 MA 执行一次任务所需工作时间。

### 5.1 实例仿真

在 OPNET 平台上, 采用 MDSSO-MIP 算法进行实例仿真, 将仿真实验设定为 2 次执行监控节点数据采集任务。MDSSO-MIP 在确定工作节点的基

础上最终给出 MIP 方案。表 1 给出了 MDSSO-MIP 性能指标, 图 3 给出了不同阶段多 MA 路径方案。

表 1 MDSSO-MIP 性能指标

阶段	$E_{\text{total}}/\text{J}$	$T_{\text{net}}/\text{s}$	$t_{\text{re}}/\text{s}$	$Et_{\text{net}}/(\text{J}\cdot\text{s})$	网络覆盖率
1	0.83	$1.32 \times 10^{-6}$	0.13	0.11	93%
2	0.95	$6.4 \times 10^{-5}$	0.18	0.17	87%

从图 3 可以看出, 网络只需要 25 个左右节点处于工作状态就能够满足覆盖要求, 而且 MDSSO-MIP 给出了合理 MA 访问路径。

### 5.2 对比实验

#### 1) MDSSO-MIP 参数设置实验

种群规模  $P$  和最大迭代次数  $T_{\max}$  是 MDSSO 重要参数, 分别设置不同  $P$ 、 $T_{\max}$  取值, 以验证对网络能耗  $E_{\text{total}}$  影响, 图 4 给出了仿真结果。

从仿真结果可以看出, 随着  $P$  取值变大,  $E_{\text{total}}$  逐渐降低, 但是当  $P \geq 120$  时,  $E_{\text{total}}$  几乎没有发生变化; 对于  $T_{\max}$ ,  $T_{\max} = 350$  是  $E_{\text{total}}$  变化趋势临界点。

#### 2) 对比实验结果分析

为了进一步分析 MDSSO-MIP 性能, 选取 GA-MIP<sup>[8]</sup>、LCF<sup>[16]</sup>和 DMP-MIP<sup>[6]</sup>进行对比实验, 其中, LCF 为单移动 agent 路径规划方案, GA-MIP 与 DMP-MIP 为基于智能优化算法的多移动 agent 路径规划方案。对比实验中, 分别设置不同监控节点数量, 每种算法重复实验 100 次, 取评价指标均值进行比较。图 5 给出了 4 种算法评价指标对比结果。

从图 5 可以看出, 除了  $E_{\text{total}}$  指标 MDSSO-MIP 高于 LCF 外, 对于其他 3 个指标, MDSSO-MIP 要优于 GA-MIP、LCF 和 DMP-MIP, 而且网络总能耗降低了 15%左右, 生存期提高了 23%左右。这是因为 MDSSO-MIP 采用多 MA 策略, 降低了网络总能耗, 而且引入网络负载均衡概念, 进一步延长了网络生存时间, 而 LCF 为单移动 agent 路径规划方案, 随着网络规模增加, 网络延时问题逐渐加大。

为了分析 MDSSO 收敛性能, 定义归一适应度值为  $B_i$ , 即  $B_i$  每次迭代种群 Pareto 最优解与算法收敛时 Pareto 最优解的函数适应度比值为

$$B_i = \frac{f[F^*(t)]}{f[F^*]} \quad (20)$$

图 6 给出了 MDSSO 收敛曲线 (选取网络总能耗优化函数)。

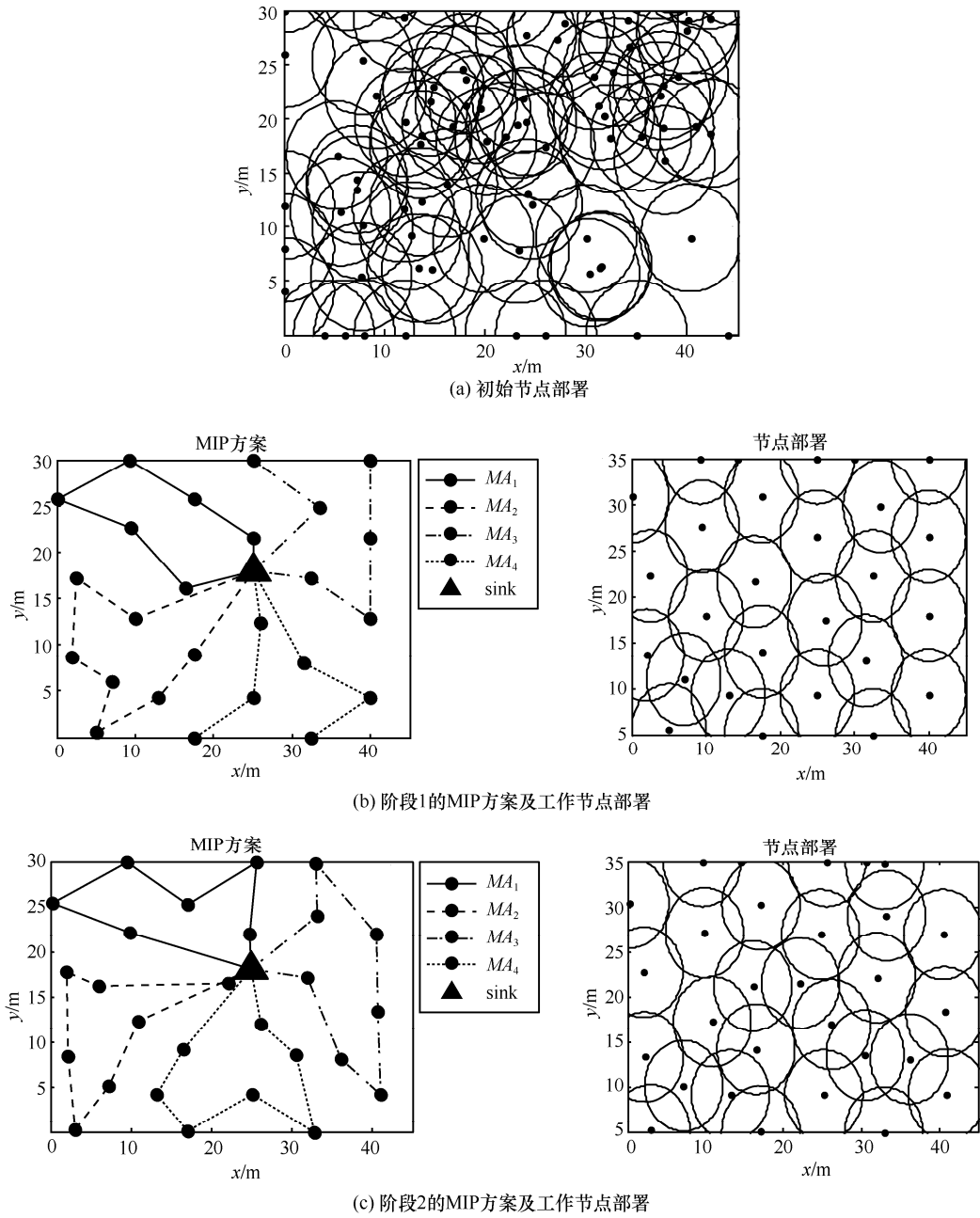


图 3 不同阶段多 MA 路径方案

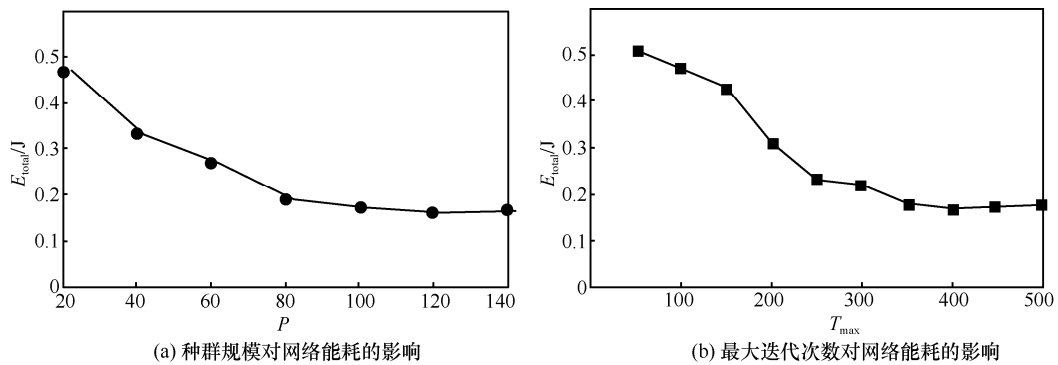


图 4 参数设置对算法性能影响

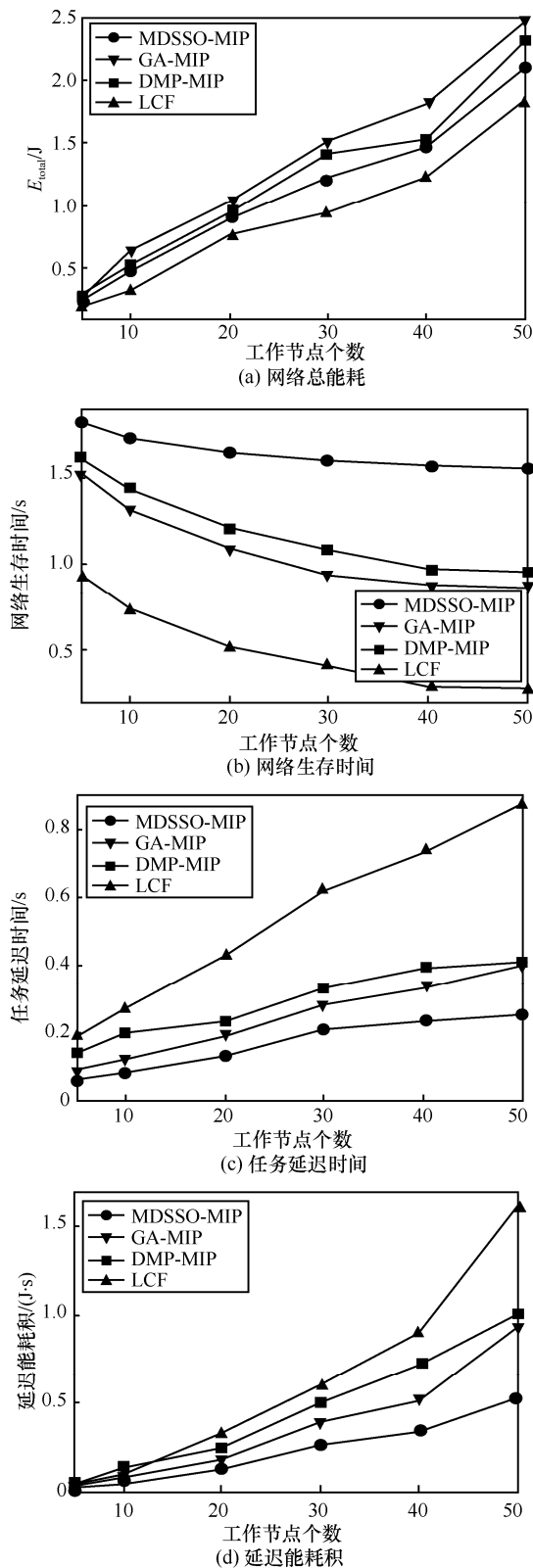


图 5 4 种算法评价指标对比结果

从图 6 可以看出, 随着迭代次数增加, 算法收敛精度逐渐提高, 特别是当迭代次数增加到一定次数时, 算法已经收敛于最优解; 随着监控节点数量

增加, MDSSO 收敛迭代次数也随之增加, 但是, MDSSO 都能够以较少迭代次数收敛于全局最优解, 即使节点个数增加到 50 个, 算法只需要 360 次迭代即可完成目标函数优化, 这也表明 MDSSO-MIP 具有良好的问题求解性能。

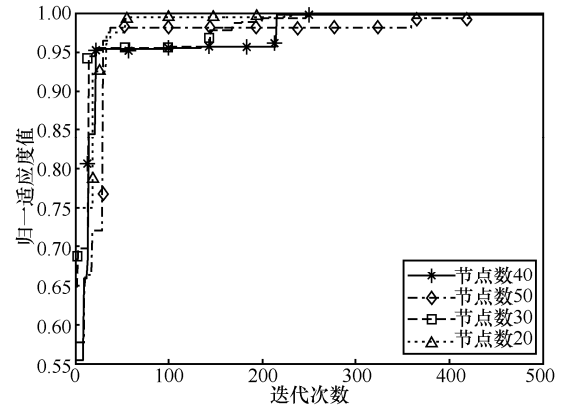


图 6 不同工作节点下的 MDSSO 收敛曲线

### 6 结束语

本文提出了一种基于网络覆盖和多目标离散群集蜘蛛算法的多移动 agent 规划算法。分别对网络节点休眠机制、多目标离散群集蜘蛛算法设计以及多移动 agent 规划实现问题进行了研究, 研究结果对 MIP 问题降低网络总能耗, 平衡负载能耗等具有很好的借鉴意义, 仿真结果表明该算法与同类算法相比, 具有一定优势。下一步, 可以进一步研究提高算法抵抗安全威胁、强烈干扰的能力。

### 参考文献:

- [1] 夏俊, 凌培亮, 虞丽娟, 等. 基于量子遗传算法的无线传感网络路由优化[J]. 同济大学学报(自然科学版), 2015, 43(7):1097-1103. XIA J, LING P L, YU L J, et al. Optimization of wireless sensor network routing based on quantum genetic algorithm[J]. Journal of Tongji University (Natural Science Edition), 2015, 43 (7): 1097-1103.
- [2] 侯建军, 朱明强, 刘颖. 一种强背景噪声下的 WSN 目标定位算法[J]. 北京理工大学学报, 2016, 36(5):535-540. HOU J J, ZHU M Q, LIU Y. A WSN target localization algorithm under strong background noise[J]. Journal of Beijing Institute of Technology, 2016, 36(5): 535-540.
- [3] SALEEM F, MOEEN Y, BEHZAD M, et al. IDDR: improved density controlled divide-and-rule scheme for energy efficient routing in wireless sensor networks[J]. Procedia Computer Science, 2014, 34: 212-219.
- [4] 杨挺, 杨青, 唐勇. WSN 的移动 agent 随机模式与分析[J]. 电子科技大学学报, 2015, 44(3): 439-444. YANG T, YANG Q, TANG Y. WSN mobile agent stochastic model and analysis[J]. Journal of University of Electronic Science and Tech-

- nology, 2015, 44 (3): 439-444.
- [5] 苏金树, 郭文忠, 余朝龙, 等. 负载均衡感知的无线传感器网络容错分簇算法[J]. 计算机学报, 2014, 37(2):445-456.  
SU J S, GUO W Z, YU Z L, et al. Wireless sensor network fault tolerant load balancing aware clustering algorithm[J]. Journal of Computer, 2014, 37(2): 445-456.
- [6] 史霄波, 张引, 赵杉, 等. 基于离散多目标优化粒子群算法的多移动代理协作规划[J].通信学报, 2016, 37(6): 29-37.  
SHI X B, ZHANG Y, ZHAO S, et al. Cooperative programming of multiple mobile agents based on discrete multi-objective optimization particle swarm optimization[J]. Journal on Communications, 2016, 37(6): 29-37.
- [7] CHEN M, LEUNG V, MAO S W, et al. Energy-efficient itinerary planning for mobile agents in wireless sensor networks[C]//IEEE International Conference on Communications (ICC'09). 2009: 1-5.
- [8] CAI W, CHEN M, HARA T, et al. A genetic algorithm approach to multi-agent itinerary planning in wireless sensor networks[J]. Mobile Networks and Applications, 2011, 16(6): 782-793.
- [9] KONSTANTOPOULOS C, MPITZIOPOULOS A, GAVALAS D, et al. Effective determination of mobile agent itineraries for data aggregation on sensor networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(12): 1679-1693.
- [10] CHEN M, CAI W, GONZALEZ S, et al. Balanced itinerary planning for multiple mobile agents in wireless sensor networks[M]//Ad Hoc Networks. Springer Berlin Heidelberg, 2010: 416-428.
- [11] 张亚明, 史浩山, 刘燕, 等. WSN 中基于蚁群模拟退火算法的移动 agent 访问路径规划[J].西北工业大学学报, 2012, 30(5): 629-635.  
ZHANG Y M, SHI H S, LIU Y, et al. WSN mobile agent ant colony simulated annealing algorithm based on access path planning[J]. Journal of Northwestern Polytechnical University, 2012, 30 (5): 629-635.
- [12] 刘洲洲, 王福豹, 张克旺. 基于改进萤火虫优化算法的 WSN 覆盖优化分析[J].传感技术学报, 2013, 26(5): 675-682.  
LIU Z Z, WANG F B, ZHANG K W. Optimization analysis of WSN cover based on improved firefly optimization algorithm[J]. Chinese Journal of Sensors and Actuators, 2013, 26(5): 675-682.
- [13] CUEVAS E, CIENFUEGOS M, ZALDIVA D, et al. A swarm optimization algorithm inspired in the behavior of the social spider[J]. Expert System with Applications, 2013, 40(16): 6374-6384.
- [14] 王艳娇, 李晓杰, 肖婧. 基于动态学习策略的群集蜘蛛优化算法[J]. 控制与决策, 2015, 30(9): 1575-1582.  
WANG Y J, LI X J, XIAO J. The cluster optimization algorithm based on spider dynamic learning strategy[J]. Control and Decision, 2015, 30 (9): 1575-1582.
- [15] 王文川, 雷冠军, 刘惠敏, 等. 基于群居蜘蛛优化算法的自适应数值积分皮尔逊-III 型曲线参数估计[J]. 应用基础与工程科学学报, 2015, 23(S1): 122-123.  
WANG W C, LEI G J, LIU H M, et al. Adaptive numerical integration based on social spider optimization algorithm parameter estimation of Pearson-III curve[J]. Applied Foundation and Journal of Engineering Science, 2015, 23 (S1): 122-123.
- [16] QI H R, WANG F Y. Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks[C]//Proceedings of the IEEE. 2001: 147-153.

#### 作者简介:



刘洲洲 (1981-), 男, 陕西延安人, 西北工业大学博士后, 西安航空学院副教授, 主要研究方向为无线传感器网络、智能优化算法和不确定性推理。



李士宁 (1967-), 男, 陕西延安人, 博士, 西北工业大学教授、博士生导师, 主要研究方向为智能计算、无线传感器网络。