

# 无线传感器/执行器网络任务动态调度策略

易 军<sup>1,2</sup>, 石为人<sup>1</sup>, 唐云建<sup>1,3</sup>, 许 磊<sup>1</sup>

(1. 重庆大学自动化学院, 重庆 400044; 2. 重庆科技学院电子信息工程学院, 重庆 400044;  
3. 中国科学院深圳先进技术研究院, 广东深圳 518055)

**摘 要:** 针对任务在各执行器的协作问题, 提出一种动态调度策略, 根据执行器节点的剩余能量和工作状态, 利用混合模拟退火的微粒群算法, 在任务时效期内, 统一安排各任务在执行器上的执行周期, 最小化最大完成时间. 仿真结果表明, 算法具有良好的收敛性能, 各执行器的任务完成响应时间和能耗均衡情况均得到改善.

**关键词:** 无线传感器/执行器网络; 任务调度; 微粒群算法; 模拟退火算法

**中图分类号:** TP393      **文献标识码:** A      **文章编号:** 0372-2112 (2010) 06-1239-06

## A Dynamic Task Scheduling for Wireless Sensor and Actuator Networks

YI Jun<sup>1,2</sup>, SHI Wei-ren<sup>1</sup>, TANG Yun-jian<sup>1,3</sup>, XU Lei<sup>1</sup>

(1. College of Automation, Chongqing University, Chongqing 400044, China;  
2. College of Electronic & Information Engineering, Chongqing University of Science & Technology, Chongqing 400044, China;  
3. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China)

**Abstract:** A dynamic task scheduling approach based on particle swarm optimization (PSO) and simulated annealing (SA) technique for wireless sensor and actuator networks is proposed to solve the execution problem of tasks collaboratively among actuators. The purpose of approach is minimizing the maximum response time in the actuators subject to residual energy constraints and schedule execution period of each task operation within given time. Simulation results have shown that the proposed hybrid approach is of high convergence speed and good performance between task response time and balancing the energy dissipation among actuators.

**Key words:** wireless sensor and actuator network; task scheduling; particle swarm optimization; simulated annealing

### 1 引言

无线传感器网络(Wireless Sensor Networks, WSNs)的发展趋势是在布置大量传感器节点的基础上融合多个分布式设备, 对感知的信息进行处理并作出相应的动作. 无线传感器/执行器网络(Wireless Sensor and Actuator Networks, WSANs)是近年来发展起来的一种新型、自组织的无线网络<sup>[1]</sup>, 由大量传感器和少量执行器组成, 完成各种分布式感知任务和执行任务. 该网络可以整合到军事、灾害、人体监护等各种应用中, 如传感器检测到火灾发生后, 执行器发出声光报警信息和洒水动作, 避免火势蔓延. 与无线传感器网络相比, 无线传感器/执行器网络具有两个重要的特点: 实时性和协同合作. 前者需要执行器对感知信息迅速做出反应, 后者由于网络中有两种异构节点, 存在传感器与执行器的协作和执行器与执行器之间的合作问题. 特别是执行器与执行器之间, 由哪些执行器来执行哪些任务? 它们之间如何协调?

执行某个任务需要多少执行器参与? 怎样才能保证全网执行器任务完成时间最短? 这些都是 WSANs 面临的挑战<sup>[1,2]</sup>.

解决执行器之间协作问题一般有两条思路: 一种是执行器之间互相谈判协商, 另一种是在任务时效期内, 由汇聚节点统一安排任务在各执行器上执行. 本文提出一种任务动态调度算法, 根据执行器节点的剩余能量和工作状态, 利用混合模拟退火的微粒群算法, 统一安排各任务在节点的执行周期, 最小化最大完成时间, 既保证任务结果的实时反馈, 又节约节点的能量开销.

### 2 相关工作

传统的高性能计算领域通常将任务划分为映射和调度两个问题进行研究, 这样的思想也被借鉴到无线传感器网络中. Heemin Park 等在文献[3]提出适用于 Ad-hoc 传感器网络的能量有效型任务分配框架. 该框架将优化任务的分配和迁移用最小化代价函数来描述, 函数

反映了全网的节点能耗、时延和给定约束. 整个算法包括两个阶段: 在设计阶段进行集中式任务分解转化和分配; 在执行任务阶段, 各节点执行分布式任务迁移策略. Younis M 等提出的一种任务分配优化算法<sup>[4]</sup>, 主要关注全局任务在簇头上的分配问题, 通过提高簇头的能量有效性, 使全网的生存期得到延长. 任务分配问题被建模成 0-1 非线性优化问题, 目标函数是最大化所有簇头的剩余生存期的权重之和. 同时考虑通信时间、带宽和能量的约束. 文献<sup>[5]</sup>设计了一种在线任务调度机制, 用于在簇内周期性应用任务中分配网络资源, 根据预先设定的任务执行频率上限, 优化每个节点的任务频率. 然而, 该机制并没有把任务映射到传感器节点上. 文献<sup>[6]</sup>提出一种分布式计算构架, 将任务分为两级: 传感器节点的低级感知任务和簇头的高级处理任务. 但是高级处理任务可能超出簇头的计算能力, 而且算法的通用性不强.

近年来, 在无线传感器网络<sup>[7-9]</sup>和移动计算<sup>[10]</sup>领域, 任务的映射和调度常被合二为一. EcoMapS 算法<sup>[8]</sup>在能量约束条件下, 将单跳簇网络的通讯和计算任务, 映射和调度同时进行. 然而, 该算法并不能保证应用的时限. RT-Maps<sup>[7]</sup>算法则在保证调度时限的前提下, 最小化节点能耗. Yang Yu 等人提出了一种基于能量平衡的任务分配算法 EBTA<sup>[9]</sup>, 该算法用于解决单跳簇内的通信和计算任务, 提出一个用于评价任务的时间和能量代价的整数线性规划公式, 设计了三阶段启发算法. 这些算法都只能应用于单跳簇网络, 限制了算法的通用性. DCTMP 算法<sup>[11]</sup>利用多跳信道模型, 借助有向无环图将通信和计算任务的映射和调度结合起来, 是一种适用于视频传感器网络的实时任务分配算法.

上述算法都不是专门针对无线传感器/执行器网络的, 有必要针对该网络特点提出新的任务调度算法.

### 3 问题描述

典型的无线传感器/执行器网络结构如图 1 所示, 为了避免所有传感器都与多个执行器通信, 消耗过多能量, 通常采用分簇结构, 传感器只与簇内的一个执行器进行通信.

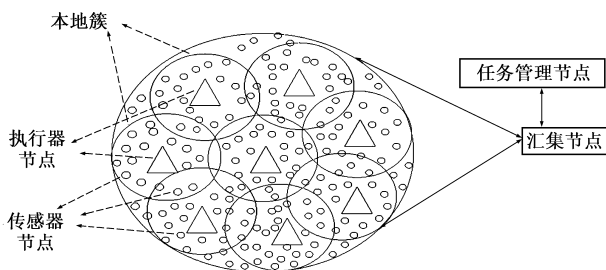


图1 无线传感器/执行器网络的典型架构

与传统的 WSNs 只具有智能感知功能相比, WSNs 不但能感知环境事件的发生, 而且能通过一个执行器或者多个执行器协同作业, 对事件进行相应的处理. 例如对于城市下水管道排水问题, 管网规模大, 覆盖面广, 情况复杂. 由于设计或者其他原因, 一旦发生暴雨, 常容易引起路段积水, 严重者甚至威胁生命和财产安全. 利用部署大量传感器进行检测, 发生溢流事件后, 通过无线网络报告管理人员, 统一安排相应管道执行器(排水阀)及时排水, 杜绝积水问题发生. 典型应用如下图 SCADA 网络<sup>[12]</sup>:

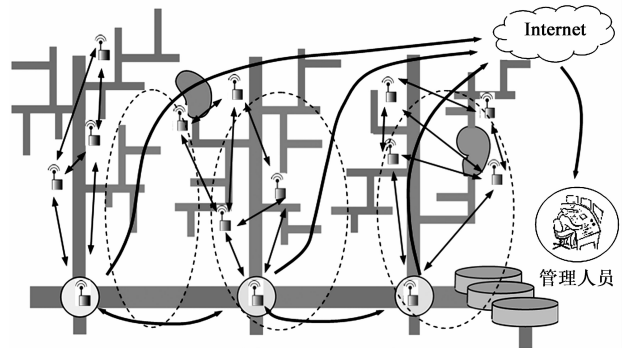


图2 SCADA网络

典型的下水管执行器任务由多个执行器(排水阀)打开(关闭)操作组成, 各个执行器阀门开启量也不相同, 且所有操作可能需要符合一定顺序, 以保证水流顺利排出, 如图 3 所示.

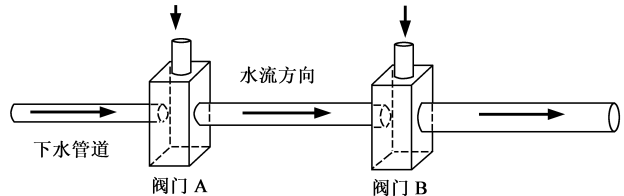


图3 典型下水管执行器任务

类似的还有利用 WSNs 解决大范围智能交通疏导问题、大规模污水处理流程问题等. 这里, 我们将完成一个特定目标的行为定义为一个任务, 包括通信、数据处理、执行动作等行为.

**定义 1** 一个任务在一个执行器上的加工称为一次操作, 一个任务可以分为多个操作. 操作集  $O = \{o_1, o_2, \dots, o_n\}$  由所有  $n$  个任务的操作子集构成, 其中每个子集表示一个任务在所有执行器上的执行.

**定义 2** 对于任务  $t$ , 记其所有  $o_i$  个操作的序号为  $l_{i-1}+1, \dots, l_{i-1}+o_i$ , 其中  $l_{i-1} = \sum_{i=1}^{i-1} o_i$  表示前  $t-1$  个任务的操作总数. 显然,  $n$  个任务的操作集  $o = \sum_{i=1}^n o_i$ , 同时任务  $t$  的所有  $o_i$  个操作必须符合一定的执行顺序, 即任务约束条件.

**定义 3** 一个由单个执行器完成的任务,称为单执行器任务(Single-Actuator Task, SAT);如果一个任务由多个执行器共同完成并具有任务约束条件,则称为多执行器任务(Multi-Actuator Task, MAT)。

**推论** 如果一个任务由多个执行器共同完成但不具有任务约束条件,则可划分为若干独立的子任务独立完成,则该任务仍然属于单执行器任务。

考虑  $n$  个任务在  $m$  个执行器上的执行过程,每个任务在各执行器上的完成时间已知,并且每个任务必须按照事先规定的顺序完成其在各执行器上执行,即任务约束条件.如果一个任务只需一个执行器完成,则该任务在其他执行器的执行时间用零表示.要求确定在每个执行器上执行的所有任务的顺序或开工与完工时间,使得在符合各任务的约束条件的前提下最优化最大完成时间.这里假设:一个执行器同时只能执行一个操作;任务没有抢先执行的特权;每一项操作的执行时间和能耗可以估算;操作结果的数据报文格式一致,即传输结果的通信能耗一致。

### 3.1 最小化最大完成时间

设  $p_{i,j}$  为任务  $i$  在执行器  $j$  上的执行时间,  $C(j_i, k)$  为任务  $j_i$  在执行器  $k$  的执行完成时间,  $\pi = (j_1, j_2, \dots, j_n)$  为  $n$  个任务的一次排序,  $\Pi$  为  $n$  个任务的所有可能排序集合.一般地:无线传感器/执行器网络中一个操作的执行时间  $t_{\text{oper}}$  包括:指令下达时间  $t_{\text{comm}}$ 、数据采集/执行动作时间  $t_{\text{colle}}$ 、数据处理时间  $t_{\text{proc}}$ 。

$$p_{i,j} = t_{\text{oper}} = t_{\text{comm}} + t_{\text{colle}} + t_{\text{proc}} \quad (1)$$

则各任务在每个执行器上完成时间的数学公式可描述如下:

$$C(j_1, 1) = p_{j_1, 1} \quad (2)$$

$$C(j_i, 1) = C(j_{i-1}, 1) + p_{j_i, 1} x_{j_i, 1}, i = 2, \dots, n \quad (3)$$

$$C(j_i, k) = C(j_1, k-1) + p_{j_i, k} x_{j_i, k}, k = 2, \dots, m \quad (4)$$

其中,  $x_{j_i, k} = 1$  表示该执行器  $k$  被选中执行  $j_i$  任务,  $x_{j_i, k} = 0$  表示没被选中.式(2)表示执行器 1 执行任务  $j_1$  的完成时间等于执行时间.式(3)表示执行器 1 执行任务  $j_i$  的完成时间等于执行任务  $j_{i-1}$  的完成时间加上  $j_i$  任务的执行时间.同理:式(4)表示执行器  $k$  执行任务  $j_i$  的完成时间等于执行器  $k-1$  执行  $j_1$  的完成时间加上执行器  $k$  执行  $j_i$  的时间。

$$C(j_i, k) = \max\{C(j_{i-1}, k), C(j_i, k-1)\} + p_{j_i, k} x_{j_i, k}, \\ i = 2, \dots, n; k = 2, \dots, m \quad (5)$$

式(5)表示执行器  $k$  执行任务  $j_i$  必须在执行器  $k$  已经执行完上一个任务  $j_{i-1}$  之后,同时任务  $j_i$  在上一个执行器  $k-1$  的执行也已经完成的基础上进行。

则最大完成时间:

$$C_{\max}(\pi) = C(j_n, m) \quad (6)$$

最小化最大完成时间的调度方案:

$$\pi^* = \arg\{C_{\max}(\pi) = C(j_n, m)\} \rightarrow \min, \forall \pi \in \Pi \quad (7)$$

### 3.2 执行器剩余能量约束

设  $P(t)$  为单位时间内执行器执行操作所需能耗,  $\gamma l$  表示采集  $l$  长度数据所需能耗,则执行器  $k$  所需执行能耗:

$$E_{k-\text{action}} = \sum_{j=1}^n \sum_{i=1}^m (P(t) p_{j_i, k} + \gamma l) x_{j_i, k} \quad (8)$$

可行分配方案必须考虑到执行器剩余能量的约束,每个执行器的剩余能量不但要足以完成所有操作序列,还要将执行结果数据传给下一执行器或汇集节点.根据文献[13]发射硬件能耗模型,  $r$  为执行器通信距离,  $l$  表示帧长,则传输和接收能耗公式为:

$$\begin{cases} E_{\text{tx}}(r, l) = (\alpha r^{n'} + \beta) l \\ E_{\text{rx}}(l) = \beta l \end{cases} \quad (9)$$

$\alpha r^{n'}$  取决于在距离  $r$  传输的发射功率,  $\beta$  是发射电路(如 PLLs, VCOs) 能耗,  $n'$  为信道衰减倍数,取决于环境.则一个执行器  $k$  所有操作所需的总能耗:

$$E_{k-\text{all}} = \sum_{j=1}^n \sum_{i=1}^m (P(t) p_{j_i, k} + \gamma l + (\alpha r^{n'} + \beta) l + \beta l) x_{j_i, k} \quad (10)$$

执行器  $k$  的能量约束条件:

$$E_{k-\text{all}} \leq E_{k-\text{rest}}, k = 1, 2, \dots, m \quad (11)$$

其中  $E_{k-\text{rest}}$  为执行器  $k$  的剩余能量。

## 4 算法描述

解决上述带能量约束条件的最小化最大完成时间问题,我们采用微粒群优化算法,利用该算法全局搜索能力强的特点,与模拟退火算法的局部改良能力相结合,提高微粒群算法的搜索精度,寻找全局最优解.由于任务分配问题需要映射到微粒的位置矢量上,所以首先需要进行编码,同时利用可行性规则对能量约束条件进行处理,详细介绍如下:

### 4.1 微粒群进化搜索

微粒群优化算法(PSO)<sup>[14]</sup>是一种进化计算技术,由 Eberhart 博士和 Kennedy 博士发明. PSO 首先在可行解空间和速度空间随机确定微粒的初始位置和初始速度,其中位置用于表征问题解.  $d$  维搜索空间的第  $i$  个微粒的位置和速度可分别表示为  $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$  和  $V_i = [v_{i,1}, v_{i,2}, \dots, v_{i,d}]$ . 通过评价各微粒的目标函数,确定  $t$  时刻每个微粒所经过的最佳位置 ( $p_{\text{best}}$ )  $P_i = [p_{i,1}, p_{i,2}, \dots, p_{i,d}]$  以及群体最佳位置 ( $g_{\text{best}}$ )  $P_g$ , 再按如下公式分别更新各微粒的速度和位置:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_1 [p_{i,j} - x_{i,j}(t)]$$

$$+ c_2 r_2 [p_{g,j} - x_{i,j}(t)] \quad (12)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), j = 1, \dots, d \quad (13)$$

其中,  $w$  为惯性权因子,  $c_1$  和  $c_2$  为正的加速常数,  $r_1$  和  $r_2$  为在 0 到 1 之间均匀分布的随机数.

#### 4.2 基于优先权的编码

基于优先权的编码<sup>[15,16]</sup>将微粒的位置矢量用一个长度为  $n \times m$  的优先分配序表示, 每个微粒位置值代表与之相关的操作优先权, 则对于  $n$  个任务在  $m$  个执行器的调度问题, 从第 1 维分量到第  $n \times m$  维分量, 微粒  $k$  的位置矢量就可以用一个  $n \times m$  的矩阵  $X_k$  表示:

$$X_k = \begin{bmatrix} x_{11}^k & x_{12}^k & \dots & x_{1m}^k \\ x_{21}^k & x_{22}^k & \dots & x_{2m}^k \\ \dots & \dots & \dots & \dots \\ x_{n1}^k & x_{n2}^k & \dots & x_{nm}^k \end{bmatrix}$$

其中,  $x_{ij}^k$  为一个实数, 代表操作  $o_{ij}$  的优先权. 文献[15]给出了利用 Giffler-Thompson 算法将  $X_k$  解码为一个活动调度的具体过程. 在该编码方式下, PSO 算法是在连续空间上进行搜索, 无需修改标准的 PSO 操作.

#### 4.3 基于可行性规则的约束条件处理

可行性规则<sup>[17]</sup>是一种简单实用的约束处理技术, 该方法无须额外参数, 算法效率高. 假设  $p_i^t$  为种群中第  $i$  个微粒在第  $t$  代时的历史最优位置,  $x_i^{t+1}$  表示该微粒在第  $t+1$  代时所在位置, 若以下任意一种情况发生, 就将当前位置设为历史最优, 即执行  $p_i^{t+1} = x_i^{t+1}$ , 否则不变, 即  $p_i^{t+1} = p_i^t$ .

(1)  $p_i^t$  不可行, 而  $x_i^{t+1}$  可行;

(2)  $p_i^t$  和  $x_i^{t+1}$  均可行, 且  $x_i^{t+1}$  的目标值优于  $p_i^t$  的目标值, 即:  $f(x_i^{t+1}) < f(p_i^t)$ ;

(3)  $p_i^t$  和  $x_i^{t+1}$  均不可行, 且  $\text{viol}(x_i^{t+1}) < \text{viol}(p_i^t)$ . 其中  $\text{viol}(x)$  是不可行解约束违反量的函数.

#### 4.4 基于模拟退火局部搜索的混合 PSO 算法流程

标准 PSO 算法的全局搜索能力强, 收敛速度快, 因此容易早熟收敛. 本文利用基于 SA 的局部搜索能力强, 通过概率突跳, 可有效避免搜索过程陷于局部极小. 先通过 PSO 进化得到一组具有较好质量和分散度的初始解, 再采用 SA 机制对这些解进行局部邻域搜索, 有利于对优良解的局部改良. 对于第  $t$  代种群的最优位置  $p_g^t$ , 采用基于可行性规则进行更新, 局部搜索重要参数描述如下:

SA 算法采用下式产生一个新解:

$x' = p_g^t + \eta \times (X_{\max} - X_{\min}) \times N(0, 1)$ , 其中  $\eta$  为搜索步长,  $N(0, 1)$  为服从均值为 0 方差为 1 的高斯分布随机数,  $X_{\max}$ 、 $X_{\min}$  为变量上下界. 则产生的新解  $x'$  替代  $x$  成为新的当前解的接受概率  $p_a$

$$p_a = \begin{cases} 1, & \text{if } x' \text{ (feasible)}; p_g^t \text{ (unfeasible)} \\ 0, & \text{if } x' \text{ (unfeasible)}; p_g^t \text{ (feasible)} \\ \min\{1, \exp[(f(p_g^t) - f(x'))/t_k]\}, & \text{if } x' \text{ (feasible)}; p_g^t \text{ (feasible)} \\ \min\{1, \exp[(\text{viol}(p_g^t) - \text{viol}(x'))/t_k]\}, & \text{if } x' \text{ (unfeasible)}; p_g^t \text{ (unfeasible)} \end{cases}$$

若  $p_a \geq U(0, 1)$ , 则  $p_g^t = x'$ . 若局部搜索步长大于事先设定的最大局部搜索步长  $L$ , 则输出  $p_g^t$  作为种群新的最优位置, 否则重复搜索.

初温:  $t_0 = -(f_{\max} - f_{\min})/\ln(p_\tau)$  其中  $p_\tau$  为初始接受概率, 取 0.1,  $f_{\max}$  和  $f_{\min}$  分别是初始种群中最佳个体和最差个体的目标值.

温度更新函数:  $t_{k+1} = \lambda t_k$ , 其中,  $0 < \lambda < 1$ .

抽样稳定准则: 按一定的步数抽样

算法终止准则: 迭代次数  $k$  等于最大进化迭代次数  $G_{\max}$ .

PSO 算法能够快速地搜索到较好解区域, 但收敛精度不高. 而 SA 具有较强的局部搜索能力, 但全局搜索能力弱. 算法如下:

初始阶段:

(1) 随机初始化  $M$  个微粒的位置和速度, 令  $k = 0$ ,  $t(k) = t_0$ ;

(2) 令各微粒的自身最佳位置  $p_{\text{best}}$  为其本身, 并根据可行性规则确定群体的最佳位置  $g_{\text{best}}$ ;

重复阶段:

(1) 判断迭代次数  $k$  等于最大进化迭代次数  $G_{\max}$ , 输出  $g_{\text{best}}$  对应的调度方案和目标值, 否则, 执行下一步;

(2) 对各微粒进行 PSO 进化搜索, 更新各微粒的速度和位置;

(3) 根据可行性规则评价各微粒的目标值;

(4) 更新各微粒的  $p_{\text{best}}$  和群体的  $g_{\text{best}}$ ;

(5) 对群体最佳位置应用基于 SA 的局部搜索策略;

(6) 更新温度  $t(k+1) = \lambda \cdot t(k)$ ,  $k = k + 1$ .

## 5 数值仿真研究

为了考察混合 PSO 算法解决执行器任务调度的性能, 利用两类典型 JSP 算例: 一类由 Fisher 和 Thompson<sup>[18]</sup>设计, 包括 FT06、FT10 和 FT20 这 3 个典型问题; 一类由 Lawrence<sup>[19]</sup>给出, 选取 6 个不同规模的问题, 考察最大完成时间和各执行器能耗均衡情况, 见表 2.

仿真硬件环境: Intel Pentium IV/2.2GHz/512MB RAM; 软件平台: Windows XP, MATLAB 7.0.

由表 2 可见, 算法对各算例均能获得最优解. 图 4 为算法收敛时间图.

表 1 算法参数表

PSO		SA		执行器通信参数	
				执行器空闲等待能耗	0 J
群体规模 $p_s$	30			通信距离 $r$	100m
惯性权因子 $w$	1.0	局部搜索执行概率 $p_b$	0.4	数据包长 $l$	128 byte
加速因子 $c_1, c_2$	2	与初温相关概率 $p_r$	0.1	发送/接收电路能耗 $\beta$	50 $nJ \cdot bit^{-1}$
微粒最小位置值 $x_{min}$	0	退温速率	0.9	放大倍数 $\alpha$	$0.0013 \mu J \cdot (bit \cdot m^{-2})^{-1}$
微粒最大位置值 $x_{max}$	4.0	终止条件参数 $L$	10	Two-ray ground 模型指数 $n$	4
微粒最小速度值 $v_{min}$	-4.0	温度更新系数 $\lambda$	0.95	采集能耗指数 $\gamma$	5 $nJ \cdot bit^{-1}$
微粒最大速度值 $v_{max}$	4.0	抽样步数	$n \times m$	单位时间内执行能耗 $P(t)$	5000 $nJ \cdot s^{-1}$

表 2 混合 PSO 算法的统计性能表

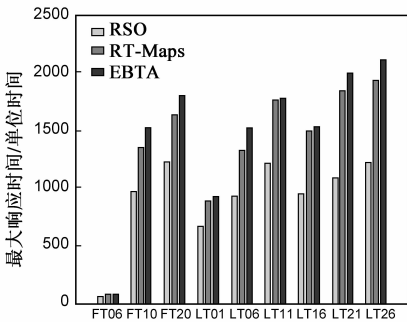
问题	$n, m$	已知最好解	带能量约束条件的混合 PSO 算法			
			最好解	平均值	最差解	平均时间(s)
FT06	6, 6	55	55	55	55	6.1
FT10	10, 10	930	930	956.4	968	154.3
FT20	20, 5	1165	1165	1185.3	1227	298.4
LA01	10, 5	666	666	666	666	15.7
LA06	15, 5	926	926	926	926	57.4
LA11	20, 5	1222	1222	1222.4	1223	141.3
LA16	10, 10	945	945	945.7	947	146.5
LA21	15, 10	1046	1046	1065.4	1089	724.8
LA26	20, 10	1218	1218	1218.6	1219	2359.1

从图 4 可见,各算例由于求解复杂度不同,其进化代数也各不相同,FT06 在进化 19 代即获得最优解,而 LT26 需要进化 151 代才获得最优解,算法总的收敛效果良好.考虑执行器剩余能量各不相同,见表 3,与算法 RT-Maps、EBTA 比较,考查任务完成后的响应时间和能耗均衡情况.

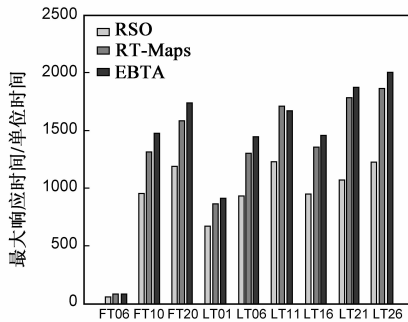
表 3 执行器剩余能量

执行器号/剩余能量(J)	1/1	2/0.3	3/1	4/0.5	5/2	6/0.1	7/0.2	8/0.5	9/0.7	10/1
--------------	-----	-------	-----	-------	-----	-------	-------	-------	-------	------

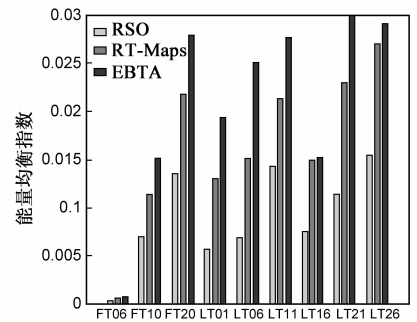
从图 5(a)、(b)观察到,本算法由于已最小化最大



(a) 最大响应时间对比



(b) 平均响应时间对比



(c) 能量均衡指数对比

图 5 各算法网络性能对比图

完成时间为优化目标,与算法 RT-Maps、EBTA 相比,无论是最大响应时间还是平均响应时间,时延都明显好于后两种算法.

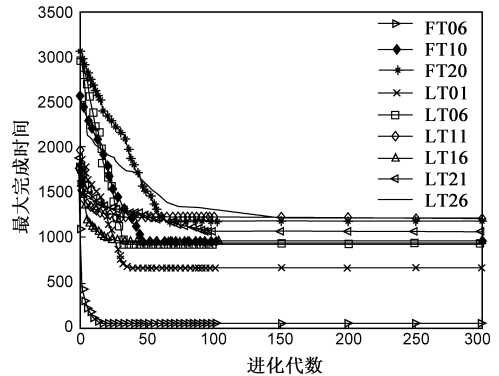


图 4 算法收敛图

为比较算法能耗均衡问题,构造了能量均衡指数:

$$C = \frac{\sum_{i=1}^m (E_{i-all} / E_{i-rest})}{m} \quad (14)$$

表示各执行器所消耗能量与剩余能量的平均比值,值越小,表示剩余能量多的执行器承担执行任务多,达到能量均衡的目的.从图 5(c)观察到,混合 PSO 算法在寻求可行解时充分考虑剩余能量问题,使执行器能耗更加均衡.

## 6 结论

无线传感器/执行器网络任务动态调度机制以最大完成时间为优化目标,考虑各执行器剩余能量约束,利用 PSO 算法与 SA 算法相结合,有效解决了各任务在各执行器上的动态调度问题.仿真结果表明,算法具有良好的收敛性能,各执行器的任务完成响应时间和能耗均衡情况均好于比较算法.

### 参考文献:

[1] I F Akyildiz, I H Kasimoglu. Wireless sensor and actor networks: research challenges [J]. Ad Hoc Networks Journal, 2004, 2(4): 351 - 367.

- [2] 林恺,赵海,等.一种基于能量预测的无线传感器网络分簇算法[J].电子学报,2008,36(4):824-828.  
LIN Kai, ZHAO Hai, et al. A clustering hierarchy arithmetic based on energy prediction for wireless sensor networks[J]. Acta Electronica Sinica, 2008, 36(4): 824-828. (in Chinese)
- [3] Heemin P, Mani B S. Energy-efficient task assignment framework for wireless sensor networks[R]. UC Los Angeles: The Berkeley Electronic Press, 2003.
- [4] Younis M, Akkaya K, Kunjithapatham A. Optimization of task allocation in a cluster-based sensor network[A]. Proceedings of the Eighth IEEE International Symposium on Computers and Communication[C]. Netherlands: IEEE Computer Press, 2003. 329-334.
- [5] S Giannecchini, M Caccamo, et al. Collaborative resource allocation in wireless sensor networks[A]. Proceedings of the 16th Euromicro Conference on Real-Time Systems[C]. Washington, DC: IEEE Computer Society, 2004. 35-44.
- [6] A Wang, A Chandrakasan. Energy-efficient DSPs for wireless sensor networks[J]. IEEE Signal Processsing Mag, 2002, 19(4):68-78.
- [7] Y Tian, J Boangoat, E Ekici, F O'zgu'ner. Real-time task mapping and scheduling for collaborative in-network[A]. Proceedings of the 20th International Parallel and Distributed Processing Symposium[C]. Rhodes Island: IEEE Computer Press, 2006. 1-10.
- [8] Y Tian, E Ekici, F O'zgu'ner. Energy-constrained task mapping and scheduling in wireless sensor networks[A]. Proceedings of the Workshop on Resource Provisioning and Management in Sensor Networks[C]. Washington, DC: IEEE Computer Society, 2005. 211-218.
- [9] Y Yu, V K Prasanna. Energy-balanced task allocation for collaborative processing in wireless sensor networks[J]. ACM/Kluwer J Mobile Networks Appl, 2005, 10(1/2): 115-131.
- [10] S Shible, R Castain, et al. Static mapping of subtasks in a heterogeneous ad hoc grid environment[A]. Proceedings of the 18th International Parallel and Distributed Processing Symposium[C]. Santa Fe: IEEE Computer Society, 2004. 110-123.
- [11] Yaoyao Gu. Real-time multimedia processing in video sensor networks[J]. Signal Processing: Image Communication, 2007, 22(3):237-251.
- [12] Howard Hassig, Randy Clark, et al. SCADA System Makes CSO Incidents a Thing of the Past[EB/OL]. [http://www.pennnet.com/display\\_article/308489/41/ARTCL/none/none/1/SCADA-System-Makes-CSO-Incidents-a-Thing-of-the-Past/](http://www.pennnet.com/display_article/308489/41/ARTCL/none/none/1/SCADA-System-Makes-CSO-Incidents-a-Thing-of-the-Past/), 2007-10-01.
- [13] T Rappaport. Wireless Communication Principles and Practice (2nd Edition)[M]. London: Prentice Hall PTR, 2002. 142-153.
- [14] J Kennedy, R C Eberhart. Particle swarm optimization[A]. Proceedings of the IEEE International Conference on Neural Networks IV[C]. Piscataway, NJ: IEEE Service Center, 1995. 1942-1948.
- [15] She D Y, Hsu C Y. A hybrid particle swarm optimization for job shop scheduling problem[J]. Computers & Industrial Engineering, 2006, 51(4): 791-808.
- [16] Zhang H, Li X, Li H. et al. Particle swarm optimization-based schemes for resource-constrained project scheduling[J]. Automation in Construction, 2005, 14(3): 393-404.
- [17] Deb K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2/4): 311-338.
- [18] Fisher H, Thompson G L. Probabilistic Learning Combinations of Local Job-shop Scheduling Rules[M]. Englewood Cliffs, NJ: Prentice-Hall, 1963. 225-251.
- [19] Lawrence S. Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques[D]. Pittsburgh: School of Industrial Administration, Carnegie Mellon University, 1984.

#### 作者简介:



易 军 男, 1973 年 10 月出生于重庆市, 现为重庆大学自动化学院博士研究生, 研究方向为无线传感器网络、嵌入式系统。

E-mail: laoyifrcq@163.com



石为人 男, 1948 年出生于重庆市, 现为重庆大学自动化学院教授, 博士生导师, 研究方向为智能感知、控制与决策。

E-mail: wrs@cqu.edu.cn

唐云建 男, 1982 年出生于重庆市, 现为重庆大学自动化学院博士研究生, 研究方向为无线传感器网络。

E-mail: yunjian.tang@qq.com

许 磊 男, 1982 年出生于重庆市, 现为重庆大学自动化学院博士研究生, 研究方向为无线传感器网络。

E-mail: xulei1981@yahoo.com.cn

