

无线传感器网络优化的任务管理算法研究

易本顺 陈杰 肖进胜
(武汉大学电子信息学院 武汉 430079)

摘要: 该文针对多跳分簇无线传感器网络多节点协同式并行处理应用,提出了一种新的基于改进粒子群优化算法的任务管理算法,该算法建立了基于复制的变异操作,并采用基于熵权的逼近理想解的排序法对算法结果进行客观评价与择优。详述了算法的各个要素,仿真结果证实,算法搜索效率高、可获得多目标优化的任务分配与调度解,且比文献中提出的其他算法的解性能优越。

关键词: 无线传感器网络; 任务管理; 多目标优化; 粒子群优化; 逼近理想解的排序法

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2010)11-2606-06

DOI: 10.3724/SP.J.1146.2009.01595

Research on Optimized Task Management Algorithms for Wireless Sensor Networks

Yi Ben-shun Chen Jie Xiao Jin-sheng

(School of Electronic Information, Wuhan University, Wuhan 430079, China)

Abstract: An algorithm based on an improved particle swarm optimization algorithm is proposed for the task management in the applications to multihop clustered wireless sensor networks which require collaborative processing executed in parallel on sensor nodes. Duplication-based mutation operation is set up and weighted entropy based technique for order preference by similarity to ideal solution is used to evaluate and choose objectively on the result of the algorithm. The components of the algorithm are explicitly presented and simulation results validate that it searches effectively and can obtain multi-objective optimized task allocation and scheduling solutions which outperform compared with that of other algorithms' in the literature.

Key words: Wireless Sensor Networks (WSNs); Task management; Multi-objective optimization; Particle Swarm Optimization (PSO); Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

1 引言

如今,许多新兴的无线传感器网络(WSNs)大数据量采集与传输应用因实时性要求而需网内处理。在这些应用中,进行本地信息处理、将数据量大幅降低的结果发送给汇聚节点比以多跳的方式直接发送原始数据更为节能。同时,较少的通信量能降低发送延迟,确保实时性。然而,在其中的图像、视频监控等应用中,处理算法(如视频压缩算法)能量与时间开销较大,将其分解为多个任务、多节点协同并行处理更能合理地利用网络能量资源,延长网络寿命且保证实时性^[1]。任务管理为多节点协同任务处理机制中的关键问题,包括任务分配与任务调度两方面的内容,研究如何将指定的任务集迁移到最合适的节点上去处理,以满足特定的应用服务质量

要求,由于延长网络寿命是WSNs应用设计必须考虑的问题,因此,WSNs任务管理的研究还包括在任务分配与调度的同时如何节省与均衡网络的能量资源,以最大化网络寿命。

到目前为止,只有较少的WSNs任务管理机制的研究成果报道,而互联多处理器平台下的任务管理机制已得到了深入与广泛的研究^[2-4],相关的研究成果由于基于有线互联、算法复杂度高、不考虑能耗指标等原因而不适用于WSNs。因WSNs的任务管理问题为NP完全问题,采用智能优化算法在任务分配与调度解空间进行寻优可在有限的时间复杂度内获得较优解^[1]。文献[5,6]提出了两种基于遗传算法(GA)的WSNs任务管理算法,其中的初始解构造及邻域移动(交叉与变异)方法只适用于无入口任务分配约束的场合^[7];文献[3]提出了一种基于复制的状态转移方法,能引导任务分配与调度解朝着较优的邻域转移,然而该方法允许某一任务可在多个处理器上处理,以增大能耗换取较短的调度长度。

2009-12-15 收到, 2010-05-24 改回

湖北省自然科学基金(2009CDB328)和武汉大学自主科研项目(6082003)资助课题

通信作者: 陈杰 Jay35781812@163.com

评价机制的优劣对算法输出解的性能有较大影响。采用加权和的方法建立优化目标的适值函数并依其大小判断解的优劣为常见的评价方法^[1,5,6], 判断结果较为主观, 虽然文献[2]提出了能量自适应窗口的概念, 文献[4]提出了基于能量梯度的判断准则, 但其判断依据仍不够充分。

本文提出了一种基于改进粒子群优化(PSO)算法的 WSNs 任务管理优化算法 OTM(Optimized Task Management), 其建立了基于复制的变异操作, 并应用基于熵权的逼近理想解的排序法(TOPSIS)多目标评价机制客观、充分地评价任务分配与调度解, 操作简单、搜索效率高, 算法设计遵循 WSNs 任务管理机制设计的各项要求, 最终能得到性能较优的任务分配与调度解。

2 问题描述

2.1 网络模型与应用模型

应用分簇算法, 将网络同构传感器节点划分为多个 k 跳的簇, 且假设各节点的位置信息可知, 同时, 网络采用时分复用或频道跳变机制, 以使各个簇的簇内通信互不干扰, 而利用若干特殊时隙或簇内信道的正交信道完成簇间信息的传递, 构建 WSNs 任务管理机制的网络模型。单簇处理在网络运行过程中汇聚节点所指派的应用请求, 当收到该请求后, 簇头便执行相应的任务分配与调度算法, 给簇内节点分配任务及安排其处理时刻, 节点获知调度结果后, 应用在该簇内便被开始处理。在应用处理过程中, 节点根据任务处理时刻合理地控制状态, 空闲时休眠、需处理通信或计算任务时唤醒, 假定网络节点间连通性好、数据传输不会丢包与出错且节点可同时进行数据通信与任务处理。

由一组计算任务构成的应用可用有向无环图(DAG)描述^[1]。一个 DAG 一般由多个入口任务(入口任务集为 V_c)及一个出口任务构成, 本文考虑入口任务数少于簇内可用节点数的情况, 而当存在多个出口任务时, 将其连接到一个零开销的虚拟出口任务。与文献[1]相同, 在应用 DAG 中计算任务 v_i 后加入通信任务 R_i , 对其进行扩展, 以明确地表现计算任务之间相关的通信事件及无线通信的广播特性。WSNs 簇中 DAG 有效的任务分配与调度解除需遵守 WSNs 本身的特性外, 还需遵守截止期约束、入口任务分配约束及通信关系约束^[7]。

2.2 问题定义

定义 $H^x = \{h_1^x, h_2^x, \dots, h_n^x\}$ 为 DAG T 在 WSNs 簇 G 上的一个任务分配与调度解, x 为解空间索引。元素 $h_i^x \in H^x$ 的形式为 $(v_i, m_k, s_{v_i, m_k}, f_{v_i, m_k}, t_{v_i, m_k}, c_{v_i, m_k})$,

表示将任务 v_i 分配于 m_k 处理, s_{v_i, m_k} , f_{v_i, m_k} , t_{v_i, m_k} 及 c_{v_i, m_k} 为相应的起始时刻、完成时刻、时间开销及能量开销。WSNs 任务管理机制设计的目的在于获得最优任务分配与调度解 $H^o \in \{H^x\}$, 在满足所有约束的前提下, 应用处理实时性高、网络能耗最小且均衡。处理实时性与网络能耗的模型如下:

$$\text{length}(H^x) = \max_{i,k} f_{v_i, m_k} \leq DL \quad (1)$$

$$\text{energy}(H^x) = \sum_{v_i \in V, k} c_{v_i, m_k} + \sum_{v_i \in R, k} c_{v_i, m_k} \quad (2)$$

其中 $\text{length}(H^x)$ 及 $\text{energy}(H^x)$ 分别为 H^x 的调度长度及能耗, R 为扩展 DAG 中的通信任务集, DL 为截止期, 其余约束的数学模型详见文献[7]。因此, WSNs 任务管理机制设计的目标即为寻找最优的任务分配与调度解 H^o , 在满足所有约束的前提下, 使 $\text{energy}(H^o)$ 及 $\text{length}(H^o)$ 二者权衡最小且网络能量均衡。文中使用的其余标记如下: ① $\text{pred}(v_i)$ 与 $\text{succ}(v_i)$ 表示 DAG 中 v_i 的直接前趋与后继任务; ② $m(v_i)$ 表示 v_i 所在节点; ③ $T(m_k)$ 表示分配给 m_k 的任务; ④ $\text{pred}_H(v_i)$ 与 $\text{succ}_H(v_i)$ 表示扩展 DAG 中 v_i 的直接前趋与后继任务。

节点的通信能耗与任务处理能耗的计算方法详见文献[8]。本文认为节点休眠时的能耗为 0, 忽略节点状态转换所引入的开销, 而当任务与其直接前驱任务分配于相同节点时, 两者通过本地存储器进行数据通信, 因此其通信代价为 0。

3 基于改进 PSO 算法的 WSNs 任务管理算法

传感器节点以无线多跳方式通信, WSNs 平台下的任务管理机制涉及 WSNs 的应用层、MAC 层及路由层。文献[1]在分析了 WSNs 无线通信的广播特性后, 将 DAG 进行扩展、多跳无线信道抽象为虚拟节点 C , 引入通信任务在 C 上“处理”时的罚函数来实现节点之间通信的冲突避免, 最后提出了多跳通信任务调度算法 CommTaskSchedule($R_i, m(R_i), m_r$), 依循地理位置路由算法所建路由确定通信任务 R_i 从 $m(R_i)$ 调度至其接收节点 m_r 的时刻、至中间各跳节点的时刻及总能耗, 算法设计符合 WSNs 节点的通信特点, 适用于任意情况下的 WSNs, 本文基于该多跳通信任务调度算法, 进行 WSNs 任务管理机制的优化设计。

由于 PSO 算法具有控制参数少、搜索速度快、鲁棒性好、简单易实现及并行处理特性等优点, 本文应用 PSO 算法实现 WSNs 任务管理的优化。基本 PSO 算法的本质是利用本身信息、个体极值信息

和全局极值信息指导粒子的下一步迭代位置。对于本文的任务管理问题，其当前位置便是一种任务分配与调度方式，如按照基本的 PSO 算法，其速度难以表达，这里采用基于 GA 思想的 PSO 算法^[9]，依此构建的 OTM 算法如表 1 所示，其中， n 为已迭代次数， P 为种群规模，一般随机给定一个恰当的值。下面详细介绍算法中的各个构成要素。

表 1 OTM 算法

1	设定粒子数 P ，规定迭代次数 N ，随机产生 P 个初始解 X_0 ；
2	设置粒子当前的位置为个体极值位置 $pxbest$ ，应用评价机制评价粒子的个体极值，找到全局极值位置 $gxbest$ ；
3	while $n < N$ do
4	for $j = 1 : P$ do
5	第 j 个粒子位置 $X_0(j)$ 与 $gxbest$ 交叉得到 $X_1'(j)$ ；
6	$X_1'(j)$ 与 $pxbest$ 交叉得到 $X_1(j)$ ；
7	对 $X_1(j)$ 进行变异操作；
8	应用评价机制评价 $X_1(j)$ 和 $pxbest(j)$ ，如前者优于后者，则 $pxbest(j) = X_1(j)$ ；
9	end for
10	$X_0 \leftarrow X_1$ ；
11	应用评价机制对粒子的个体极值进行评价，找到全局极值位置 $gxbest$ ；
12	end while
13	if $\exists pxbest: \text{length}(pxbest) \leq DL$ then
14	应用评价机制对有效的粒子的个体极值进行评价，输出最优值 $gxbest'$ ；
15	else
16	输出调度长度最小的 $pxbest$ ；
17	end if

3.1 初始解的构造

本文个体表达或粒子位置与问题解之间为直接映射关系，一个个体为一个问题解，即为一种任务分配与调度方式。初始解的构造方法如表 2 所示，先分配入口任务以满足入口任务分配约束，对于其他任务，根据其 b-level 值(计算方法见文献[3])的大小排序并逐个分配与调度。对于所选任务 v_i ，将其分配于随机选择的有效节点 m_k ，并将各个 $\text{pred}(v_i)$ 所产生的通信任务调度至 m_k ，调度结束后且当 m_k 处理器可用时便可处理 v_i ，其中， s_{v_i, m_k} 与 f_{v_i, m_k} 为

$$s_{v_i, m_k} = \max \left\{ f_{\text{mipred}(v_i), m_k} + ct_{R_{\text{mipred}(v_i)}}, f_{\text{spp}(v_i), m_k} \right\} \quad (3)$$

$$f_{v_i, m_k} = s_{v_i, m_k} + t_{v_i, m_k} \quad (4)$$

其中 $\text{mipred}(v_i)$ 为最晚完成与 v_i 通信的 $\text{pred}(v_i)$ ， $ct_{R_{\text{mipred}(v_i)}}$ 为 $\text{mipred}(v_i)$ 与 v_i 之间的通信时延， $\text{spp}(v_i)$ 为 m_k 上处理 v_i 前所处理的任

表 2 初始解的构造算法

1	for 各个 $v_i \in V_e$ do
2	将 v_i 分配于簇内任意有效节点 m_k ，且 m_k 上无其他入口任务；
3	将 $\text{succ}_H(v_i)$ 分配于 m_k ；
4	end for
5	以任务 b-level 值递减顺序排列各个计算任务，构成表 L ；
6	移除 L 中的入口任务；
7	while L 非空 do
8	选择 L 中的首元素 v_i ；
9	将 v_i 分配于簇内任意有效节点 m_k ；
10	if $\text{pred}_H(v_i) \notin T(m_k)$ then
11	for $R_h \in \text{pred}_H(v_i) - T(m_k)$ do
12	CommTaskSchedule($R_h, m(R_h), m_k$)；
13	end for
14	end if
15	获得 v_i 的 s_{v_i, m_k} 及 f_{v_i, m_k} ；
16	将 $\text{succ}_H(v_i)$ 分配于 m_k ；
17	将 v_i 从 L 中移除；
18	end while

3.2 邻域移动方法

3.2.1 交叉操作 交叉操作将两个粒子位置进行单点交叉，即随机地选择一个在两个粒子位置中处于不同节点的任务，将每个粒子位置中该任务所在的节点替换为另一个粒子位置中该任务所在节点，如果所选任务为入口任务，还需将该任务所在的新的节点上原入口任务转移至该任务原节点，且单点交叉可执行多次。操作结束后需对操作结果中任务的起止处理时刻重新调度，调度算法与初始解的构造算法基本相同，只是此时任务所在节点已经确定。

3.2.2 变异操作 为使变异操作的结果能较好地满足各项约束，本文提出基于复制的变异操作，其主要思想为将随机产生的解的任务复制(或插入)到 $X_1(j)$ 的对应节点中，生成中间状态 $MX_1(j)$ ，根据任务的优先级遍历所有任务，在中间状态两相同的任务中保留通信任务能耗最小的一个、移除另一个，该过程严格遵守所有约束，随机解的生成方法与初始解相同。显然，在网络连通性较好假设下，移除任务的依据能同时保证能耗、通信跳数与时延较小。将 $MX_1(j)$ 中的每对 v_i 称为 v_i 的两个复制，分别用 $\overline{r_{i,a}}(1)$ 与 $\overline{r_{i,b}}(2)$ 表示，其中一个复制称为另一复制的共轭复制，在复制名称上加一横线表示，即 $\overline{r_{i,a}}(1) = r_{i,b}(2)$ 、 $\overline{r_{i,b}}(2) = r_{i,a}(1)$ ，每个复制定义一个有效变量，以指示其为有效(为 true)或无效(为 false)。基于复制的变异操作算法如表 3 所示。该过程首先确保任务复制只分配于某节点一次及只有一个出口任务的复制存在于 $MX_1(j)$ 中。对于所审查的任务，首先判断 $MX_1(j)$ 中是否存在该任务的两个复

制且皆为有效, 如存在则从中选择一个设为无效, 选择的依据为复制通信任务的能耗, 值较大的复制为无效、较小的有效(步骤 9-13); 如不存在则移除无效复制, 如果该无效复制为入口任务的复制, 则需做适当调整以满足入口任务分配约束(步骤 14-21), 对于有效复制, 判定其每个直接前驱任务的复制中的哪个有效, 判定的依据为直接前驱任务复制与该有效复制的通信能耗, 值较小的为有效(步骤 23-27)。算法中复制之间的通信任务能耗可通过执行 CommTaskSchedule 函数获得, 算法结束后仍需对操作结果中任务的起止处理时刻进行调整。

表 3 基于复制的变异操作算法

```

1  产生一个随机解, 并将其中任务对应复制或插入到  $X_1(j)$ 
   中, 得到中间状态  $MX_1(j)$ ;
2  设置  $MX_1(j)$  中每个复制的有效域为 false;
3  根据任务间的优先关系, 移除单个节点上存在同一任务的
   两个复制中的一个, 另一个有效域设为 true;
4  移除  $MX_1(j)$  中出口任务完成时间最大的复制, 另一个有
   效域设为 true;
5  以任务 b-level 值递增顺序排列各个任务, 构成表  $S$ ;
6  while  $S$  非空 do
7      选择  $S$  中的首元素  $v_i$ ;
8       $IP_i = \{\text{pred}(v_i)\}$ ;
9      if  $\text{useful}(r_{i,a}(1)) == \text{true}$  且  $\text{useful}(r_{i,b}(2)) == \text{true}$ 
   then
10          $\text{uip} = \min\left(\sum_k c_{R_i, m_k}\right)$ ;
11          $\text{useful}(\overline{\text{uip}}) = \text{false}$ ;
12     end if
13     for  $v_i$  的每个复制  $r_i(h)$  do
14         if  $r_i(h)$  不为出口任务且  $\text{useful}(r_i(h)) \neq \text{true}$ 
   then
15             移除  $r_i(h)$ ;
16             if  $r_i(h)$  为入口任务 then
17                 while  $T(m(\overline{r_i(h)}))$  还存在其他入口任务的
   复制  $r_j(l)$  do
18                     移除  $r_j(l)$ ;
19                      $r_i(h) = r_j(l)$ ;
20                 end while
21             end if
22         else
23             for 各个  $ip_i \in IP_i$  do
24                  $RIP_i$  为  $ip_i$  的复制集;
25                  $\text{uip} = \min_{RIP_i, pm \in RIP_i} \left(\sum_k c_{R_{ip_i, pm}, r_i^{(h)}}\right)$ ;
26                  $\text{useful}(\text{uip}) = \text{true}$ ;
27             end for
28         end if
29     end for
30     将  $v_i$  从  $S$  中移除;
31 end while

```

3.3 评价机制

对于算法中的多个评价对象多指标决策问题, 本文采用易于实现、评价合理的基于熵权的 TOPSIS 评价机制^[10], 以得到较为客观的评价结果。具体步骤描述如下:

(1) 建立决策矩阵 设优化指标为 f_1 及 f_2 (对应于调度长度及能耗), 如有 m 个评价对象, 则决策矩阵为

$$F = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ \vdots & \vdots \\ f_{m1} & f_{m2} \end{bmatrix} \quad (5)$$

其中 f_{ij} 为第 i 个评价对象指标 f_j ($i = 1, 2, \dots, m$, $j = 1, 2$) 的值。

(2) 计算规范化矩阵 各指标通常具有不同的量纲, 需要对指标值矩阵规范化, 得到规范化矩阵 $B = [r_{ij}]_{m \times 2}$, 其中

$$r_{ij} = f_{ij} / \sqrt{\sum_{i=1}^m f_{ij}^2} \quad (6)$$

(3) 确定评价指标的输出信息熵 在信息论中, 信息熵是系统无序程度的度量。输出信息熵的计算公式为

$$E_j = -\frac{1}{\ln m} \sum_{i=1}^m r_{ij} \ln r_{ij} \quad (7)$$

且当 $r_{ij} = 0$ 时, 规定 $r_{ij} \ln r_{ij} = 0$ 。

(4) 计算指标的权重 利用信息熵来计算各指标的权重:

$$w_j = \frac{1 - E_j}{\sum_{k=1}^2 (1 - E_k)} \quad (8)$$

(5) 构造加权规范化矩阵 为了反映各因素的重要程度不同, 将规范化的数据加权, 构成加权规范化矩阵:

$$Z = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} \\ w_1 r_{21} & w_2 r_{22} \\ \vdots & \vdots \\ w_1 r_{m1} & w_2 r_{m2} \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ \vdots & \vdots \\ z_{m1} & z_{m2} \end{bmatrix} \quad (9)$$

(6) 确定正理想解和负理想解 由于指标都是成本型, 则正理想解和负理想解分别为

$$Z^+ = \{Z_1^+, Z_2^+\}, \quad Z_j^+ = \min z_{ij} \quad (10)$$

$$Z^- = \{Z_1^-, Z_2^-\}, \quad Z_j^- = \max z_{ij} \quad (11)$$

(7) 计算各评价对象与正理想解和负理想解之间的距离

$$d_i^+ = \sqrt{\sum_{j=1}^2 (z_{ij} - Z_j^+)^2} \quad (12)$$

$$d_i^- = \sqrt{\sum_{j=1}^2 (z_{ij} - Z_j^-)^2} \quad (13)$$

(8) 确定相对接近度 评价对象与理想解的相对接近度为

$$D_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (14)$$

根据 D_i 的大小, 将评价对象的相对接近度进行排序, 值越大, 相应的评价对象性能越优。

3.4 计算复杂度分析

对于 k 跳簇网络以及由 n 个任务与 e 条有向边构成的 DAG, 由于多跳通信调度算法的复杂度为 $O(k)$, 表 2 中初始解构造算法的步骤 11-步骤 12 的执行时间为 $O(e/n)$, 步骤 7 处循环的执行时间为 $O(n)$, 因此, 初始解的构造算法的计算复杂度为 $O(ek)$ 。在表 3 的变异操作算法中, 步骤 13 处循环的执行时间为 $O(n)$, 步骤 6 处循环的执行时间也为 $O(n)$, 则该算法的计算复杂度为 $O(n^2)$ 。因变异操作结束后还对任务的起止处理时刻进行了调整, 变异操作算法的实际计算复杂度则为 $O(n^2 + ek)$, 于是可得 OTM 算法的计算复杂度为 $O(NP(n^2 + ek))$ 。

4 仿真实验及分析

通过仿真实验, 对本文提出的 OTM 性能进行分析。实验时, 设置无线信道带宽为 1 Mb/s, 发送范围为 10 m, 传感器节点处理器的处理频率为 206 MHz, 迭代过程起始产生 11 个初始解。随意生成 250 对 DAG 与 3 跳簇网络, DAG 的任务数、入口任务数及直接前趋任务最大数分别为 40, 10, 10, 任务计算量及通信数据量均匀地分布于 $[300\text{kilo-clock-cycle} \pm 10\%]$ 及 $[800\text{ bit} \pm 10\%]$ 内, 簇网络的构建方法详见文献[1], 如图 1 与图 2 所示, 首先比较 OTM 与文献[1]所提出的 MTMS* 的调度长度及一次应用处理能耗, 图中算法各个指标的实验结果为对应 250 个值的平均值。

由图 1 和图 2 可知, OTM 比 MTMS* 性能更优。MTMS* 建立了任务分配的适值函数, 采用加权和的方法计算任务分配的代价增益, 以达到对任务分配进行评价及最小化应用处理代价的目的, 获得了较

优的任务分配与调度解, 然而, 算法主观地设置适值函数权重, 虽以 0.1 的步长增大权重的方式考虑了 11 种权重取值时的情况, 但只是得到了解空间中 11 个较优解而已, 还存在更优解, 算法的寻优能力非常有限; 而 OTM 采用 PSO 算法进行迭代搜索, PSO 算法较好的全局搜索能力与并行处理特性能在较短的时间内搜索出较优解, 同时, 其采用 TOPSIS 评价机制, 使多指标解的评价与择优更为客观合理, 对之后的搜索也能起到很好的引导作用。从图 1 和图 2 中还可以看出, 算法的调度长度与能耗之间呈相反的变化趋势且都会趋于一定值, 当截止期较小时, 两算法所获得的 11 个最终任务分配与调度解可能都为无效解, 选择调度长度最小的解输出, 其能耗一般较大; 随着截止期的增大, 有效解的数量增多, 选择调度长度较大的解输出的概率增大, 于是调度长度增大、能耗减小; 当截止期较大时, 所有的解都为有效解, 继续增大截止期将不会改变最终的输出解, 曲线趋于平稳, 而 OTM 的解的最大调度长度比 MTMS* 的小, 因此, 其曲线更早地趋于平稳。

再分析 OTM 的能耗均衡性。随意生成 250 对 DAG 与单跳分簇网络, DAG 的任务数、入口任务数及直接前趋任务最大数分别为 20, 5, 5, 其余参数不变, 图 3 所示为 OTM 与 MTMS* 的最大节点能耗(MECpN)指标的性能比较, MECpN 可度量网络能耗均衡性^[7], 图中实验结果为对应 250 个值的平均值。由图可知, OTM 与 MTMS* 相比, MECpN 值更小、能耗均衡性更优、网络寿命更长, 且都随着截止期的增大而减小, 其曲线也会趋于平稳。

5 结束语

WSNs 多节点协同式并行处理机制的研究是近期 WSNs 研究领域的一个热点与难点, 任务管理机制为其中的关键问题, 其内容主要包括任务分配与调度。在研究了目前存在的一些 WSNs 任务分配与调度算法后, 引入了一种基于改进 PSO 的 WSNs 任务管理优化算法。仿真结果表明该算法能获得在多个指标上性能较优的任务分配与调度解, 从而实现了 WSNs 优化的任务管理的目的。

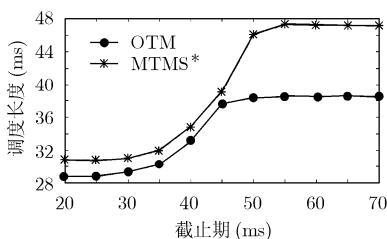


图 1 调度长度性能比较

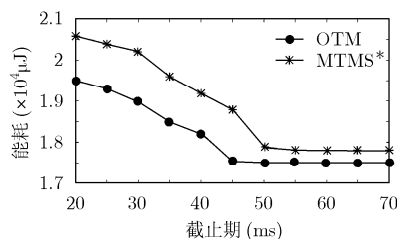


图 2 一次应用处理能耗性能比较

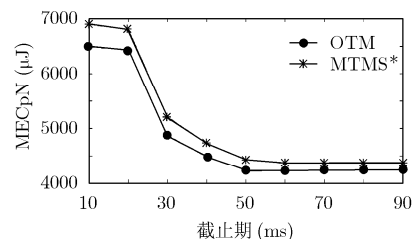


图 3 MECpN 的性能比较

参 考 文 献

- [1] Yuan T and Ekici E. Cross-Layer collaborative in-network processing in multihop wireless sensor networks [J]. *IEEE Transactions on Mobile Computing*, 2007, 6(3): 297-310.
- [2] Tao X and Xiao Q. An energy-delay tunable task allocation strategy for collaborative applications in networked embedded systems [J]. *IEEE Transactions on Computers*, 2008, 57(3): 329-343.
- [3] Lee Y C and Zomaya A Y. A novel state transition method for metaheuristic-based scheduling in heterogeneous computing systems [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2008, 19(9): 1215-1223.
- [4] Lee K G, Bharadwaj V, and Sivakumar V. Design of fast and efficient energy-aware gradient-based scheduling algorithms for heterogeneous embedded multiprocessor systems [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2009, 20(1): 1-12.
- [5] 唐恒娟, 李志刚. 传感器网络 GaSA 任务分配算法[J]. *计算机工程与应用*, 2009, 45(18): 126-128.
Tang Heng-juan and Li Zhi-gang. GaSA task allocation algorithm of sensor networks [J]. *Computer Engineering and Applications*, 2009, 45(18): 126-128.
- [6] 朱敬华, 高宏. 无线传感器网络中能源高效的任务分配算法 [J]. *软件学报*, 2007, 18(5): 1198-1207.
Zhu Jing-hua and Gao Hong. An energy efficient algorithm for task allocation in wireless sensor networks [J]. *Journal of Software*, 2007, 18(5): 1198-1207.
- [7] Yang Y and Prasanna V K. Energy-balanced task allocation for collaborative processing in wireless sensor networks [J]. *Mobile Networks and Applications*, 2005, 10(1): 115-131.
- [8] Wang A and Chandrakasan A. Energy-efficient DSPs for wireless sensor networks [J]. *IEEE Signal Processing Magazine*, 2002, 19(4): 68-78.
- [9] 高尚, 杨静宇. 武器-目标分配问题的粒子群优化算法[J]. *系统工程与电子技术*, 2005, 27(7): 1250-1252.
Gao Shang and Yang Jing-yu. Solving weapon-target assignment problem by particle swarm optimization algorithm [J]. *System Engineering and Electronics*, 2005, 27(7): 1250-1252.
- [10] 赵红梅, 李科伟. 基于改进的 TOPSIS 法在供应链中经销商的选择与评价方法研究[J]. *内蒙古农业大学学报*, 2007, 28(2): 180-185.
Zhao Hong-mei and Li Ke-wei. Research of choice and assessment method of distributor in supply chain based on TOPSIS [J]. *Journal of Inner Mongolia Agricultural University*, 2007, 28(2): 180-185.
- 易本顺: 男, 1965 年生, 教授, 博士生导师, 研究方向为光纤通信、无线通信、图像处理.
- 陈 杰: 男, 1982 年生, 博士生, 研究方向为无线传感器网络及其并行处理.
- 肖进胜: 男, 1975 年生, 博士, 副教授, 研究方向为多媒体信息处理.