

文章编号:1000-582X(2010)06-096-05

## 嵌入式实时系统周期任务能耗感知调度

罗 钧,刘永锋,吴 志

(重庆大学 光电技术及系统教育部重点实验室,重庆 400044)

**摘 要:**在电池供电的嵌入式实时系统中,针对实时周期任务集在一个可变电电压处理器上的能耗感知调度问题,在改进的人工鱼群算法(improved artificial fish school algorithm, IAFSA)基础上,提出了一种离线的能耗感知调度算法,该算法通过构造人工鱼的行为约束自然地满足时限和能耗约束,最大化系统总奖赏值。与贪婪算法相比,该算法提高系统的总奖赏值约 47%。且随着电池能量的减少,系统能耗利用效率逐渐增大,说明该算法具有良好的能耗感知特性。

**关键词:**能耗感知调度;实时系统;任务调度;人工鱼群算法

**中图分类号:** TP316.7

**文献标志码:** A

## Energy-aware schedule of periodic tasks in embedded real-time systems

LUO Jun, LIU Yong-feng, WU Zhi

(Key Laboratory for Opto-electronic Technology & System of the Ministry of Education, Chongqing University, Chongqing 400044, P. R. China)

**Abstract:** For the energy-aware schedule of battery-powered real-time embedded system, a novel offline energy-aware schedule algorithm is proposed based on the improved artificial fish school algorithm (IAFSA). The algorithm constructs the behavior constraint of the artificial fish to make the random search meet the energy constraint and deadline while maximizing the system reward. Experimental results show that the proposed algorithm improves the system reward about 47% compared with the Greedy algorithm. Besides, the energy efficiency increases while the energy budget decreases, which demonstrates the good energy-awareness efficiency of the proposed algorithm.

**Key words:** energy-aware schedule; real-time system; task schedule; artificial fish school algorithm

在电池供电的嵌入式实时系统中,能耗问题越来越突出。动态电压调节(dynamical voltage scaling, DVS)技术是一种有效的节能技术,广泛应用于能耗管理中。它通过降低处理器的频率和电压来降低系统功耗<sup>[1-2]</sup>。但降低频率的同时也会导致任务执行时间延长。如今大量的研究集中于如何在保证系统实时性的同时尽量降低系统能耗<sup>[3-7]</sup>,而如何更有效地利用有限的能量,提高系统能耗利用率

的研究才刚刚起步。能耗感知就是让系统能够根据当前能耗的提供情况调整自身行为,适应能耗约束变化,提高能耗的利用率<sup>[8]</sup>。因此,一个有效的能耗感知调度算法应能在提高系统能耗利用率的同时兼顾系统能量约束和任务时限约束。

近年来,研究人员尝试多种不同的方法提高系统的能耗感知性能<sup>[9-12]</sup>。文献[9]在贪婪算法的基础上混合了多项式时间逼近和最早截止时间优先算

收稿日期:2010-01-19

基金项目:国防科工委国防军工计量“十一五”计划重点项目(J102006B104)

作者简介:罗钧(1963-),男,重庆大学教授,主要从事智能仪器系统、信息获取及处理的研究与开发,  
(E-mail)luojun@cqu.edu.cn。

法来优化满足能耗约束下的周期实时任务的最大奖赏值调度。文献[10]提出了针对基于帧的任务集在固定的能耗预算下使系统性能最大的两种算法,并在文献[11]中又将算法扩展到了周期性任务集的情况。文献[12]针对周期性任务集在变电压处理器上的非占先式调度问题,利用遗传算法在有限的能耗预算下使系统的性能得到最优化。由于电池供电实时系统中的能耗感知调度问题是带约束条件的 NP 组合优化问题,贪婪算法等传统方法很难有效计算 NP 问题,而遗传算法结构复杂,计算量大,收敛速度慢。

鱼群算法是一种群智能多点并行随机搜索优化算法,具有克服局部极值,获得全局极值的能力,并且对初值和参数选择不敏感、鲁棒性强、收敛速度快、简单、易实现,适合具有实时性要求的场合<sup>[13]</sup>。笔者研究了人工鱼群算法在电池供电实时系统中的应用,并给出了仿真实例。

## 1 系统模型和问题定义

### 1.1 系统模型

假设一个由  $N$  个周期实时任务组成的任务集为  $T = \{T_1, T_2, \dots, T_N\}$ , 任务间相互独立, 在 0 时刻均处于就绪状态。任务  $T_i$  的周期为  $p_i$ , 其截止期  $d_i$  与  $p_i$  相等。执行任务  $T_i$  所需的最坏情况下时钟周期数为  $C_i$ , 所有任务均在一个可变电压处理器上执行, 该处理器能在运行过程中动态地改变处理器的频率(忽略频率调整开销), 这里假设处理器具有  $M$  个离散的工作频率, 即  $\{f_1, f_2, \dots, f_M\}$ 。  $T_i$  在处理器速度  $f_k$  下执行所需时间记为  $t_{i,k}$ , 所需能耗记为  $e_{i,k}$ , 所得奖赏值记为  $v_{i,k}$ 。所有任务周期的最小公倍数即任务集的超周期记为  $T_{LCM}$ , 对于周期性任务集, 每个超周期内的情况是相同的, 所以只需在  $[0, T_{LCM}]$  范围内进行研究。超周期内总能耗约束记为  $E_{budget}$ , 每个任务在超周期内最多执行一次, 不一定保证所有的任务都满足截止期, 在系统的能耗预算不足以保证所有的实例都满足截止期时, 以稳定的方式跳过一些不关键的任务<sup>[14]</sup>。

### 1.2 问题描述

系统的优化目标是: 为任务集中的每个任务指派处理器执行速度  $f_k$ , 使系统总奖赏值最大, 即

$$\max \sum_{i=1}^N \sum_{k \in [0, M]} v_{i,k}, \quad (1)$$

并且所有任务执行的总能耗满足系统能耗约束

$$s. t. \sum_{i=1}^N \sum_{k \in [0, M]} e_{i,k} \leq E_{budget}, \quad (2)$$

同时满足任务时限约束, 即所有被选中的任务执行时间不得超过其截止期

$$s. t. d_i - t_{i,k} \geq 0, \quad (3)$$

其中,  $k$  代表任务执行的处理器速度编号。当  $k=0$  时, 表明该任务被放弃, 因此  $v_{i,k}=0, e_{i,k}=0, t_{i,k}=0$ 。当  $k \neq 0$  时, 该任务在处理器速度  $f_k$  下执行。下面采用鱼群算法机制来求解问题的全局最优解。

## 2 基于 IAFSA 的能耗感知调度

### 2.1 基本 AFSA 原理

AFSA 最早是由李晓磊等人在 2002 年提出的一种新型的寻优算法, 该算法模拟鱼群的游弋觅食行为, 通过鱼之间的集体协作使群体达到最优化的目的<sup>[15]</sup>。算法主要有以下 4 种行为

1) 觅食行为: 这是鱼的基本行为, 当发现附近有食物时, 向该方向游去。

2) 追尾行为: 当某条鱼发现该处食物丰富时, 其他鱼会快速尾随而至。

3) 聚群行为: 每条鱼在游动过程中尽量向临近伙伴的中心移动并避免过分拥挤。

4) 随机行为: 当闲暇无事时, 自由游动。

算法采用自下而上的设计方法, 即首先构造人工鱼的个体模型, 个体在寻优的过程中自适应地选择合适的行为, 最后全局最优结果通过群体或者个体表现出来。

### 2.2 符号定义

AFSA 最初应用于连续函数优化问题中, 而能耗感知调度问题模型是带约束条件的离散函数优化问题, 因此, 需要对原始定义进行相应的改进, 以满足应用需求。

1)  $X_i = (x_1, x_2, \dots, x_N)$  为人工鱼的当前状态, 表示待调度任务集中  $N$  个任务的执行频率等级, 人工鱼总数为  $n_i$ 。

2)  $Y_i = f(X_i)$  为目标函数值, 表示系统的总奖赏值。

3)  $d(X_i, X_j)$  为人工鱼个体  $X_i, X_j$  之间的距离, 即  $X_i, X_j$  表示的 2 种调度方案中对应位置任务执行频率不同的个数, 令

$$X_i = \{a_1, a_2, \dots, a_N\},$$

$$X_j = \{b_1, b_2, \dots, b_N\},$$

其计算公式为

$$d(X_i, X_j) = \sum_{m=1}^N \text{sign}(|a_m - b_m|), \quad (4)$$

其中,

$$\text{sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases} \quad (5)$$

4)  $V$  即 (visual) 和  $\delta$  分别表示人工鱼的视野范围和拥挤度因子;  $\text{step}$  为人工鱼移动的步长。

5)  $N(X_i, V)$  为人工鱼  $X_i$  的视野邻域,  $X_j$  为  $X_i$  的伙伴鱼。

$$N(X_i, V) = \{X_j \mid d(X_i, X_j) \leq V\}, \quad (6)$$

6)  $X_{\text{Center}}$  为人工鱼  $X_i$  视野邻域内伙伴鱼的中心:

$$X_{\text{Center}} = \bigcup_{j=1, j \neq i}^{n_f} (X_i \cap X_j), \quad (7)$$

其中,  $n_f$  为人工鱼  $X_i$  的伙伴鱼个数。

### 2.3 行为描述

AFSA 先初始化一群人工鱼, 然后通过迭代搜寻最优解。在每次迭代中, 人工鱼主要是通过觅食、聚群和追尾等行为来更新自己, 从而实现寻优, 具体行为描述如下<sup>[14]</sup>:

#### 2.3.1 觅食行为

当前人工鱼  $X_i$  据式(6)在其视野邻域内随机选择一个状态  $X_j$ , 再由式(1)计算  $Y_i$  与  $Y_j$ 。若  $Y_j > Y_i$ , 则用  $X_j$  代替  $X_i$  成为新的当前状态; 否则,  $X_i$  继续随机选择  $X_j$ , 判断是否满足前进条件, 反复尝试  $\text{trynumber}$  次之后, 仍不满足, 则执行随机移动行为, 使  $X_i$  到达一个新的状态。

#### 2.3.2 聚群行为

当前人工鱼  $X_i$  据式(6)搜索视野邻域内的伙伴鱼, 并由式(7)计算鱼群中心  $X_{\text{Center}}$ , 若  $Y_{\text{center}} > Y_i$  且  $n_f/n_i < \delta$ , 则  $X_{\text{Center}}$  替代  $X_i$  成为当前状态, 否则执行其他行为。

#### 2.3.3 追尾行为

当前人工鱼  $X_i$  据式(6)搜索视野邻域内的所有伙伴中最优者  $X_{\text{max}}$ , 若  $Y_{\text{max}} > Y_i$ , 并且  $X_{\text{max}}$  的视野邻域内的伙伴鱼数目  $n_f$  满足  $n_f/N < \delta$ , 表明  $X_{\text{max}}$  附近有更多的食物且不太拥挤, 则  $X_i$  向  $X_{\text{max}}$  移动一步, 否则执行其他行为。

#### 2.3.4 约束行为

每条人工鱼在追求系统总奖赏值最大的同时必须满足能耗和时限约束, 为了避免迭代过程中产生不可行解, 这里定义了一个 Correct 算子, 每次人工鱼状态发生改变时该算子立即检查该人工鱼是否满

足约束条件, 若不满足则将其修正为一个与当前状态最相近的可行解。

#### 2.3.5 公告板

公告板记录整个鱼群的历史最优状态, 各人工鱼每执行完一次行动后就对比自身状态和公告板状态, 若自身状态优于公告板状态则更新公告板。

#### 2.3.6 变异机制

虽然基本 AFSA 能在一定程度上避免陷入局部最优, 但当一部分人工鱼在非全局极值点聚集较严重时, 算法的收敛速度下降, 甚至出现早熟现象。为了克服该缺点, 文中引入了类似遗传算法的变异算子, 在算法迭代过程中记录公告板连续无变化的次数, 若超过  $\text{Maxbetter}$  次则采取变异操作, 即保存当前最优人工鱼群的状态, 并对其他所有人工鱼执行变异操作, 即对人工鱼的所有维生成一个随机数  $r \in [0, 1]$ , 若  $r \leq p$  ( $p$  为变异概率) 则对该维进行随机赋值, 否则保持该维不变。该变异机制不仅能保持种群的多样性, 使人工鱼跳出局部极值点的束缚, 收敛于全局极值点, 同时也提高了算法的收敛速度。

#### 2.3.7 行为选择

上述行为没有明确的先后执行顺序, 文中人工鱼依循进步即可的原则, 即先执行追尾行为, 若没有明显进步则执行聚群行为, 若依旧没有明显进步则执行觅食行为。整个算法不需要高层的指挥者, 也不需要关于命题的先验知识的启发, 每条人工鱼按照前述的行为规则游动, 问题的满意解最终在公告板上显示出来。

#### 2.3.8 收敛准则

文中定义了如下收敛准则: 迭代过程中, 公告板前后状态是否足够逼近, 即

$$|Y_{t+1} - Y_t| < \xi, \quad (8)$$

其中,  $\xi$  为一个足够小的正数,  $t$  为迭代次数。若公告板前后差值连续  $\text{Limit}$  次小于  $\xi$  (注:  $\text{Limit} > \text{Maxbetter}$ ), 则算法终止, 输出最优解。

### 2.4 贪婪算法求解

贪婪算法 (Greedy) 是指从问题的初始状态出发。通过若干次的贪心选择而得出最优值 (或较优解) 的一种解题方法。

定义任务  $T_i$  的单位能量奖赏值为  $v_{i,k}/e_{i,k}$ 。贪婪算法求解规则如下: 优先考虑单位能量奖赏值最大的任务, 并按顺序选择使奖赏值最大的速度执行; 判断此时是否满足约束条件, 若不满足则选择奖赏值次之的速度执行, 依次类推, 直到找到满足约束且奖赏值尽可能大的速度; 如果找不到, 该任务被放弃执

行。Greedy 算法作为鱼群算法的参考。

### 3 仿真结果分析

实验模拟 Intel Xscale 结构的处理器以验证算法的性能,该处理器有 5 个离散的频率,速度与电压的对应关系如表 1 所示。

表 1 Intel Xscale 处理器频率电压表

速度编号	频率/MHz	电压/V
1	150	0.75
2	400	1.0
3	600	1.3
4	800	1.6
5	1 000	1.8

设每个任务的周期在 $[1,100]$ 之间随机产生,每个任务在最坏情况下执行所需处理器时钟周期数 $C_i$ 在 $[150,15\ 000]$ 之间随机产生,任务 $T_i$ 在速度等级 $k$ 下的执行时间 $t_{i,k} = C_i/f_k$ ,其执行时间与速度成反比。任务 $T_i$ 在速度等级为 $k$ 时的执行所需功耗 $P_{i,k} = \alpha C_L V_k^2 f_k$ (不考虑处理器的静态功耗),其中 $\alpha C_L$ 在 $[0.8,1.2]$ 之间随机产生,其能耗 $e_{i,k} = P_{i,k} \cdot t_{i,k}$ ,奖赏值 $v_{i,k}$ 在 $[1,100]$ 之间随机产生,任务执行时间越短奖赏值越高。定义系统最大能耗为 $\max E = \sum_{i=1}^N e_{i,M}$ 即所有任务在最大处理器速度下执行的能耗,系统能耗约束 $E_{\text{budget}} = \alpha \cdot \max E$ ,其中 $\alpha \in [0,1]$ ,为能耗约束比例因子。

#### 3.1 能耗感知特性分析

根据上述任务参数设置生成 3 个独立的任务集,如表 2 所示。

表 2 3 个独立的任务子集

任务集编号	任务个数	maxE(单位)
1	10	235 556
2	40	988 543
3	70	1 776 910

定义能耗利用率为系统总奖赏值与 $E_{\text{budget}}$ 的比值,当 $E_{\text{budget}}$ 一定时,总奖赏值越大能耗利用率越高。定义相对能耗利用率为当前能耗利用率与 $E_{\text{budget}} = \max E$ 时的能耗利用率之间的比值。实验中,利用 IAFSA 算法对 3 个任务子集进行测试,结果如图 1 所示,横坐标 $\alpha$ 表示能耗约束比例,纵坐标 REE 表示相对能耗利用率。

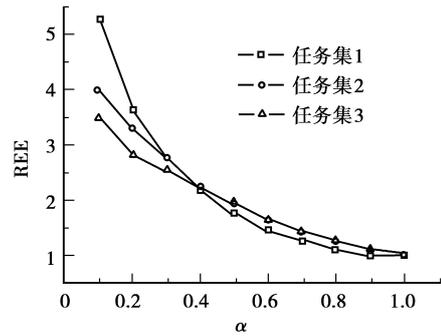


图 1 IAFSA 能耗感知特性曲线

从图 1 可以看出,能耗比例因子 $\alpha < 1$ 时,相对能耗利用率均大于 1,并且随着能耗约束比例的逐渐减小,相对能耗利用率快速增大。说明 IAFSA 算法具有较强的能耗感知能力,系统能根据外部能耗约束的变化自动调整任务调度行为,在满足能耗约束和任务时限约束的同时优化调度结果。并且对不同的任务规模所得到的结果均比较稳定。

#### 3.2 系统总奖赏值对比分析

根据前述任务参数设置生成 100 个任务集,每个任务集由 90 个任务组成。分别利用 IAFSA 算法和 Greedy 算法对其进行测试,并对比系统总奖赏值。图 2 中横坐标 $N$ 表示任务规模,纵坐标 $V$ 表示系统总奖赏值,图中每个点均为 100 次实验的平均值。

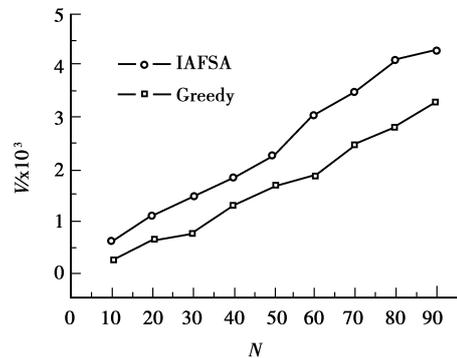


图 2 IAFSA 与 Greedy 算法所得总奖赏值对比

从图 2 可以看出,相对于 Greedy 算法来说,IAFSA 算法能提高系统平均总奖赏值约 47%,且随着任务规模的增加 IAFSA 算法性能优势越明显。因此,IAFSA 算法更能保证有价值的任务执行。

### 4 结 论

利用 IAFSA 对待调度任务集中的任务进行选择,分配执行频率,在同时满足系统能耗约束和任务

时限约束的情况下,最优化系统的能耗感知性能。与 Greedy 算法相比,IAFSA 所得的系统平均奖赏值比 Greedy 算法提高了 47%左右,并且 IAFSA 能根据外部能耗约束的变化自动调整任务调度行为,系统能耗约束越强,系统能耗利用率越高。因此,IAFSA 能有效地解决能耗感知调度问题。在此基础上如何实现在线的能耗感知调度算法,进一步提高算法性能是今后研究的方向。

#### 参考文献:

- [ 1 ] AYDIN H, MELHEM R, MOSSE D, et al. Dynamic and aggressive scheduling techniques for power-aware real-time systems [ C ] // Proceedings-Real-Time Systems Symposium, December 2- 6, 2001. London, United Kingdom. London, United Kingdom, [s. n.]. 2001; 95-105.
- [ 2 ] VASIC M, GARCIA O, OLIVER J A, et al. A DVS system based on the trade-off between energy savings and execution time [ C ] // 11th IEEE Workshop on Control and Modeling for Power Electronics, COMPEL 2008, 2008, Zurich, Switzerland. [S. l.]: IEEE, 2008; 398-406.
- [ 3 ] YAO F, DEMERS A, SHENKER S. Scheduling model for reduced CPU energy [ C ] // Annual Symposium on Foundations of Computer Science - Proceedings, Oct. 23-25, 1995, Milwaukee, WI, USA. Milwaukee, WI, USA: IEEE, 1995; 374-382.
- [ 4 ] HAN J, LI Q, ESSA A A. Dynamic power-aware scheduling algorithms for real-time task sets in parallel and distributed computing environments [ J ]. Chinese Journal of Electronics, 2006, 15(1): 41-46.
- [ 5 ] XU R, MOSSE D, MELHEM R. Minimizing expected energy consumption in real-time systems through dynamic voltage scaling [ J ]. ACM Transactions on Computer Systems, 2007, 25(4).
- [ 6 ] ZHUO J, CHAKRABARTI C. System-level energy-efficient dynamic task scheduling [ C ] // Proceedings-Design Automation Conference, June 8-12, 2005, Anaheim, CA, United states. [S. l.]: IEEE, 2005; 628-631.
- [ 7 ] RONG P, PEDRAM M. Energy-aware task scheduling and dynamic voltage scaling in a real-time system [ J ]. Journal of Low Power Electronics, 2008, 4(1):1-10.
- [ 8 ] UNSAL O S, KOREN I. System-level power-aware design techniques in real-time systems [ J ]. Proceedings of the IEEE, 2003, 91(7):1055-1069.
- [ 9 ] CHEN J J, KUO T W. Voltage scaling scheduling for periodic real-time tasks in reward maximization [ C ] // Proceedings - the 26th IEEE International Real-Time Systems Symposium, December 5-8, 2005. Miami, Florida, USA. [S. l.]: IEEE, 2005; 345-355.
- [10] RUSU C, MELHEM R, MOSSE D. Maximizing the system value while satisfying time and energy constraints [ C ] // Proceedings - Real-Time Systems Symposium, December 2-5, Austin, TX, USA. [S. l.]: IEEE, 2002; 246-255.
- [11] RUSU P, MELHEM R, MOSSE D. Multi-version scheduling in rechargeable energy-aware real-time systems [ J ]. Journal of Embedded Computing 2005, 1(2):271-283.
- [12] 解玉凤, 魏少军. 实时周期任务的非占先式能耗感知调度 [ J ]. 计算机辅助设计与图形学学报, 2006, 18(2):246-248.
- XIE YU-FENG, WEI SHAO-JUN. Energy-aware scheduling of nonpreemptive periodic tasks in Real-Time systems [ J ]. Journal of Compute-Aided Design & Computer Graphics. 2006, 18(2): 246-248.
- [13] WANG D, ZHOU Y, CAO D. Artificial fish-school algorithm for solving nonlinear equations [ J ]. Journal of Information and Computational Science, 2007, 4(1): 281-289.
- [14] ALENAWY T A, AYDIN H. On energy-constrained real-time scheduling [ C ] // Proceedings - Euromicro Conference on Real-Time Systems, June30-July 2, 2004, Catania, Italy. [S. l.]: Institute of Electrical and Electronics Engineers Inc. , 2004, 16; 165-174.
- [15] 李晓磊. 一种新型的智能优化方法-人工鱼群算法 [ D ]. 杭州: 浙江大学, 2003.

(编辑 陈移峰)