一种独立任务的同型机调度快速算法*

李小平, 徐晓飞, 战德臣

(哈尔滨工业大学 计算机科学与工程系,黑龙江 哈尔滨 150001)
E-mail: Li_xiaoping@0451.com
http://www.hit.edu.cn

摘要: 如何将 n 个独立任务调度到 m 台同型机上加工,使总完成时间最短,是一个复杂问题.通过分析 Bound Fit 预备算法的性质,结合 MULTIFIT 和 Bound Fit 提出 QUICKFIT 算法;对相同机器数和任务数,QUICKFIT 能用比 MULTIFIT 和 Bound Fit 都少的迭代次数得到相同的总完成时间.实验结果表明,任务机器比越大,QUICKFIT 算法的性能就越优于 MULTIFIT 和 Bound Fit.绝大多数情况下,总完成时间等于 MULTIFIT 和 Bound Fit 中的最小者.该算法适用于大规模同型机调度.

关 键 词: 同型机调度;装箱;LPT 算法;MULTIFIT 算法;任务机器比
中图法分类号: TP316 文献标识码: A

独立任务的多机调度问题不仅存在于计算机并行计算、操作系统等计算机技术领域,而且还广泛存在于工农业生产、交通运输及服务性行业,最典型的当属生产型企业中同一工作中心上的多台机器的生产调度问题.这类问题可描述为 n 个独立任务在 m 台完全相同的机器(同型机)上加工,如何安排使得给定的目标函数最优(如总完成时间最短、成本最小或拖期惩罚最小、资源利用率最高等).这里的“机器”是指处理机、工厂里的各种机床、维修工人,即“服务者”;“任务”则指进程或作业、等待机床加工的零件、待修理的机器等,即“服务对象”.

总完成时间最短、总资源利用率最高等都是多机调度的目标,相关任务的多机调度问题中这两个目标没有直接联系,总完成时间最短,资源利用率不一定最高;反之亦然.对于独立任务的多机调度问题中这两个目标却有一定的关系,因为同一台机器上的任务之间相互独立,同一台机器的各任务间没有空闲时间,所以总完成时间越短,总资源利用率就越高.因此独立任务的多机调度问题常以总完成时间最短为目标函数.

独立任务的多机调度使总完成时间最短的问题是一个 NP 难题[1],不可能存在多项式时间复杂度的算法以找到全局最优解.但在实际的生产调度中即便是近似最优解也能缩短工期、降低成本或减少拖期惩罚,这也是企业追求的目标.对于这类问题按照初始时刻所有机器的是否都空闲分为:SMSP(simultaneous multiprocessor scheduling problem)和 NMSP(nonsimultaneous multiprocessor scheduling problem).解决 NMSP 的多项式复杂度的寻找近似最优解的算法[2~4]大多是改进解 SMSP 的算法;而目前较典型的解 SMSP 的算法有:LPT(longest processing time)算法[5]、MULTIFIT 算法[6]、Bound Fit 算法[7].

它们都是先将加工时间按非增的顺序排序,但在时间复杂度和性能方面各有长处:(1) LPT 算法是顺序地将任务安排在总加工时间最少的机器上,时间复杂度为 $O(n\log n+n\log m)$; (2) MULTIFIT 算法是一种基于 bin-packing 的 FFD(first fit decreasing)方法的折半迭代算法,并以总加工时间代替容量,求出最短总完成时间,而 FFD 方法是对每个固定的容量,顺序地将任务安排在满足容量约束、序号最小的机器上,FFD 的时间复杂度为 $O(n\log m)$,MULTIFIT 算法 k 次迭代的复杂度为 $O(n\log n+kn\log m)$, 显然,MULTIFIT 算法可能比 LPT 算法需要

* 收稿日期: 2000-06-21; 修改日期: 2000-10-16
基金项目: 国家 863 高科技发展计划基金资助项目(863-511-944-001)

作者简介: 李小平(1970－),男,重庆人,博士生,讲师,主要研究领域为 CIMS,并行计算;徐晓飞(1962－),男,湖南长沙人,博士,教授,博士生导师,主要研究领域为 CIMS,数据库,管理与决策信息系统;战德臣(1965－),男,山东胶县人,博士,教授,主要研究领域为 CIMS,数据库,管理与决策信息系统.

更多的时间找到解,但是可找到更好的解;(3) Bound Fit 算法是一个以预备算法[7]为基础的试探性迭代算法,即将预备算法所得容量试探性地减 1,如果能安排则将容量再减 1,直到不能安排或满足条件为止;而预备算法实质上是将 FFD 方法和 LPT 算法结合,即以所有任务加工时间平均值和排序后的第 1 个任务的加工时间两者中的最大值为容量采用 FFD 方法,并固定机器数为 m,如果不能安排所有任务,则剩余的任务采用 LPT 算法安排,得到初始容量,其时间复杂度为 $O(n\log m)$.所以 k 次试探的复杂度为 $O(n\log n+kn\log m)$.Bound Fit 和 MULTIFIT 得到的解的性能基本相同,在实际应用中我们也同样关心所需时间即迭代次数 K,然而即便是相同的 n 和 m,对于不同的任务,用某一算法 A 所需的 K 也不一样,故只能用平均迭代次数 $\overline{K}_A$ 来反映,我们通过大量的实验发现:Bound Fit 的平均迭代次数 $\overline{K}_{BF}$ 和 MULTIFIT 的平均迭代次数 $\overline{K}_{MF}$ 与任务机器比 $R(n,m)$ (每台机器上的平均任务数, $R(n,m)=n/m$) 和机器数 m 有关:当 $R(n,m)$ 较小或者对于任意的 $R(n,m)$ 中的 m 较小时, $\overline{K}_{BF}<\overline{K}_{MF}$,即 Bound Fit 快于 MULTIFIT;对于相同的 $R(n,m)$, $\overline{K}_{BF}$ 随 m 的增加而增加, $\overline{K}_{MF}$ 随 m 的增加变化不大,即较稳定.

本文结合 Bound Fit,LPT 和 MULTIFIT 提出 QUICKFIT 算法,该算法在 MULTIFIT 的上下界、FFD 容量大小的确定方面进行了改进.实验表明,QUICKFIT 算法在 $R(n,m)$ 或 m 较小时,性能与 Bound Fit 基本相同,而在 m 较大时,又具有 MULTIFIT 的稳定性,且 QUICKFIT 的平均迭代次数 $\overline{K}_{QF}<\overline{K}_{MF}$.

首先给出问题的描述及预备算法,其次给出 QUICKFIT 算法,之后给出时间复杂度和性能分析,最后给出实验结果及结论.

1 问题的描述及预备算法

本文选取的目标函数为总完成时间最短,设 $T=\{T_1,T_2,...,T_n\}$ 表示 n 个独立任务的任务集, $M=\{M_1,M_2,...,M_m\}$ 表示 m ($m\geq 2$)台机器的机器集;任务 T_i 的加工时间为 p_i 且 p_i 为整数, $i=1,2,...,n$,假设加工时间按非增的顺序排序,即 $p_1\geq p_2\geq ...\geq p_n$. C_i ($i=1,2,...,m$) 表示第 i 台机器的完成时间, $C_{\max}=\max\limits_{1\leq i\leq m}C_i$ 表示总完成时间, $C^*_{\max}$ 表示最优调度的总完成时间. $C_A[T,m]$ 表示用算法 A 将任务集 T 安排在 m 台机器上的总完工时间.

下面给出文献[7]的预备算法(简称 PA).假设以 $B^{(k)}$ 为参数时的预备算法为 PA($B^{(k)}$), L_j 为机器 M_j 上已安排任务的完成时间,则文献[7]中以 B 为参数的预备算法描述如下:

预备算法 **PA(B)**.

(1) 令 $B=\max\left\{\frac{1}{m}\sum_{i=1}^{n}p_i,\ p_1\right\}$;

(2) 令 $L_j=0=0$($j=1,2,...,m$), $i=1$;

(3) 若至少存在一个 j ($1\leq j\leq m$)使 $L_j+p_i\leq B$,则将 T_i 安排在使 $L_j+p_i\leq B$ 且序号最小的机器上;否则将 T_i 安排在使 L_j+p_i ($j=1,2,...,m$)最小的机器上;

(4) 若 $i<n$,令 $i=i+1$,转向(3);否则,停机.

2 QUICKFIT 算法

设当第 k 次用预备算法所得调度的总完成时间为 $C^{(k)}_{\max}$,其中 $k\geq 1$.

引理[7]. 当 $k\geq 3$ 时,若用预备算法求任务调度存在总完成时间 $C^{(k)}_{\max}\leq C^{(k-1)}_{\max}-1$ 的调度,则令 $B^{(k)}=C^{(k-1)}_{\max}-1$ 一定可得到 $C^{(k)}_{\max}\leq C^{(k-1)}_{\max}-1$ 的调度.

推广引理可得到如下定理:

定理 **1**. 当 $k\geq 3$ 时,若用预备算法求任务调度存在总完成时间 $C^{(k)}_{\max}\leq C^{(k-1)}_{\max}-j$ 的调度,则令 $B^{(k)}=C^{(k-1)}_{\max}-j$ 一定可得到 $C^{(k)}_{\max}\leq C^{(k-1)}_{\max}-j$ 的调度,其中 j 为大于等于 1 的整数.

证明方法同引理.

与引理类似,定理 1 中满足小于等于 $C^{(k-1)}_{\max}-j$ 的调度可能很多,而定理 1 找到的是最接近 $C^{(k-1)}_{\max}-j$ 的那个解,如果要找到小于 $C^{(k-1)}_{\max}$ 的某个指定的解,则有如下定理:

定理 2. 当 $k \geq 3$ 时,若用预备算法求任务调度存在总完成时间 $C_{\max}^{(k)} = C_{\max}^{(k-1)} - j$ 的调度,则令 $B^{(k)} = C_{\max}^{(k-1)} - j$ 一定可得到 $C_{\max}^{(k)} = C_{\max}^{(k-1)} - j$ 的调度,其中 j 为大于等于 1 的整数.

证明:设第 $k-1$ 次调用预备算法得到的解为 $C_{\max}^{(k-1)}$,如果存在总完成时间为 $C_{\max}^{(k-1)} - j$ 的调度,则第 k 次调用预备算法时令 $B^{(k)} = C_{\max}^{(k-1)} - j$,由于预备算法就是 FFD 算法的改进,所以预备算法得到的解至少是 $C_{\max}^{(k-1)} - j$,即 $C_{\max}^{(k)} \geq C_{\max}^{(k-1)} - j$;由定理 1 可知,当 $k \geq 3$ 时,若存在总完成时间 $C_{\max}^{(k)} \leq C_{\max}^{(k-1)} - j$ 的调度,则令 $B^{(k)} = C_{\max}^{(k-1)} - j$ 有 $C_{\max}^{(k)} \leq B^{(k)}$,故 $C_{\max}^{(k)} = C_{\max}^{(k-1)} - j$. □

显然, j 的选取即 $B^{(k)}$ 的大小直接影响到迭代次数, $B^{(k)}$ 越小,平均迭代次数就可能越少(Bound Fit 将 j 取为 1);但是,如果 $B^{(k)}$ 小于 $C_{\max}^{*}$,则得不到 $C_{\max}^{*}$.采用 MULTIFIT 折半迭代的思想, $B^{(k)}$ 取为 $\lceil (CL(k)+CU(k))/2 \rceil$,调用预备算法得到的解 $C_{\max}^{(k)} = C_{PA(B^{(k)})}[T,m]$,(1) 如果 $C_{\max}^{(k)}$ 小于等于 $B^{(k)}$,则由定理 1 可知, $C_{\max}^{*}$ 一定在区间 $[CL(k), C_{\max}^{(k)}]$上;(2) 如果 $C_{\max}^{(k)}$ 在区间($B^{(k)}, CU(k)$)上,则由 MULTIFIT 可知,下次迭代的下界 $CL(k+1) = B^{(k)}$,最终可由定理 2 求出解;(3) 如果 $C_{\max}^{(k)} > CU(k)$,则由 Bound Fit 算法试探 $CU(k)-1$ 是否能得到解.令 $B^{(0)} = \max\left\{\frac{1}{m}\sum_{i=1}^{n} p_i, p_1\right\}$,LPT 得到的解可能比调用一次预备算法得到的解更好,即 $C_{LPT}[T,m] < C_{PA(B^{(0)})}[T,m]$ 可能成立,故可用 $\min\{C_{LPT}[T,m], C_{PA(B^{(0)})}[T,m]\}$ 作为第 1 次 MULTIFIT 迭代的上界;如果 $B^{(0)}=\min\{C_{LPT}[T,m], C_{PA(B^{(0)})}[T,m]\}$,则找到最优解,停机.由以上分析,可将 LPT,MULTIFIT 和 Bound Fit 结合得到如下算法.

QUICKFIT 算法.

(1) 令 $B^{(0)} = \max\left\{\frac{1}{m}\sum_{i=1}^{n} p_i, p_1\right\}$, $k = 0$;

(2) $k = k+1$,若 $C_{LPT}[T,m] = B^{(0)}$,则停机.

(3) $k = k+1$,若 $C_{PA(B^{(0)})}[T,m] = B^{(0)}$,则停机.

(4) $k = k+1$, $CL(k) = B^{(0)}$, $CU(k) = \min\{C_{LPT}[T,m], C_{PA(B^{(0)})}[T,m]\}$.

(5) $B^{(k)} = \lceil (CL(k)+CU(k))/2 \rceil$, $C_{\max}^{(k)} = C_{PA(B^{(k)})}[T,m]$.

· 若 $C_{\max}^{(k)} \leq B^{(k)}$,则 $CU(k+1) = C_{\max}^{(k)}$, $CL(k+1) = CL(k)$,转(6).

· 若 $B^{(k)} < C_{\max}^{(k)} > CU(k)$,则 $CU(k+1) = C_{\max}^{(k)}$, $CL(k+1) = B^{(k)}$,转(6).

· 若 $C_{\max}^{(k)} > CU(k)$,则 $B^{(k+1)} = B^{(k)} - 1$,转(7).

(6) 如果 $\lceil (CU(k)-CL(k))/2 \rceil > 1$,则 $k = k+1$,转(5);否则停机.

(7) $k = k+1$, $C_{\max}^{(k)} = C_{PA(B^{(k)})}[T,m]$,如果 $C_{\max}^{(k)} \leq B^{(k)}$,则 $B^{(k+1)} = B^{(k)} - 1$,转(7);否则停机.

3 时间复杂度和性能分析

LPT 的时间复杂度为 $O(n\log n + n\log m)$,QUICKFIT 算法中调用预备算法和 FFD 的总次数为 $k-1$,所以 QUICKFIT 算法总的时间复杂度为 $O(n\log n + kn\log m)$.

由文献[7]可知,Bound Fit 的绝对性能 $R_m(BF) = \lim_{k\to\infty} R_m(MUL)$, $R_m(MUL) = 1.20 + (1/2)^k$,而 QUICKFIT 的绝对性能 $R_m(QF)$ 为:当执行步骤(7)时为 $R_m(QF) \leq R_m(MUL)$,否则为 $R_m(BF)$,即 QUICKFIT 的性能不会比 Bound Fit 和 MULTIFIT 差.所以 $R_m(QF) \leq \min\{R_m(MUL), R_m(BF)\}$.

几种算法的复杂度和性能比较见表 1.

Table 1 Complexity and performance of several algorithms
表 1 几种算法的复杂度和性能比较表

Algorithms① / Index②	LPT	MULTIFIT	Bound Fit	QUICKFIT
Time complexity③	$O(n\log n+n\log m)$	$O(n\log n+k_{MF}n\log m)$	$O(n\log n+k_{BF}n\log m)$	$O(n\log n+k_{QF}n\log m)$
Performance(≤)④	$\frac{4}{3}-\frac{1}{3m}$	$1.20+(1/2)^k$	$\lim\limits_{k\to\infty}R_m(MUL)$	$\min\{R_m(MUL),R_m(BF)\}$

①算法,②指标,③时间复杂度,④性能.

4 实验结果

我们随机产生 100 个加工时间为 1～1000 的任务,用 MULTIFIT,Bound Fit 和 QUICKFIT 进行 100 次实验,得到数据见表 2(M 为机器数, T_{BFQ} 为 100 次实验中 Bound Fit 快于 QUICKFIT 的次数, T_{QBM} , T_{QBB} 分别为 100 次实验中 QUICKFIT 得到的解优于 MULTIFIT 和 Bound Fit 的次数).

Table 2 Experimental result of several algorithms
表 2 几种算法的实验结果

M	The average number of iterations①			T_{BFQ}	T_{QBM}	T_{QBB}	M	The average number of iterations			T_{BFQ}	T_{QBM}	T_{QBB}
	$\overline{K}_{MF}$	$\overline{K}_{BF}$	$\overline{K}_{QF}$					$\overline{K}_{MF}$	$\overline{K}_{BF}$	$\overline{K}_{QF}$			
2	15	2.43	2.6	39	35	28	27	11	22.74	6.76	0	0	0
3	14.31	3.38	3.61	40	20	18	28	11	21.27	6.93	1	0	0
4	14	3.76	3.84	32	14	11	29	11	22.1	7	0	0	1
5	13.77	4.84	4.29	37	17	16	30	11	22.42	6.91	0	0	1
6	13.26	5.48	4.63	33	2	2	31	11	24.31	7.1	0	2	2
7	13	6.26	4.79	23	3	2	32	10.99	25.09	7.44	0	0	0
8	13	6.63	4.81	28	3	4	33	10.97	26.03	7.33	1	2	1
9	12.95	8.66	5.2	17	5	5	34	10.94	25.03	7.03	0	0	0
10	12.67	10.05	5.26	8	1	1	35	10.89	27.32	7.23	1	0	1
11	12.36	9.55	5.4	11	1	1	36	10.81	25.98	7.26	1	0	0
12	12.13	10.59	5.47	9	1	1	37	10.85	28.95	7.12	2	1	3
13	12	13.24	5.85	3	1	0	38	10.68	28.12	7.39	4	1	1
14	12	12.39	5.62	7	1	1	39	10.59	24.57	7.32	10	2	3
15	12	13.13	5.8	0	0	0	40	10.51	25.04	7.52	16	1	1
16	11.99	14.86	6.01	3	0	1	41	10.51	24.93	7.2	21	0	0
17	11.96	14.56	6.02	2	0	0	42	10.44	21.41	7.09	18	0	0
18	11.91	16.34	6.13	3	0	0	43	10.48	18.45	7.21	36	0	0
19	11.74	17.2	6.2	0	0	0	44	10.29	13.17	6.86	48	2	0
20	11.55	19.18	6.65	1	1	1	45	10.23	12.07	6.82	52	2	0
21	11.45	18.06	6.31	1	0	1	46	10.15	8.45	6.84	60	3	0
22	11.3	21.37	6.58	0	0	0	47	10.14	7.23	6.72	61	7	0
23	11.19	20.77	6.71	0	0	0	48	10.09	6.39	6.2	68	15	0
24	11.04	20.45	6.87	0	0	0	49	10.08	7.44	6.46	60	16	0
25	11.02	22.75	6.83	1	0	0	50	10.04	4.92	5.47	56	30	0
26	11.01	19.84	6.55	1	0	0	51	10.04	4.96	4.93	52	37	0

①平均迭代次数.

图 1 反映了 $\overline{K}_{BF}$, $\overline{K}_{MF}$ 和 $\overline{K}_{QF}$ 随 $R(n,m)$ 的变化趋势,即在给定任务数时,各参数随机器数增加($R(n,m)$ 减小)的变化情况;图 2~图 4 反映的是对于给定的 $R(n,m)$ 各参数随机器数变化的情况.由表 2 及图 1~图 4(其中 K 为平均迭代次数, m 为机器数)可以看出:

(1) 在 m 较小($m<13$)或者 $R(n,m)\le 2.2$ 时, $\overline{K}_{BF}<\overline{K}_{MF}$,即 Bound Fit 快于 MULTIFIT,而且在 $R(n,m)$ 较小时 Bound Fit 能得到比 MULTIFIT 更好的解.在实际中 m 通常不会很大,所以 Bound Fit 在应用中优于 MULTIFIT;甚至在 m 较小($m<6$)或者 $R(n,m)\le 2.2$ 时,还会有 $\overline{K}_{BF}<\overline{K}_{QF}$.因为 Bound Fit 不需要调用 LPT,但在 m 较小时 Bound Fit 得到的解不如 QUICKFIT,因为 Bound Fit 没有调用 LPT.另外, $\overline{K}_{BF}$ 随 $R(n,m)$ 和 m 的改变产生的变化较大.

(2) $\overline{K}_{MF}$ 随 $R(n,m)$ 和 m 的改变产生的变化不大,但 $\overline{K}_{QF}$ 最多约为 $\overline{K}_{MF}$ 的 2/3,而且在大多数情况下 MULTIFIT 得到的解不如 QUICKFIT.

(3) $\overline{K}_{QF}$ 随 $R(n,m)$ 和 m 的改变产生的变化不大,对于任意的 $R(n,m)$ 和 m 都有 $\overline{K}_{QF} < \overline{K}_{MF}$;对于 $R(n,m) \geq 2.2$ 且 $m \geq 7$ 有 $\overline{K}_{QF} < \overline{K}_{BF}$,而且在 $m < 7$ 时,Bound Fit 所需的迭代次数可能小于 QUICKFIT,但 QUICKFIT 能得到比 MULTIFIT 和 Bound Fit 更优的解.

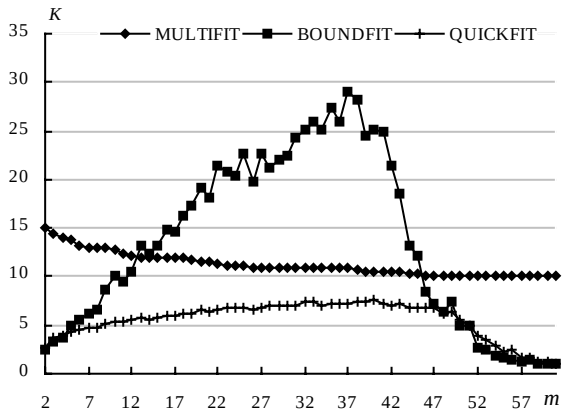

Fig.1 Curves of K changing with m when tasks=100

图1 任务数为100时 K 随 m 变化曲线图

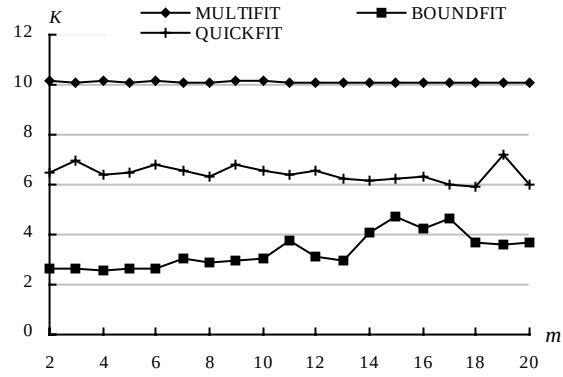

Fig.2 Curves of K changing with m when $R(n,m)$=2

图2 $R(n,m)$=2时 K 随 m 变化曲线图

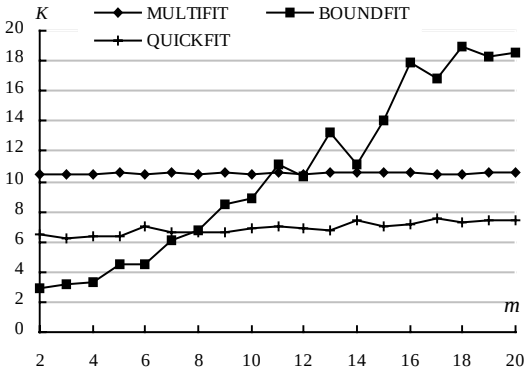

Fig.3 Curves of K changing with m when $R(n,m)$=2.5

图3 $R(n,m)$=2.5时 K 随 m 变化曲线图

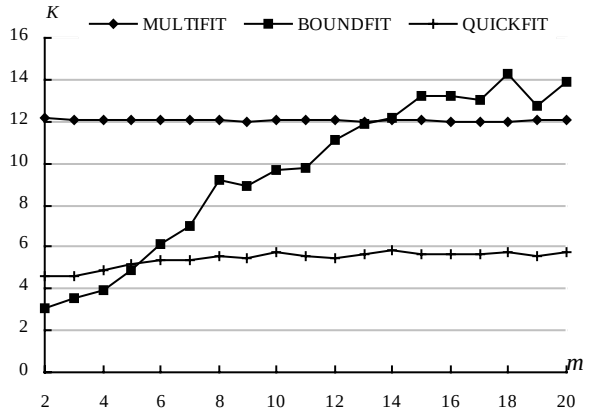

Fig.4 Curves of K changing with m when $R(n,m)$=8

图4 $R(n,m)$=8时 K 随 m 变化曲线图

5 结 论

通过改进 MULTIFIT 的上下界、FFD 容量确定方案,结合 Bound Fit,LPT 和 MULTIFIT 等算法,提出 QUICKFIT 算法.理论分析和试验数据表明,QUICKFIT 算法具有较少的平均迭代次数,并能得到较好的解;在 $R(n,m)$ 或 m 较小时,性能与 Bound Fit 基本相同,而在 m 较大时,又具有 MULTIFIT 的稳定性.对于生产调度问题、操作系统的进程、线程调度等 SMSP 问题有一定的理论和实用价值.但实际的调度中还有许多问题属于 NMSP,如何用 QUICKFIT 算法或类似的算法解决这类问题值得深入研究.

References:

[1] Ullman, J.D. Complexity of sequencing problems. In: Computer and Job/Shop Scheduling Theory. New York: John Wiley and Sons, Inc., 1976.

[2] Lee, C.Y. Parallel machines scheduling with nonsimultaneously machine available time. Discrete Applied Mathematics, 1991,30(1):53~61.

[3] Kellerer, H. Algorithms for multiprocessor scheduling with machine release times. Institute of Industrial Engineers Transactions, 1998,30(11):991~999.

[4] Chang, S.Y., Hwang, H.C. The worst-case analysis of the MULTIFIT algorithm for scheduling nonsimultaneous parallel machines. Discrete Applied Mathematics, 1999,92(2/3):135~147.

[5] Graham, R.L. Bounds on multiprocessing timing anomalies. SIAM Journal on Applied Mathematics, 1969,17(2):416~429.

[6] Jr Coffman, F.G., Graey, M.R., Johnson, D.S. An application of bin-packing to multiprocessor scheduling. SIAM Journal on Computing, 1978,7(1):1~17.

[7] Kang Yi-mei, Zheng Ying-ping. Independent tasks scheduling on identical parallel processors. Acta Automatica Sinica, 1997,23(1):81~84 (in Chinese).

附中文参考文献:

[7] 康一梅,郑应平.同等并行处理机上独立任务的调度.自动化学报,1997,23(1):81~84.

A Quick Algorithm for Independent Tasks Scheduling on Identical Parallel Processors*

LI Xiao-ping, XU Xiao-fei, ZHAN De-chen

(*Department of Computer Science and Engineering*, *Harbin Institute of Technology*, *Harbin* 150001, *China*)

E-mail: Li_xiaoping@0451.com

http://www.hit.edu.cn

Abstract: QUICKFIT, a quick algorithm is developed by analyzing the properties of the preparatory algorithm of Bound Fit and integrating Bound Fit with MULTIFIT for scheduling independent tasks on identical parallel processors to minimize the make span. For the same tasks and machines, QUICKFIT needs a fewer iterations than MULTIFIT and Bound Fit do to obtain the same make span. Experimental results show that the greater the task-to-machine ratio is, the better the performance of QUICKFIT is than MULTIFIT and Bound Fit, and the make span of QUICKFIT is the least among those of LPT, MULTIFIT and Bound Fit in most of cases, and QUICKFIT is suitable for the large scale identical scheduling.

Key words: identical multiprocessor scheduling; bin packing; LPT algorithm; MULTIFIT algorithm; task-to-machine ratio

---

* Received June 21, 2000; accepted October 16, 2000
Supported by the National High Technology Development 863 Program of China under Grant No.863-511-944-001