基于多种群及水平集的任务调度算法

兰 舟,孙世新
LAN Zhou,SUN Shi-xin

电子科技大学 计算机科学与工程学院,成都 610054
School of Computer Science and Engineering,University of Electronic Science and Technology of China,Chengdu 610054,China
E-mail:lzlanzhou@163.com

LAN Zhou,SUN Shi-xin.Task scheduling algorithm based on multi-population and level set.Computer Engineering and Applications,2008,44(19):53-56.

Abstract: Task scheduling is one of the most important issues to achieve high performance for multiprocessor systems.With the extensive studies of this issue,many new methods including Genetic Algorithms (GAs) have been introduced in this field. However,traditional GAs has two serious demerits,premature convergence and evolutional stagnation.To overcome those weaknesses,a novel GA,namely Multi-Population and Level Set based task scheduling algorithm (MPLS),had been developed for multiprocessor systems.MPLS employed the idea of multi-population coevolution to ensure the population diversity and introduced the concept of level set into task scheduling to speed up the iterative convergence.Based on the third-party benchmark,MPLS' performance had been compared with other three GAs,including GTMS,MSGS and NGS.The comparative results show that MPLS can obtain much better schedule lengths than GTMS and MSGS,and slightly better than NGS.MPLS keeps the population diversity with the highest level of all the four GAs.Consequently,MPLS outperforms the others GAs in terms of schedule length and population diversity.

Key words: multi-population;level set;population diversity;Genetic Algorithm(GA);task scheduling

摘 要:随着任务调度问题的广泛研究,包括遗传算法在内的许多新方法被引入到任务调度领域。然而,传统的遗传算法存在早熟收敛和后期进化停滞两个严重不足。为了克服这些不足,提出了算法 MPLS。MPLS 算法采用多种群共同进化的思想来维持种群多样性。同时,MPLS 算法将水平集概念引入到任务调度研究中,以改进迭代收敛速度。基于第三方测试数据集,将 MPLS 的性能和 GTMS、MSGS 和 NGS 算法进行了对比。比较结果表明,MPLS 算法获得的调度长度远好于 GTMS、MSGS 算法,略好于 NGS 算法。MPLS 算法能将种群多样性维持在一个很高的水平。MPLS 算法在调度长度和种群多样性方面要优于其它算法。

关键词:多种群;水平集;种群多样性;遗传算法;任务调度
DOI:10.3778/j.issn.1002-8331.2008.19.016 **文章编号**:1002-8331(2008)19-0053-04 **文献标识码**:A **中图分类号**:TP316.4

1 引言

遗传算法(Genetic Algorithms,GAs)是一种借鉴生物界自然选择思想和自然遗传机制的全局随机搜索算法。因其全局搜索能力,群体搜索策略带来的高效性,与目标梯度信息的无关性以及简单通用性,使遗传算法得到了广泛的研究和应用。Hou 等率先将遗传算法引入到任务调度问题研究中,为用遗传算法解决任务调度问题提供了基本框架[1]。随后,Tsuchiya、Correa、陆鑫达、Wu、Dhodhi 等在应用遗传算法求解任务调度问题方面进行了研究[2-6]。钟求喜等对标准遗传算法进行改进,设计了三种新的遗传算子(本文称该算法为 GTMS)[7]。Yao 等提出了主序列遗传调度(MSGS)算法,将主序列的信息应用于初始种群的构造和遗传操作中,能以较短的运行时间获得较好的调度结果[8]。

然而,传统的遗传算法有两个严重的不足,即容易过早收敛,以及在进化后期搜索效率较低。其原因是早期的选择压力会导致个体的有效等位基因缺失,使得种群内的个体在模式上很快趋同,种群多样性迅速下降,继而导致遗传操作很难产生新的优良个体。因此,保证种群多样性是改善上述不足的根本途径和有效手段。

针对遗传算法的这两个不足,许多学者从不同角度进行了探讨。Kwok、钟求喜等采用多种群进化的思想来维持种群的多样性[9,10]。杨启文、李庆华等分别采用二元变异算子,平衡变异算子来对传统二元变异操作进行改进,以提高种群的多样性[11,12]。李庆华等还将水平集的概念引入到遗传算法的选择操作中,以加快算法的进化速度[12]。Jiao 等则采用免疫调节机制来维持种群的多样性[13]。还有学者采用混合遗传算法,以改善算法的收敛速度等性能[14]。

为了解决传统遗传算法存在的不足,本文提出了任务调度遗传算法 MPLS,该算法采用水平集及多种群进化的思想来保证

作者简介:兰舟(1969-),博士研究生,主要研究方向:网络计算技术与高性能并行计算;孙世新(1940-),教授,博士生导师,主要研究方向:网络计算技术,并行/分布式计算及其应用,信息压缩技术,数值计算与组合算法等。

收稿日期:2007-09-28 **修回日期**:2007-12-19

种群的多样性,以改善早熟收敛和进化后期收敛速度慢等问题。

2 任务调度模型

本文采用广泛接受的多处理器系统和并行应用模型[3,15]。任务调度模型可以用七元组有向无环图(DAG)表示,$G=(P,V,E,W,C,ST,FT)$。多处理器系统 P 由 m 个充分互连的同构处理器组成,处理器对之间的通信效率相同,每个处理器可以同时进行计算和通信。顶点集 V 由 v 个非剥夺任务组成,v_i 表示第 i 个任务。边集 E 表示任务间的优先关系,有向边 $e_{i,j}\in E$ 表示 v_j 在收到 v_i 的数据之前是不能开始执行的。如果 $e_{i,j}\in E$,则 v_i 称之为 v_j 的父任务,v_j 称之为 v_i 的子任务。$\boldsymbol{W}$ 是长度为 v 的向量,w_i 表示 v_i 的运行开销。$\boldsymbol{C}$ 为 $v\times v$ 矩阵,$c_{i,j}$ 表示从 v_i 到 v_j 的通信开销。如果 v_i,v_j 分配到同一处理器,则 $c_{i,j}$ 可以忽略不计。ST 是任务开始时间集,st_i 表示 v_i 的最早开始时间;FT 为任务完成时间集,ft_i 表示 v_i 开始于 st_i 的完成时间。

在一个 DAG 中,没有任何父任务和子任务的任务分别称之为开始任务和结束任务。不失一般性,假定 DAG 仅有一个开始任务和一个结束任务[15],分别用 n_s,n_e 表示。

定义 1 调度长度(Schedule Length,SL)为从 n_s 开始运行到 n_e 结束的时间。

任务调度的目标就是在满足约束关系的前提下,把 V 中的任务合理分配到 P 中运行,使 SL 最短。

3 MPLS 算法

遗传算法使用群体搜索技术,通过对当前种群施加选择,交叉,变异等一系列操作,从而产生新一代种群,并逐步使种群进化到包含或接近最优解的状态。针对传统遗传算法存在的不足,MPLS 算法采用多种群进化策略,以保证种群多样性为主要手段来改善早熟收敛及后期搜索效率低的问题;并用水平集选择来改善进化速度。

3.1 编码和解码

编码方案是遗传算法的重要组成部分,关系到遗传算子是否容易操作,目标函数是否容易计算,是影响算法性能的关键因素。在任务分配到处理器的策略确定之后,对任务调度来说,任务调度的顺序或任务调度序列是最重要的。因此,本文采用任务调度序列 SQ 来构成染色体,染色体长度为 v,sq_i 为染色体中的第 i 个任务,是第 i 个分配的任务。很显然,为保证调度方案的合法性和完整性,每个任务在染色体中当且仅当出现一次,且父任务应排在子任务之前。

至于任务如何分配,分配到哪个处理器,在处理器中的执行顺序等,则由解码算法完成。解码时,MPLS 算法按照染色体对应的调度序列,依次将任务分配到处理器中,直至所有任务均已分配。在分配某一任务时,MPLS 算法借鉴启发算法中贪心算法的思想,将该任务分配到能使其获得最早开始时间的处理器上,以此缩短调度长度。

3.2 种群初始化

初始种群作为遗传算法的迭代起点,其质量好坏将直接影响算法的性能。现有大多数遗传算法依据任务高度值来对任务排序而产生任务调度序列,使得有些可行解不可能出现在初始种群中,人为地缩小了遗传算法的搜索空间[3],很难保证初始种群的多样性。

MPLS 算法在构造初始种群时,仅依据任务间优先关系,而不是一个具体的优先权值,使得任何可行解均有可能在初始种群中出现。一个任务的所有父任务均已被调度,则称之为就绪任务(ready task)。就绪任务组成的集合称为就绪任务集。很显然,一个任务能被调度的前提条件就是该任务为就绪任务。MPLS 算法在构造一条染色体时,首先随机从就绪任务集中选取一个就绪任务(初始时就绪任务集中仅有 v_s)并添加至 SQ,然后从就绪任务集中去掉已选取任务,并添加新的就绪任务到就绪任务集中。重复上述步骤直至就绪任务集为空,此时就完成了一条染色体的构造。重复构造若干条染色体就可完成种群初始化。很显然,构造出的染色体均是合法的染色体,并且能随机地分布在搜索空间中,保证了初始种群的多样性。

3.3 交叉和变异算子

在交叉算子中,MPLS 算法采用两点交叉,按交叉概率 p_c 对两条父染色体进行交叉操作。MPLS 算法根据一条染色体中任务的顺序来调整另一条染色体交叉部分任务的顺序,从而得到这些任务在另一条染色体中顺序特征,以达到交叉的目的。

在变异算子中,MPLS 算法采用单点变异,按变异概率 p_m 对父染色体进行变异。对父染色体中需要变异的任务,随机地在其最近的左边父任务和右边子任务之间改变位置,其实质就是改变任务的调度顺序。

很容易证明,经过上述交叉和变异操作得到的子染色体仍然是合法的染色体,不需要额外对染色体进行操作。

3.4 选择算子

染色体经解码后得到的调度长度可直观反映染色体的好坏,故可定义适应度函数为:

$$f(i)=W_{sum}-SL_i \tag{1}$$

其中 W_{sum} 为所有任务运行开销之和,SL_i 为染色体 i 所对应的调度长度。

杂交和变异操作完成后,MPLS 算法采用和文献[7]类似的演化策略,从父代和子代中选择若干个个体形成下一代种群。

定义 2 假设种群规模为 n,设有第 t 代父代种群 $P(t)=\{ch_{p1}^t,ch_{p2}^t,\cdots,ch_{pn}^t\}$及其子代种群 $S(t)=\{ch_{s1}^t,ch_{s2}^t,\cdots,ch_{sn}^t\}$,令 $f_t=\dfrac{1}{2n}\sum_{i=1}^{n}(f(ch_{pi}^t)+f(ch_{si}^t))$,则称集合 $H_{f_t}=\{ch_i^t\in P(t)\cup S(t)\mid f(ch_i^t)\geqslant f_t\}$为关于 $P(t)$和 $S(t)$的水平集。

在进行选择操作时,水平集 H_{f_t}中的个体直接进入第 $t+1$ 代父代种群 $P(t+1)$,不足部分个体则从$(P(t)\cup S(t)-H_{f_t})$选取。为保证种群多样性,要求选取进入 $P(t+1)$的个体互不相同。如经过上述两种选取后,个体数量仍然不足,则随机地从 $S(t)$中选取。为了确保最优个体不会丢失,或其良好模式不被破坏,在选择过程中,将第 t 代父代种群 $P(t)$及子代种群 $S(t)$中的最优个体直接进入第 $t+1$ 代父代种群 $P(t+1)$。

很显然,通过水平集选择使得种群的整体质量不断提高,改善了收敛速度慢的问题;同时通过约束被选个体两两互不相同,维持了种群的多样性。

3.5 多种群进化

自然界中,很多物种会以多个种群的形式存在不同的环境中,独立进化,种群内个体逐渐趋同。但也会由于一些偶发因素,如自然环境改变、种群迁徙、个体逃逸等,使得一个种群内会有新的个体移民进来,增加了种群多样性,从而促进种群的进化。多种群进化的思想正是基于上述自然现象提出来的。与单一种群进化不同的是,多种群进化将初始种群划分为若干子种群,各个子种群按照一定的模式独立进化,适当的时候在子

种群之间交换信息，从而维持种群的多样性，以此促进各个种群的进化。

多种群进化需要确定子种群数目、子种群规模、移民频率、移民数量、移民方式等，目前尚没有公用的准则，大多靠经验和反复测试确定。针对任务调度算法特点，MPLS 算法确定子种群数目为 4；子种群规模为 20；种群每进化 20 代移民一次；移民数量为 3，其中两个为最好个体，另外一个随机选取，移民后取代对方种群内最差的三个个体；所有子种群组成一个环，某个子种群向相邻的两个子种群进行移民。

为了保证子种群的多样性，MPLS 算法在构造完初始种群之后，将初始种群的个体按适应度值从大到小排序，并从左向右每 4 个个体划分为一组，将每组中的个体按每子种群一个随机投放，这样分配之后，子种群中个体的适应度值分布较为均匀，其种群多样性较容易得到保证。

3.6 MPLS 算法框架

综合上述各个部分，得到 MPLS 算法的总体框架，如图 1 所示。

```
1.  initialize population
2.  construct subpopulations
3.  for i←1 to maximum of iterative generations
4.      for all the subpopulations
5.          make crossover operator with p_c
6.          make mutation operator with p_m
7.          evaluate subpopulation
8.          select parent population
9.      end for
10.     if migration condition is satisfied then
11.         migrate the selected individuals
12. end for
13. return the best schedule scheme and its SL
```

图 1　MPLS 算法框架

4　算法仿真及分析

4.1　仿真环境

为了简化和公平起见，本文利用文献[16]提供的测试数据集 Bench.zip，选用其中任务数为 100，i 分别为 1、2、3，共 15 个任务图作为测试用例。

为了对比 MPLS 算法的性能，本文实现了 GTMS[7]、MSGS[8] 两种遗传算法，还设计了一种常规遗传算法 NGS(Normal Genetic algorithm)。NGS 算法采用的编码和解码方案、交叉和交异算子、种群初始化等和 MPLS 一样，但不采用水平集选择和多种群进化思想，而采用赌轮选择和保优相结合的选择策略。

在仿真测试中，选用的参数为处理器数 8、种群规模 80、迭代代数 1 000，其它参数如表 1 所示。

表 1　四种遗传算法的其它参数

算法	P_c	P_m	P_{NCX}	$P_{migration}$
GTMS	1.0	0.05	0.8	0.2
MSGS	0.8	0.02		
NGS	0.6	0.20		
MPLS	0.6	0.20		

4.2　仿真结果及分析

为了对比种群多样性，先定义多样性度量指标。

定义 2　设有种群 $P(t)=\{ch_1,ch_2,\cdots,ch_n\}$，$n$ 为种群规模。假设 k 为种群中不同个体的数目，则种群多样性度量指标 D_w 为：

$$D_w=\frac{2k\sum_{i=1}^{n}\sum_{j=i+1}^{n}a_{i,j}}{n^2\cdot(n-1)} \tag{2}$$

其中：

$$a_{i,j}=\begin{cases}0 & \text{if } ch_i=ch_j\\ 1 & \text{otherwise}\end{cases} \tag{3}$$

很显然，D_w 可直观地反映种群多样性的好坏。当种群中所有个体相同时，D_w=0，此时种群的多样性最差；当种群中所有个体互不相同时，D_w=1，此时种群的多样性最好。

表 2 对比了 4 种算法对测试数据运行后得到的结果，表中迭代代数为取得最优调度长度的代数。从表中可以看出，MPLS 算所获得的调度长度均要明显优于 GTMS 和 MSGS 算法；多数情况优于 NGS 算法，其余情况和 NGS 算法相当。获得最优调度长度的迭代代数，MPLS 算法普遍要少于 NGS 算法，绝大多数情况下要远少于 GTMS 和 MSGS 算法。4 种算法的平均种群多样性指标 Div，MPLS 算法最好，除 t100_8_1 为 86.93%外，其余情况均在 95%以上；GTMS 算法次之，主要维持在 78%~90%之间；NGS 算法则普遍维持在 49%~60%之间；MSGS 算法最差，除 t100_20_3 为 10.34%外，其余情况均在 6.5%以下。其原因主要是因为 MPLS 和 GTMS 算法采用了利于维持种群多样

表 2　算法性能对比

测试数据	最优调度长度				迭代代数				平均种群多样性 D_w			
	GTMS	MSGS	NGS	MPLS	GTMS	MSGS	NGS	MPLS	GTMS	MSGS	NGS	MPLS
t100_20_1	556	790	543	538	495	610	400	108	7 852	636	5 919	9 795
t100_20_2	1 155	1 431	1 142	1 127	174	259	751	74	8 364	629	5 935	9 770
t100_20_3	506	734	501	499	252	702	324	126	9 179	1 034	5 971	9 769
t100_40_1	1 072	1 177	1 033	1 033	283	195	493	120	8 855	554	5 860	9 700
t100_40_2	1 922	2 039	1 884	1 870	259	168	78	114	8 977	362	5 749	9 818
t100_40_3	1 029	1 131	1 024	1 024	941	239	38	85	8 711	562	5 818	9 831
t100_50_1	1 434	1 473	1 416	1 404	458	42	146	68	8 495	417	4 912	9 733
t100_50_2	2 007	2 073	1 999	1 999	754	892	31	52	8 750	151	5 581	9 816
t100_50_3	1 104	1 275	1 088	1 088	761	999	27	27	8 324	335	5 655	9 843
t100_60_1	1 367	1 404	1 354	1 342	699	777	29	54	8 194	98	4 962	9 792
t100_60_2	2 066	2 223	2 060	2 056	509	794	76	22	8 483	127	5 499	9 851
t100_60_3	1 338	1 404	1 334	1 334	361	27	88	19	7 884	287	5 149	9 833
t100_80_1	1 729	1 716	1 713	1 702	583	55	15	12	7 906	105	964	8 693
t100_80_2	2 853	2 825	2 836	2 836	603	623	14	22	7 683	80	1 046	9 519
t100_80_3	1 725	1 753	1 720	1 720	404	722	10	11	7 882	100	967	9 677

性的演化选择策略。MPLS 算法还引入了水平集选择策略,并设计了多层次选择,其种群多样性要好于 GTMS 算法。NGS 和 MSGS 算法则仅采用了常规赌轮选择和保优相结合的选择策略。MSGS 算法在遗传操作中还考虑了主序列信息,缩小了算法的搜索空间,其种群多样性要差于 NGS 算法。

5 结束语

本文基于多种群及水平集提出了一种新颖的遗传调度算法 MPLS,该算法针对传统遗传算法的不足,采用多种群共同进化的思想来维持种群的多样性,并用水平集选择策略来提高算法的收敛速度。为了验证算法的有效性,本文采用第 3 方测试数据集,将 MPLS 算法和其它 3 种不采用多种群及水平集思想的 3 种算法进行了对比。对比实验表明,MPLS 算法取得的最优调度长度及收敛速度要优于其它 3 种算法,MPLS 算法的种群多样是可以得到保证的,说明 MPLS 算法采用多种群进化及水平集选择策略来改善传统遗传算法存在的不足是可行的,效果是肯定的。

参考文献:

[1] Hou E S H,Ansari N,Ren H.A genetic algorithm for multiprocessor scheduling[J].IEEE Transactions Parallel and Distributed Systems,1994,5(2):113-120.

[2] Tsuchiya T,Osada T,Kikuno T.Genetics-based multiprocessor scheduling using task duplication[J].Microprocessors and Microsystems,1998,22(3/4):197-207.

[3] Correa R C,Ferreira A,Rebreyend P.Scheduling multiprocessor tasks with genetic algorithms[J].IEEE Transactions Parallel and Distributed Systems,1999,10(8):825-837.

[4] 陆鑫达,郑飞,陈楚询.异构计算机系统的任务调度算法 SMT-GA[J].小型微型计算机系统,1999,20(4):241-245.

[5] Wu A S,Yu H,Jin S,et al.An incremental genetic algorithm approach to multiprocessor scheduling[J].IEEE Transactions Parallel and Distributed Systems,2004,15(9):824-834.

[6] Dhodhi M K,Ahmad I,Yatama A,et,al.An integrated technique for task matching and scheduling onto distributed heterogeneous computing systems[J].Journal of Parallel and Distributed Computing,2002,62(9):1338-1361.

[7] 钟求喜,谢涛,陈火旺.基于遗传算法的任务分配与调度[J].计算机研究与发展,2000,37(10):1197-1203.

[8] Yao W,You J,Li B.Main sequence genetic scheduling for multiprocessor systems using task duplication[J].Microprocessors and Microsystems,2004,28(5):85-94.

[9] Kwok Y K,Ahmad I.Efficient scheduling of arbitrary task graphs to multiprocessors using a parallel genetic algorithm[J].Journal of Parallel and Distributed Computing,1997,47(1):58-77.

[10] 钟求喜,谢涛,陈火旺.任务分配与调度的共同进化算法[J].计算机学报,2001,24(3):58-77.

[11] 杨启文,蒋静坪,张国宏.遗传算法优化速度的改进[J].软件学报,2001,12(2):270-275.

[12] 李庆华,杨世达,阮幼林.基于水平集的遗传算法优化的改进[J].计算机研究与发展,2006,43(9):1624-1629.

[13] Jiao Li-cheng,Wang Lei.A novel genetic algorithm based on inmunity[J].IEEE Trans on Systems,Man and Cybernetics,2000,30(5):552-561.

[14] Barada H,Sait S M,Baig N.A simulated evolution approach to task matcging and scheduling in heterogeneous computing environments[J].Engineering Applications of Artificial Intelligence,2002,15(9):491-500.

[15] Darbha S,Agrawal D P.Optimal scheduling algorithm for distributed-memory machines[J].IEEE Transactions on Parallel and Distributed Systems,1998,9(1):87-95.

[16] Davidvic T,Crainic T G.Benchmark-problem instances for static scheduling of task graphs with communication delays on homogeneous multiprocessor systems[J].Computers & Operation Research,2006,33(8):2155-2177.

(上接 52 页)

表 1 在加密及三种测试下图像性能的具体表现

类型	参数	MSE	SNR/dB	PSNR/dB
加密	S 盒及 P 值	4 181.053 1	6.734 5	11.917 9
剪切测试	剪切 1/4	4 452.498 2	6.461 4	11.644 8
	剪切 1/16	1 127.839 9	11.869 4	17.608 3
噪声测试	标准偏差 0.01	645.284 4	14.849 9	20.033 3
	标准偏差 0.02	1 240.338 6	12.012 0	17.195 4
分布均匀性测试	无	19 608.821 9	0.022 882	5.206 3

都很少,说明其加密性能好,而其他测试结果也都表明其抗剪切和抗噪音的能力很强,分布均匀性测试效果也是明显非常理想。只是在剪切测试中其对图像品质的评价的低估或高估的缺点又显露出来,从图像上可以明显看出,剪切解密后的图比密图的品质要高。

5 算法比较

与基本算法相比,改进算法在加密与测试效果来看,性能相当,对于混沌随机以及 S 盒置换的加密,由于它们都是非线性的部件,同时 S 盒是基于 AES 中 Rijndael 算法中的 S 盒设计,其抗差分密码分析和线性密码分析方面都很强。所以敌手攻击的主要是穷举法。密钥空间大大增加会使安全性大大加强。

如果混沌理论中的初始的参数加入到密钥中,这里有 7 个数位,用 4 位二进制数表示 1 位则需要增加 28 位密钥。对于增加28 位密钥,则对于搜索的难度就增加了 2^{28} 倍。

如果使用混沌产生的 p 个的 1~30 的序号数组, 由于其最大值为 30,所以需要用 5 位二进制表示 1 个元素,增加密钥的位数为 $5p$,搜索的难度就增加了 2^{5p} 倍。例如这里 p=16,则其搜索的难度就增加了 2^{80} 倍。

6 结束语

本文在基于 S 盒图像置乱加密算法基础上,进一步将其与混沌置乱算法融合运用,弥补了二者单独使用时的不足,特别是加入了混沌理论使用多轮不同混沌序列号的 S 盒加密,提高了加密图像信息的安全性,也增加了解密难度。

本算法与同类算法相比,优点在于加密速度快、安全性能大大提高、可以得到精确的恢复图像;并且具有很好的抗差分密码分析以及线性密码分析的能力和抗剪切能力,但 S 盒置乱加密的理论分析还有待进一步完善。

参考文献:

[1] 易开祥,孙鑫,石教英.一种基于混沌序列的图像加密算法[J].计算机辅助设计与图形学学报,2000,12(9):672-676.

[2] 王丽娜,张焕国.信息隐藏技术与应用[M].武汉:武汉大学出版社,2003.

[3] 王丽娜,郭迟,李鹏.信息隐藏技术实验教程[M].武汉:武汉大学出版社,2003.