

任务分配与调度中的遗传算法:知识表示与遗传算子研究^{*}

Representation and Genetic Operators in Tasks Matching & Scheduling by Genetic Algorithms

钟求喜 谢 涛 陈火旺

(国防科技大学计算机学院 长沙 410073)

Abstract Task matching and scheduling play an important role in parallel and distributed systems. In order to use genetic algorithms(GAs) for tasks matching and scheduling, not only appropriate representations of solutions but also genetic operators' efficiency and generality are very important. In this paper, analysis between problem space and representation space is given at the first. Then based on the representation of permutation, two general efficient genetic operators are proposed, order crossover (OCX) and migration. OCX generates new schedules with heuristic due to the problem space with constraints among tasks. Migration transfers a task from one processor to another within a schedule. The simulation results of algorithms and conclusions are given at last.

Keywords Tasks matching & scheduling, Genetic algorithms, Order crossover, Migration

1 引言

分布与并行系统中的任务分配与调度,对发挥系统的并行性能和保持负载平衡有重大意义,也是被公认的NP问题^[1]。Richard F. Freund^[2]等通过对一个假设的具有多层并发度计算需求的计算实例的计算分析表明,对于普通串型机需要100个时间单位,对于向量机需要50个时间单位,而采用由5种机器组成的异构网络计算系统计算时仅需5个时间单位,而所需的代价是必须进行适当的任务分解、分配与调度。

遗传算法(GA)^[6]是一类用机器模拟生物界自然选择和自然遗传机制来求解复杂问题的随机搜索和优化方法。目前,已经有许多学者,将健壮的遗传算法用于任务分配与调度^[1,5,7,10]。在传统的遗传算法中^[6],一般采用 $\{0,1\}^k$ 上的定长二进制串编码方式,一个解就是一个长度为k的二进制串(一个串就是一个染色体,一个二进制位为一个基因位)。在空间 $\{0,1\}^k$ 上主要的遗传算子有杂交和变异两种,杂交算子是主要的解空间搜索算子。标准的单点杂交算子,将两个给定的串(两个父本)随机地从任意同一位置分成前后两部分,再将两个串的后半部分相互交换,生成两个新的串(两个子代个体)。在应用问题中,一个染色体就是一个候选解(可行或不可行),一般将候选解的编码称为知识

表示。

2 知识表示空间与问题空间

本文将满足问题的可行解集合称为问题空间,记为 Ω 。将知识表示所定义的空间称为知识表示空间,记为 Ψ 。在求解具体问题时,一般将 Ω 中的点和 Ψ 中的点作一一映射。 Ω 与 Ψ 的映射关系一般有三种,如图1所示。图1a表示问题空间与知识表示空间正好一致, Ω 中的点与 Ψ 中的点一一对应。图1b所示的情况是,知识表示空间比问题空间要大,问题空间的所有点,在知识表示空间中都有相应的点与之对应,但知识表示空间的部分点,在问题空间不存在对应点。图1c表示问题空间比知识表示空间要大,知识表示空间不能完全地表示出问题空间的所有点,即,知识表示是不充分的。

当知识表示空间与问题解空间一致时,杂交算子不会产生不合法(不满足问题要求)的解。但当知识表示空间大于问题解空间时,则必须要考虑解的合法性问题,因为合法解通过遗传算子生成的新解,有可能是非法的。遗传算子会产生非法的解,其根本原因有,一是问题本身对合法解或可行解的某些约束,二是知识表示空间与问题空间的非一致性。

^{*} 本文得到了国家自然科学基金(No. 69903010)的资助。钟求喜 博士生,研究方向为进化计算、分布式任务分配与调度及Job-Shop调度等。谢 涛 博士后,研究领域有:智能控制、模式识别、过程辨识与建模、仿生计算等等。陈火旺 院士,博士生导师,研究领域为软件自动化、人工智能、机器翻译等。

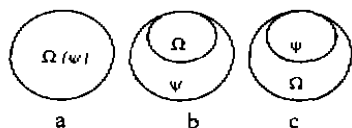


图1 问题空间与知识表示空间

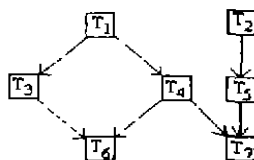


图2 7个子任务 DAG 图

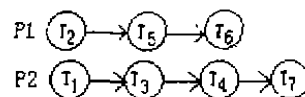


图3 分配调度

在任务分配与调度问题中,合适的知识表示,应能表达任务分配(将一个任务指派给某个处理机)和任务的调度(合理安排好各任务在所有处理机上的执行顺序)两方面的信息。将 n 个任务(或子任务)分配到 m 个处理机上并安排好它们的执行顺序,传统的二进制方式难以胜任,一般采用一维显式^[10]或隐式编码^[7]、二维显式编码^[11]和独立混合编码^[12]等基于排列的知识表示方式。由于二维显式编码能自然地表示出任务的分配和调度的先后关系,本文采用该知识表示方式。

在二维显式编码中,知识表示空间 $|\Psi|$ 要大于问题空间 $|\Omega|$ 。设大应用任务已经分解为 n 个子任务, $T = \{T_1, T_2, \dots, T_n\}$, 要分配到 m 个处理机上执行, $P = \{P_1, P_2, \dots, P_m\}$, 则一个分配与调度 S 可表示为, $S = (L_1, L_2, \dots, L_m)$, $L_i = (T_{i1}, T_{i2}, \dots, T_{ik})$ ($T_{ik} \in T, 1 \leq i \leq m$), L_i 为处理机 P_i 上要执行的任务列表。如果不考虑处理机排列的差别及任务间的约束关系,则知识表示空间大小 $|\Psi|$ 为:

$$|\Psi| = P_m! C_{n-1}^{m-1}$$

显然,满足子任务约束关系的分配调度数目 $|\Omega|$ 要远小于 $|\Psi|$ 。各子任务间的约束关系由依赖关系图给出,依赖关系图通常是有向无回路图(DAG)。如图2是7个子任务的依赖关系图,对于一个 DAG 图 G , 设 $G = (T, E, Et)$, 其中, T 为子任务集, 一个子任务 T_i 就是图 G 中的一个节点, E 是任务依赖关系图中的有向边集, 数组 $Et[1..n]$ 为子任务的估计运行时间, $Et[1]$ 为子任务 T_1 的估计运行时间, $(T_i, T_j) \in E$, 则表示在子任务 T_i 没有完成之前,任务 T_j 不能执行。这时称 T_i 为 T_j 的一个前继, T_j 为 T_i 的一个后继。不妨设 S 为某个 DAG 图的一个分配与调度(图3是图2中 DAG 在两个处理机上的一个分配与调度), $T_s(P_i)$ 表示在分配调度 S 下,当处理机 P_i 完成指派到本处理机上最后一个子任务所要花费的总时间,设 $T(S)$ 表示整个分配与调度 S 所要花费的总时间,简称调度 S 的完成时间,则有 $T(S) = \max(T_s(P_i), \forall 1 \leq i \leq m)$ 。目标就是: $\min(T(S) \forall S)$, 即为 DAG 图寻找一个分配调度 S , 使得 $T(S)$ 最小。

3 有序杂交算子

根据文[9],好的杂交算子应该有三个特点:适度大小的搜索子空间、启发式选择的杂交点和两个个体产生的多个个体在探索子空间要有代表性。在任务分配与调度问题中,更应关注的是杂交算子生成的解是否合法。文[1]给出了一种任务分配与调度的杂交算子,可以看成一种广义的杂交算子,共有 m 个杂交点。为了保证生成解的合法性,对杂交点的选择作了很强的限制,影响了算子的搜索能力。

本文提出的有序杂交算子 OCX,是从标准单点杂交算子演化而来,对杂交点的选择没有限制,象标准的单点杂交算子一样,随机地生成杂交点,从而产生出新的解。有序杂交算子的主要思想是,对两个给定的分配与调度 S_1 和 S_2 , 分别对其中的所有任务列表 L_i , 根据 DAG 图的拓扑排序,合并成一个任务列表,对得到的两个拓扑排序的任务列表 t_1 和 t_2 , 进行单点杂交,杂交后再作拓扑排序,就得到两个新的分配与调度 S_1' 和 S_2' 。有序杂交算子的操作过程如下:

1 拓扑排序:将分配调度 S_1 和 S_2 根据 DAG 图和任务在各自调度中的顺序分别作拓扑排序,得到 $t_1: (T_{11}, T_{12}, \dots, T_{1n})$ 和 $t_2: (T_{21}, T_{22}, \dots, T_{2n})$;

2 选择杂交点:随机地生成一个整数 p ($0 \leq p \leq n-1$) 作为杂交点;

3 杂交:对 t_1 , 保留前 p 个任务,然后将后面的 $(n-p)$ 个任务按 DAG 图和 t_2 的顺序重新作拓扑排序,生成新的两个拓扑排序。对 t_2 的处理与 t_1 相似;

4 转换:将新的拓扑排序转换成新的分配与调度 S_1' 和 S_2' 。

定理1 两个合法的分配与调度 S_1 和 S_2 , 经过 OCX 生成的新分配与调度 S_1' 和 S_2' 仍是合法的。

证明: 根据 OCX 的操作过程,显然,OCX 能保证每个子任务在分配调度 S_1' 和 S_2' 中出现的唯一性,即不会有子任务的丢失或重复出现。并且,OCX 生成的新调度,由于一直是基于 DAG 图的拓扑排序,所以不会破坏 DAG 图的约束关系,因 S_1' 和 S_2' 而是合法的。□

根据自然界中“物以类聚”，好点周围存在好点的自然现象，则一个好的分配与调度周围可能有更好的分配与调度。在 OCX 算子操作的第四步中，将一拓扑排序转换成分配与调度，这种转换一般是不唯一的。也正由于这种转换的不唯一性，使得分配与调度的生成有多种可能性。本文采用启发式方法，从多种可能的分配与调度中，以期获得较好的一个分配与调度。

任务分配与调度的最终目标是为了获得最短的完成时间。对于给定的 DAG 图而言，可以用关键路径，来给出完成时间的一个下界。即，所有任务的完成时间，不管如何分配与调度，是不可能比完成关键路径的所有任务时间更少。如果整个 DAG 图的任务要在关键路径给定的时间内完成，则对每个子任务而言，其最早可以执行的时刻和最迟执行的时刻，通过 DAG 图是可以计算出来的。设 T_{min} 为关键路径的完成时间， $T_{early}(T_i)$ 为子任务 T_i 的最早执行的时刻，有：

$$T_{early}(T_i) = \begin{cases} 0 & pred(T_i) = \Phi \\ \max(T_{early}(T_j) + Et[j]) & \text{否则} \end{cases}$$

其中 $T_j \in pred(T_i)$, $pred(T_i)$ 为 T_i 的前继节点集。设 $T_{late}(T_i)$ 为子任务 T_i 的最迟执行的时刻，则有：

$$T_{late}(T_i) = \begin{cases} T_{max} - Et[i] & succ(T_i) = \Phi \\ \min(T_{late}(T_j) - Et[i]) & \text{否则} \end{cases}$$

其中 $T_j \in succ(T_i)$, $succ(T_i)$ 为 T_i 的后继节点集。

对于给定的 DAG 图，为了获得最短的完成时间，当然希望每个任务能“马不停蹄”或“刻不容缓”一个接一个尽早地执行。通过 T_{early} 和 T_{late} 作指导，当要将一个拓扑排序结构转换成一个分配与调度时，对分配到同一处理机上的所有任务，可以对此处理机上任务列表按最早执行时刻 T_{early} 从小到大排序（“马不停蹄”），也按最早执行时刻 T_{early} 从大到小排序（“刻不容缓”）。这

样排序的分配与调度比单纯地将拓扑排序转换的分配与调度，就可能获得更短的完成时间。本文按 T_{early} 排序。

4 变异算子：迁移

遗传算法中的变异操作是一种辅助算子，在解群局部收敛时，通过变异算子的突变作用，来保持解群一定的多样性，在分配调度问题中，变异操作就是在一个调度内，随机交换子任务。显然，这种交换不能破坏任务间的约束关系。迁移算子是将一个子任务从一个处理机上迁移到另一个处理机上。对同一处理机上的多个串行执行的任务（根据 DAG 图，它们并没有相互的约束关系），如果存在迁移算子，当这多个任务被迁移到不同的处理机上时，就可能获得并行执行的性能，整个任务的完成时间就可能更短。仿真实验结果也表明，迁移算子较任务交换操作对任务分配与调度问题更有效^[8]。

表1 与文[1]算法相比较的仿真结果(单位时间)

处理机 个数	子任务 个数	完成时间		进化代数	
		算法[1]	本文算法	算法[1]	本文算法
2	20	224	209	121	108
	30	490	485	162	169
	50	660	603	101	149
3	20	179	141	129	90
	30	441	335	195	143
	50	541	452	227	175
4	20	164	141	65	73
	30	358	315	97	85
	50	461	369	335	165

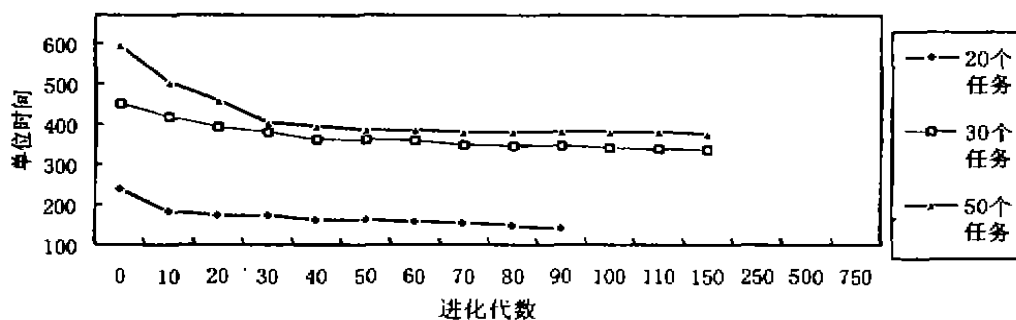


图4 算法的静态性能曲线($m=3, n=20, 30, 50$)

有序杂交算子中采用了启发式方法，由于这种启发式方法不总是最好的，因而本文在变异算子中，也为这种启发式方法提供一种突变作用。即，经过迁移操作

后，分配与调度内的所有任务列表不一定是按 T_{early} 排序的。因为按 T_{early} 排序的分配调度相对较好，但不总是最好的。

5 仿真结果

本文仿真实验是在一台 Pentium II 350/64M 的 PC 机进行的。DAG 图是随机生成的。算法的主要参数有:解的群体规模为10,用于模拟算法^[1]的杂交和变异概率分别为1.0和0.05,本文中迁移概率分别为0.3,算法的最多迭代代数数为1000代。算法中都采用了精英策略来保证最优解的非单调下降性。算法如果连续100代没找到更好的解,则认为算法基本收敛、算法终止。DAG 图是任意生成的,每个节点有1至4个后继,估计运行时间为1至50间的随机数,当算法能收敛时,运行时间通常在2秒钟以内,模拟结果见表1。表中的进化代数是五次运行的平均代数。在表1所列的数据中,有三种情况,本文算法找到了最优解。这三种情况是:20个任务在3个和4个处理机上的调度,30个任务在4个处理机上的调度。图4给出了基于3个处理机的分配与调度算法收敛的静态性能曲线图,列出了算法在不同进化代时所找到的最优解。

从表1中可以看出,本文算法比文[1]算法能找到更好的解,收敛速度在大多数情况下也较之有优势。这其中的主要原因有两点:一是启发式有序杂交算子,对杂交点的选择没有限制,更具通用性,同时又引入了领域知识的启发式方法,因而有更强的搜索能力;二是迁移算子在相当程度上更能挖掘系统中的并行性,减少了因同一处理机串行执行必须的等待时间。

结论 应用遗传算法求解任务分配与调度问题,首先要确定合适的知识表示。其次,在知识表示的基础上,当问题空间与知识表示空间不一致时,要解决遗传算子引起的非法解问题。适当地引入领域知识的启发式方法,对问题求解是有帮助的。仿真实验结果表明,本文提出的遗传算子,对任务分配与调度问题是比较

有效的。

进一步要研究的是,若考虑处理机处理能力的差异并且考虑数据传输的路由和数据量的大小,如何进行任务分配与调度,更现实的问题是,对于实用系统,系统中的任务到达是不可预测的动态过程,这样,对任务的分配与调度,也会是一个动态过程,静态调度是不行的。如何在同构或异构系统中,做到对任务的动态分配与调度,探索自适应的任务调度技术,已经成为一个十分重要的研究方向。

参考文献

1. Hou E S H, Ausani N. Genetic Algorithm for Multiprocessor Scheduling. IEEE trans. on parallel and distributed systems, 1994, 5(2): 113~120
2. Freund R F, Siegel H J. Heterogeneous Processing. IEEE Comput, 1993, 26(June): 13~17
3. Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc. 1989
4. Kasahara H, Narita S. Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing. IEEE trans. on Computers, 1984, C-33(11): 1023~1029
5. Ahmad I, Dhodhi M K. Multiprocessor Scheduling in a Genetic Paradigm. parallel computing, 1996, 22: 395~406
6. Holland J H. Adaptation in Natural and Artificial Systems. Univ. of Michigan Press, 1975
7. Wang L, et al. Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach. Jour. Of Parallel and Distributed Computing, 1997, 47(1): 8~23
8. Zhong Quixi, Chen Huowang. Generic Operators in Multiprocessor Scheduling. In: The Fourth Interl. Symposium On Future Software Technology, Nanjin, Oct. 1999
9. Wu Shao-Yan, Xu Zhou-Qun. A Heuristic Policy for Constructing Crossover in Genetic Algorithms. Chinese J. Computers, 1998, 21(11): 1003~1008
10. Kwok Yu-Kwong, Ahmad Ishfaq. Efficient Scheduling of Arbitrary Task Graphs to Multiprocessors Using a Parallel Genetic Algorithm. J. Parallel and Distributed Computing, 1997, 47(1): 58~77

(上接第57页)

此外,在利用纹理特征的同时,如能进一步将颜色、形状等特征及小波变换等方法结合起来,必将取得更好的检索结果。

参考文献

1. 白雪生,徐光佑.基于内容的图像检索及其相关技术的研究.机器人,1997,19(3):231~240
2. 徐晖,廖孟扬.医学图像数据库中基于图像内容查询的研究.计算机工程与应用,1991(Jan):52~54
3. 李国辉,柳伟,曹莉华.一种基于颜色特征的图像检索方法.中国图像图形学报,1999,4A(3):248~251
4. Manjunath B S, Ma W Y. Texture Features for Browsing

and Retrieval of Image Data. IEEE PAMI, 1996, 18(8): 837~841

5. Cascia M J, et al. Combining Texture and Visual Cues for Content-based Image Retrieval on the World Wide Web. IEEE Workshop on Content-based Access of Image and Video Libraries, June 2nd, 1998. Kenneth R. Castleman, Digital Image Processing, Prentice Hall, 1998
6. 余英林.数字图像处理与模式识别.华南理工大学出版社,广州,1994.150~200
7. 王新成.多媒体实用技术(图像分册).电子科技大学出版社,1995.217~300
8. 余英林,李海洲.神经网络与信号分析.华南理工大学出版社,1996.10~100