公式：$R_i=|\{t_i|t_i\in\|C_i\|\wedge satisfy(t_i,W_i)\wedge relation(t,t_i,P_i)\wedge R_{i+1}\}|\leqslant n$

其含义为："至多 n"。

· Q_i=Just n

公式：$R_i=|\{t_i|t_i\in\|C_i\|\wedge satisfy(t_i,W_i)\wedge relation(t,t_i,P_i)\wedge R_{i+1}\}|=n$

其含义为："恰好 n"。

· $Q_i=n_1$ To n_2

公式：$R_i=n_1\leqslant|\{t_i|t_i\in\|C_i\|\wedge satisfy(t_i,W_i)\wedge relation(t,t_i,P_i)\wedge R_{i+1}\}|\leqslant n_2$

其含义为："n_1至 n_2个"。

谓词 satisfy(t,W)表示：t 满足选择条件 W。当 W=ε 时，satisfy(t,W)取值为真。

3.2.3 转换实例 本节给出以谓词公式描述的复杂选择条件的形式化的语义。

转换规则：将选择条件中每一个由 Where 引导的选择条件及其所限定的元组对象变量名 t 转化成谓词公式 satisfy(t,W)。而 satisfy(t,W)的获得，可通过将选择条件 W 根据3.2.2节所述的转化方法以及将 W 中的 And，Or，Not 分别变换为 $\wedge$，$\vee$，$\neg$而获得。对整个选择条件来说，转化的结果是嵌套形式的谓词公式。

例 将3.1节中的例子的选择条件转化为如下嵌套形式的谓词公式：

satisfy(k_1,W_1)=(|{k_2|($k_2\in\|ITEM\|$)(satisfy(k_2,W_2)$\wedge$relation(k_1,k_2,P_2)$\wedge$($\forall$ $k_3\in\|SALE\|$)

(satisfy(k_3,W_3)$\wedge$relation(k_1,k_3,P_3)$\rightarrow$(k_2.price$\times k_3$.volume>10000))}|$\geqslant$5)

satisfy(k_2,W_2)=(k_2.description.color="红")

satisfy(k_3,W_3)=($k_3\in k_2$.is_sold)

其中：

· W_1 是：(At least 5 k_2 Of ITEM Through TRADFR.sells，SALE，ITEM.is_sold Where k_2.description.color ="红") (Each k_3 Of SALE Through TRADER.sells Where $k_3\in K_2$.is_sold}) (k_2.price$\times k_3$.volume>10000)

· W_2是：k_2.description.color="红"

· W_3是：$k_3\in K_2$.is_sold

· P_2是：TRADER.sells，SALE，ITEM.is_sold

· P_3是：TRADER.sells

结束语 在 OODB 的选择查询中，查询的表达能力即选择条件语义的表达能力至关重要。从元组对象和集合对象的复合关系出发，我们以文法的形式给出了选择查询中不同层次上的对象间的量的约束关系的表达方法；描述了将此复杂选择操作转化成对象谓词演算的方法。由于一条选择操作语句能表达一类基于路径和复杂量词限定的复杂查询语义，因此，本文提出的表达方法具有较强的非过程性。

主要参考文献

[1] The Object Database Standard：ODMG-93，edited by R.G.G.Cattell，Morgan Kaufmann Publishers

[2] V.D.Bussche and A.Heuer，Using SQL with Object-Oriented Database.Info.Sys.，18(7)1993

[3] V.D.Bussche，Evaluation and Optimization of complex object selection，Lecture Notes in Computer Science，No.566

[4] M.Roth et al.，Extended algebra and calculus for nested relational databases，ACM Trans.Database Sys.，13(4)1988

[5] N.Bhalla and S.Balasundaram，Operation and Queries in Object-Oriented Database Supporting Complex Objects，Information and Software Technology，35(1)1993

[6] 钟武、胡守仁，OODB 数据模型及查询代数，计算机科学，24(2)1997

(上接第88页)

[2] Stephanie Forrest et al.，Genetic Algorithm：Principle of Natural Selection Applied to Computation，Science，1993

[3] V.Kreinovich et al.，Genetic Algorithm：What Fitness Scaling is Optimal?，Cybernetics and Systems，24(1)1991

[4] X.Yao，A review of evolutionary artificial neual network，Intel.J.of Intell.Systems，1993，8

[5] Philipp Schmid，The Mapping Problem：A Neural Network Approach，Intel.Neural Network Conference，1990，Paris，France

[6] Hwang，K.and Ghoshu，J.，Critical Issues in Mapping Neural Networks on Message-Passing Multicomputers，Intel.Symposium on Computer Artichitecture，ACM/IEEE 1988

计算机科学1998 Vol. 25№. 2

神经网络并行计算的一种有效分配算法*)

An Efficient Mapping Algorithm for Neural Parallel Computing

罗 莉 胡守仁

(国防科技大学计算机系 长沙410073)

摘 要 Neural network mapping schedule is an important research content in the neural network implementation, The optimum schedule is very meaningful for the performance of neurocomputer. In this paper transformation of mapping schedule to general code of genetic algorithm is given. In the end, the simulated results are shown.

关键词 Neural network mapping, Genetic algorithm, Fitness value

高性能计算的并行神经计算机是目前人工神经网络硬件实现的主流。并行神经机一般采用P个处理结点,完成N个神经元组成的神经网络模拟(P<N),如何将神经网络并行计算任务进行划分,寻找并行子任务与处理结点之间的最佳映射,获得神经计算的最佳性能,这是并行神经机系统的一个关键问题。本文中,将通过对神经网络计算特点、处理结点速度和通讯性能对映射影响的分析,给出一个有效的分配遗传算法。

一、神经计算映射模型

1.1 神经计算机系统的模型

并行神经机就是根据神经网络及其计算特点,利用传统计算机技术和并行处理技术以一定结构连接而成的并行计算环境,一般是同构型计算环境,它可以被定义为图 $HN(P,L)$,其中:$P=\{p_1,p_2,\cdots,p_M\}$,表示M个处理结点;$L\subseteq P\times P$ 表示各处理结点的连接。

1.2 神经计算分配映射模型

神经计算是串、并行相结合的一种计算。并行意味着某一时刻内可能有许多神经元的计算可同时进行,串行即表示某些神经元计算之间存在着顺序相关。

神经计算一般从神经网络计算级、步骤级、神经元级、微任务级四个级别开发并行性,我们认为从神经元一级开发神经计算的并行子任务直观,也更能体会神经计算潜在的并行性。

神经计算可以定义为图 $TG(O,C)$,其中 $O=\{o_1,o_2,\cdots,o_N\}$,表示N个的神经元;$C\subseteq O\times O$ 表示各个神经元之间的连接关系。因此,神经网络的映射可以定义为:对于神经计算任务图TG,寻找一个神经元 o_i 到处理结点 p_j 的映射关系,使神经计算在该环境中获得最佳的并行计算性能。

1.3 映射任务的分析

最佳分配与映射的目标在于:能在最短的时间获得问题的解。任务完成需等待所有子任务的完成,因而要求 $f=\max f_j(j=1,2,\cdots,M)$ 为最小,其中 f_j 为处理结点 P_j 所分配的并行任务的执行时间。

定义1 设神经元i计算代码量为 q_i。

定义2 处理结点执行指令的速度为E。

定义3 神经元i和神经元k的数据通讯量为 $com(i,k)$,$1<i,k<N$,N为神经元的个数。

定义4 任意两处理结点的通讯速度为 $cs(j,l)$,$(j,l\in M)$。

定义5 定义分配矩阵A

$$A_{i,j}=\begin{cases}1, & \text{neuron } o_i \text{ 被分配在处理机 } p_j \text{ 上}\\ 0, & \text{neuron } o_i \text{ 未被分配在处理机 } p_j \text{ 上}\end{cases}$$

这样,我们即可得到各个处理结点在神经网络任务映射之后,其任务执行的时间开销:

$$f_j=\Big[\sum_{i=1}^{N}q_i\times A_{i,j}\Big]/E+\sum_{i=1}^{N}\sum_{k=1,k\neq i}^{N}\sum_{l=1,l\neq j}^{M}[(1-A_{i,j})\times com(i,k)\times(1-A_{k,l})]/cs(j,l)$$

令 $C=1/E$,$D(j,l)=1/cs(j,l)$,C,$D(j,l)$ 与神经计算机系统硬件有关,C,$D(j,l)$ 数值越小,系统性能

*)本文得到国家"863"高技术计划的资助,课题代号863-306-06-1。

越高，则处理机分配即成为：

$$f=\max f_i=\max\Big\{\sum_{i=1}^{N}C\times q_i\times A_{i,j}+\sum_{i=1}^{N}\sum_{k=1,k\neq i}^{N}\sum_{l=1,l\neq j}^{M}(1-A_{i,j})\times D(j,l)com(i,k)(1-A_{k,l})\Big\}\quad(1)$$

处理机分配即要求 f 获得最小值。

上述的处理机分配模型是一个 NP 完备性问题，为有效找出其解，我们提出了基于遗传算法的映射分配。遗传算法近几年得到国际学术界普遍重视，因其算法简单，在组合优化问题和复杂的函数优化问题上具有很强的计算能力。

二、神经并行计算映射算法

2.1 问题的表示及算法

在遗传算法中，问题的解用数字串（基因链码或称为个体）来表示，遗传算子是对串直接进行操作，这样，如何用适当的串表示问题的解是遗传算法的首要问题。我们把映射分配问题的解表示成基因链码，如此定义：

$$\begin{matrix}o_{1,1} & o_{1,2} & \cdots & o_{1,k_1}\\ o_{2,1} & o_{2,2} & \cdots & o_{2,k_2}\\ \vdots & \vdots & \cdots & \vdots\\ o_{M,1} & o_{M,2} & \cdots & o_{M,kM}\end{matrix}$$

其中 M 表示处理单元 PE 的数目，k_i 表示第 i 台 PE 上处理的神经元数目，$o_{i,j}$表示在第 i 台 PE 上处理的某神经元，用一个整数表示，$o_{i,j}\in\{1,2,\cdots,N\}$，N 是神经元的个数。此处基因链码不是通常使用的一条链码表示，而是 M 个子段（每段对应一个 PE）构成一个大的基因链码，这个链码的第 i 段子链码就表示处理器 PE_i 的处理顺序。

由于对映射调度的特殊表示，因此做交叉与变异的方法与一般方法有所不同。假定选择两个链码做双亲，在交叉时产生一个随机数来选择哪台处理器 PE_j，再随机产生一个交叉位置，没有选中的处理器处理顺序不变，即 M-1段子链码不变，对选中的第 j 段子链码，第一个后代 x'_1 交叉点的前一小段链码不变，后一小段从 x_1 中剔掉，然后把前一小段的基因从 x_2 中剔掉，x_2 中剩下的基因保持顺序不变，构成 x'_1 的后一小段链码。同理产生第2个后代的第 j 段子链码。如 x_1，x_2 的第 j 段子链码分别为：

x_1：① ② ⑦ | ④ ⑤ ③ ⑥
x_2：① ② ⑥ | ⑦ ④ ⑤ ③
交叉点

交叉结束后：
x'_1：① ② ⑦ ⑥ ④ ⑤ ③
x'_2：① ② ⑥ ⑦ ④ ⑤ ③

在做变异时，不能简单地将某位基因的值做改变，由于问题的特殊性，这样简单的改变有可能无实际意义。如第 j 段子链码为①②③④⑤⑥⑦，把第5位变异为⑦就变为：①②③④⑦⑥⑦。在实际问题中这是不允许的。我们采用变异方法，随机产生变异位置，将变异点的前后基因互变位置，上例的变异为：

变异前 ① ② ③ ④ ⑤ | ⑥ ⑦
变异后 ① ② ③ ④ ⑥ | ⑤ ⑦
变异点

适应度的定义：一个最佳的神经网络映射分配，处理机分配即要求任务执行时间开销获得最小值，公式(1)定义为适应度。

下面给出遗传算法设计神经网络映射分配的步骤：

(1)设定神经网络和处理结点的有关信息（神经元数目、处理结点数目），随机选择第0代群体的所有基因链码；

(2)根据适应度计算每条链码的适应度；

(3)让群体中的基因链码进行自然选择，选择优质双亲交叉和变异运算以产生下一代；

(4)重复(2)和(3)，直至算法收敛或达到预先给定的世代数。

我们设置世代数为100，群体中链码数目为40，0.2的变异率，0.2的交叉率。使用上述算法，有时得到的链码并不是问题的解，我们称为死锁调度。对于二层神经网络结构（如图1）。

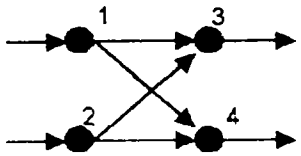

图1 二层前馈神经网络模型

网络模型： 前馈网络
神经元个数： 4
处理单元个数：2
层数： 2
第一层神经元：1 2
第二层神经元：3 4
互连结构：同层神经元不互连，不同层互连
处理结点通讯方式：广播式
分配结果
PE1： 3 1
PE2： 4 2

上述得到的基因链码不是问题的解，出现了死锁调度。死锁调度可以如此定义，将并行任务按神经计算执行时间顺序加上时间步数标记，如基因链码第 i 段子链码 $o_{i,1}^{t1}o_{i,2}^{t2}\cdots o_{i,k_2}^{tn}$，若 t1＞t2，则为死锁调度。

为解决这个问题，我们又引入一个链码学习过

程。对已给的调度方案，每段子链码按时间顺序重新排序，如解性能好，则保留；否则迭代。迭代次数为(Km－Tm)，算法结束。Km 为链码长度，Tm 为神经计算总时间步数。采用这个算法作为个体链码学习过程，因此链码能在这一过程中提高对环境的适应度。改进的算法在前述算法第二步加入链码学习过程。

2.2 算法分析

在上述并行神经计算映射算法中，适应度定义为 $f=\max f_j(j=1,2,\cdots,M)$，要求执行时间短，则要力求系统均衡；通讯常数 $cs(j,l)$ 的引入，在神经计算机不同的通讯方式，神经网络不同拓扑结构造成的通讯开销差异，算法利用这一信息，自动将通讯开销减少。但负载均衡和通讯开销并不是同步递减关系，常常系统均衡减小的同时，通讯开销有可能增大。由此我们的算法在负载不均和通讯开销之间，寻求一种最佳组合，获得高效的并行速度。

另一方面，适应度 $f=\max f_j(j=1,2,\cdots,M)$，还将该算法在完成并行神经计算的最佳映射的同时，得出该任务在多机环境下具有最佳并行计算性能，所使用的处理结点的个数。神经计算映射多个处理结点，执行时间开销减少，通讯开销增大，结点的增多不一定得到性能的提高。处理结点的个数也是一个优化问题。

三、实例示范

我们已利用该算法对多种神经网络模型进行映射分配，实验结果证明该算法是通用有效的，这里给出验证实例及验证结果。

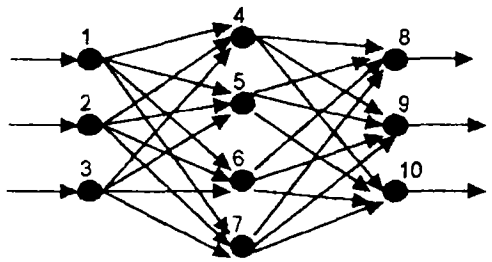

图2 三层前馈神经网络模型分配

网络模型(如图2)：前馈网络
神经元个数： 10
层数： 3
第一层神经元： 1 2 3
第二层神经元： 4 5 6 7
第三层神经元： 8 9 10
互连结构： 同层神经元不互连，不同层互连
(设各处理结点的处理速度、通讯速度一致)

(1)分配在三台处理结点

PE1： 1 4 7 8
PE2： 2 5 9
PE3： 3 6 10
广播通讯方式下 $f=4C+3D_{广播}$；
点到点通讯方式下 $f=4C+6D_{点播}$；

(2)分配在四台处理结点

PE1： 1 5 8
PE2： 2 6 9
PE3： 3 7 10
PE4： 4
广播通讯方式下 $f=3C+2D_{广播}$；
点到点通讯方式下 $f=3C+5D_{点播}$；

(3)分配在5台处理结点

PE1： 1 6
PE2： 2 7
PE3： 3 8
PE4： 4 9
PE5： 5 10
广播通讯方式下 $f=3C+2D_{广播}$；
点到点通讯方式下 $f=3C+5D_{点播}$；

从上述分配结果可看出，通讯方式不论采用点到点或广播，该模型选用四台处理结点执行时间开销较小，性能价格比好。

我们利用该算法在一些典型实例中其结果如下：

* 26个大写英文字母识别。网络模型是三层前馈网络，输入为16×16点阵，输入层神经元为256个，隐含层神经元为60，输出层神经元为26。互连结构是同层神经元不互连，不同层互连。

分配任务在4台处理结点
广播通讯方式下 $f=86C+79D_{广播}$；
点到点通讯方式下 $f=86C+237D_{点播}$；

分配任务在5台处理结点
广播通讯方式下 $f=70C+64D_{广播}$；
点到点通讯方式下 $f=70C+252D_{点播}$；

* TSP 问题。求解10个城市的最短路径，网络模型是 Hopfield 网络，100个神经元全互联。

分配任务在4台处理结点
广播通讯方式下 $f=25C+25D_{广播}$；
点到点通讯方式下 $f=25C+75D_{点播}$；

分配任务在5台处理结点：
广播通讯方式下 $f=20C+20D_{广播}$；
点到点通讯方式下 $f=20C+80D_{点播}$；

从上述实例结果我们得到，多机系统进行神经计算仿真，广播通讯减少了系统的开销，是种较好的方式；点到点的通讯方式随问题规模的扩大，通讯开销增大，处理结点数目的增多，总通讯开销也增大；处理结点数目增多，并不一定带来性能的提高。

参考文献

[1] 韩桢祥、文福拴，模拟进化优化方法及其应用，计算机科学，1995，2

(下转第85页)