

多核集群系统下的混合并行遗传算法研究

王竹荣 巨涛 马凡

(西安理工大学计算机科学与工程学院 西安 710048)

摘要 为应对传统遗传算法在处理大规模组合优化问题面临的进化速度缓慢,难以达到实时要求的严峻挑战,提出了一种在多核 PC 集群系统上实现“粗粒度-主从式”混合并行遗传算法的模型;通过把“粗粒度-主从式”并行遗传算法映射到多核 PC 集群上,结合消息传递和共享存储两种并行编程模型,在节点间使用消息传递模型(MPI),对应的遗传算法为粗粒度并行遗传算法,在节点内使用共享存储模型(OpenMP),对应的遗传算法为主从式并行遗传算法,用 MPI 和 OpenMP 混合编程的方式以进程和线程两级并行在多核集群上实现具体的混合并行遗传算法。理论分析和实验结果表明,提出的实现模型有较好的性能,可大大改进传统遗传算法的缺陷。为利用并行遗传算法在普通多核 PC 集群上处理大规模组合优化问题提出了一种有效、可行的解决方案。

关键词 混合并行遗传算法,多核集群系统,OpenMP,MPI

中图法分类号 TP18 **文献标识码** A

Research of Hybrid Parallel Genetic Algorithm Based on Multi-core Cluster System

WANG Zhu-rong JU Tao MA Fan

(School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China)

Abstract In response to the challenge of the traditional genetic algorithms facing the slow pace of the evolution and difficulties of unable to meet real-time requirements in handling large-scale combinatorial optimization problems, this paper proposed the coarse-grained-master-slave hybrid parallel genetic algorithm (HPGA) model run on multi-core cluster system. Through integrating the features of the message-passing model and the shared-memory model, we used message-passing model—MPI among nodes to correspond coarse-grained PGA, used share-memory model—OpenMP within one node to correspond master-slave PGA and thus combined the higher parallel computing ability of multi-core cluster system with inherent parallelism of PGA. Realized the HPGA based on two-layer parallel both process and thread by using hybrid parallel programming models combing MPI and OpenMP on multi-core cluster system. Theoretical analysis and experimental results show that the HPGA model of this paper has high performance and overcomes traditional GA's defects. It put forward an effective and viable solution for using Parallel Genetic Algorithm based on simple multi-core PC cluster to deal with the large-scale combinatorial optimization problems.

Keywords Hybrid PGA, Multi-core cluster system, OpenMP, MPI

1 引言

如何在解空间中找到组合优化和函数优化问题的满意解和最优解一直是各学科领域普遍关心的难题。遗传算法是一种全局优化搜索算法,它不依赖于所求问题的相关领域,对问题的种类有很强的鲁棒性,所以被广泛地应用在很多学科领域。随着科学技术的迅速发展,优化问题的规模变得越来越大,搜索空间的复杂程度也越来越高,对传统遗传算法的求解速度和求解质量也提出了更高的要求。利用传统遗传算法(SGA)求解以上问题时,由于需要较多的个体数量和大量的计算,进化速度缓慢难以达到适时的要求,故传统遗传算法的这种串行处理方式就显得“力不从心”^[1,2]。遗传算法又具有天然的并行性,非常适合在大规模并行机上实现,如果能把遗传算法的这种固有的并行性和并行机的高速并行计算性能很

好地结合起来,实现遗传算法的并行化,就可以大大地改善传统遗传算法的缺陷,最大限度地提高遗传算法的求解质量,加快遗传算法的收敛速度。

近年来并行遗传算法(Parallel Genetic Algorithms, PGA)的研究出现热潮,许多学者致力于设计各种并行化策略,把并行遗传算法应用到各种并行计算机上。

由于传统并行机价格昂贵,一般用户难以承担,这无疑阻碍了普通用户用并行遗传算法解决大规模组合优化问题的步伐,也不能充分发挥 PGA 解决大规模搜索问题的优势。随着技术的成熟与价格的下降,多核 CPU 逐渐普及,而用高性能多核 PC 组建的集群系统具有投资风险小、结构灵活、可扩展性强、易于实现的特点,有很高的性价比^[3],可以容易地利用较低的成本获得传统并行机的高计算性能。构建以多核 PC 为单计算节点的集群系统已成为并行计算机体系结构的一个

到稿日期:2010-08-31 返修日期:2010-11-04 本文受国家自然科学基金(60873035),陕西省教育厅自然科学专项研究基金(2010JK713)资助。

王竹荣 男,博士,副教授,CCF 会员,主要研究方向为智能计算、智能优化系统等,E-mail: wangzhurong@xaut.edu.cn;巨涛 男,硕士生,主要研究方向为智能计算、并行计算;马凡 女,硕士生,主要研究方向为智能计算。

发展趋势,也是一种有效适用的并行计算机系统的实现方法。

如何将并行遗传算法和现有并行计算机的体系结构很好地结合起来,设计有效的并行遗传算法及相应的实现系统,对遗传算法的理论研究和应用研究都有积极的意义,所以并行遗传算法实现模型的研究已经成为并行遗传算法的一个重要研究方向。

2 相关研究

并行遗传算法的研究从遗传算法研究之初就开始了,相关理论方面的研究已经非常成熟。主要有以下方面的研究:对种群规模、编码方式、参数设置等对 PGA 效率影响方面的研究^[3,4];对 PGA 模型的研究,通过建立并行遗传算法的实现模型,来分析遗传算法的优化过程,研究迁移拓扑、迁移策略及其参数设置方法,以在实践中指导并行操作^[5,6];对混合并行遗传算法的研究主要有两个方面:一方面在现有的 PGA 的基础上如何将遗传算法和其它局部优化算法结合起来,以提高求解质量^[7];另一方面是并行模型混合化,即将基本的并行遗传算法模型结合起来,构成一种新的混合模型^[8]。还有一个主要的研究方面就是对并行遗传算法应用的研究,目前 PGA 已成功地应用到人工神经网络、机器学习、工程计算等许多复杂的组合优化问题当中^[9-12]。

以上研究主要是在理论方面对 PGA 的内在特性进行研究,要充分发挥 PGA 的并行性必须要和具体的并行化策略结合起来,即把并行遗传算法和并行编程技术结合起来,从根本上改变以往的串行方式模拟并行遗传算法的并发实现方式,实现真正的并行。近年来很多学者在多核处理器及集群系统的并行程序设计方面做了大量的工作。文献^[13]对多核节点集群系统上的混合编程模型——MPI+OpenMP 进行了深入的研究,将两种并行编程模型优点结合起来以获得更好的计算性能。文献^[9]在 MPI 的基础上实现了主从式并行遗传算法框架,但主从式并行遗传算法由于本身通信模式的限制不能充分发挥集群各节点的独立运算性能,不能很好地发挥并行遗传算法的内在并行性和集群的高速并行计算性能,对大规模复杂的组合优化问题不能较好地处理。

为了充分发挥多核 PC 的高速并行计算性能和混合并行遗传算法的内在并行性,本文提出多核集群下的混合并行遗传算法实现模型,将当前多核 PC 集群的物理拓扑结构和“粗粒度-主从式”混合并行遗传算法逻辑结构完美结合,通过 MPI 和 OpenMP 混合编程完成进程和线程的两级并行,实现节点间的粗粒度和节点内多核间的主从式混合并行遗传算法,从根本上提高了并行遗传算法的求解质量和收敛速度。在一定程度上降低了并行遗传算法实现难度,扩大了 PGA 的应用范围,充分发挥了 PGA 在解决大规模搜索问题上的优势,为普通用户在较低成本的基础上应用并行遗传算法解决复杂的组合优化和函数优化问题提出了一种有效的解决方案。

3 多核集群下的混合并行遗传算法模型的设计与实现

3.1 并行遗传算法模型分析

并行遗传算法是将并行计算机的高速并行性和遗传算法的内在并行性结合起来,加快遗传算法的搜索过程,保持和丰富种群的多样性,减少早熟现象发生,从而有效且高速地完成

对复杂问题的搜索。PGA 的基本思想就是通过多种群并行进化和引入迁移算子进行种群间信息的交换,对传统遗传算法进行并行化。将种群的进化过程划分到并行计算机的不同计算节点上进行分布式进化,并通过迁移算子利用种群间信息交换策略实现优良基因的交换,使“优良”的遗传信息在各个子种群间快速传播,充分融合,使得算法跨越局部收敛的障碍,向全局最优方向进化,从而加快遗传算法的收敛速度,提高求解质量。

将种群划分到不同的计算节点可以采用不同的方法和途径,从而出现了不同类型的并行遗传算法模型。目前并行遗传算法主要有以下 4 种模型:主从式并行遗传算法(Master-slave PGA, MPGA),粗粒度并行遗传算法(Coarse-grained PGA, CPGA),细粒度并行遗传算法(Fine-grained PGA, FPGGA)和混合并行遗传算法(Hybrid PGA, HPGA)^[1]。其中混合并行遗传算法是结合了前 3 种基本并行遗传算法模型而出现的一种多层并行模型,当前混合模型主要有 3 种:粗粒度-细粒度模型。粗粒度-粗粒度模型,粗粒度-主从式模型^[2]。通常混合并行遗传算法通过层次结构进行组织,最上层常用粗粒度模型,低层可以用粗粒度模型、细粒度模型和主从式模型任何一种。实际中应用较多的是粗粒度-主从式模型。

3.2 多核集群下的混合并行遗传算法模型的设计

以往混合并行遗传算法的实现要么在单机上以串行方式模拟,要么就在分布式系统上并行实现,但由于通信延迟,负载均衡、同步等不好控制,实时性、可扩展性差等因素,因此不能完全发挥并行遗传算法的固有并行性。也有在并行机上直接实现的,但计算成本高。而多核 PC 集群系统的物理结构正好和“粗粒度-主从式”混合并行遗传算法逻辑结构相吻合,所以可以很好地将“粗粒度-主从式”混合并行遗传算法映射到多核 PC 集群上。具体拓扑结构如图 1、图 2 所示。

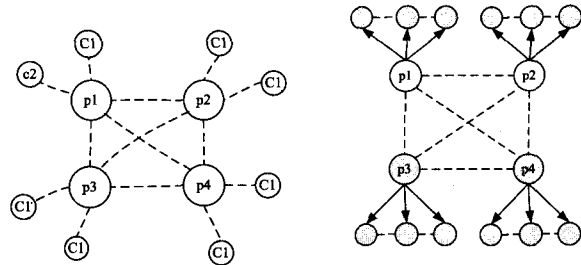


图 1 多核集群物理拓扑结构

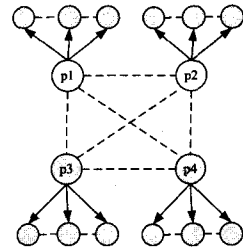


图 2 粗粒度-主从式混合并行遗传算法模型

图 1 中 p_1, p_2, p_3, p_4 分别代表集群中 4 个多核计算机节点, c_1, c_2 分别代表每个多核计算节点内的各处理核(文中提到的集群由 4 个双核 pc 节点组成)。图 2 中的 p_1, p_2, p_3, p_4 分别代表混合并行遗传算法结构中的上层各个子种群,而每个子种群内部又分为若干个小种群,子种群内部按主从式并行遗传算法进化,各个子种群之间按粗粒度并行遗传算法进化。分析这两种结构的特点后,可以将图 2 中混合并行遗传算法结构中的上层子种群粗粒度进化映射到多核 PC 集群的各个计算节点上并行执行,而将混合并行遗传算法结构中的子种群内的主从式进化映射到多核集群每个计算节点内的多个处理核上并行执行。从而实现计算节点间和节点内核间的两级并行。

结合了多核 PC 集群系统就可以充分发挥“粗粒度-主从式”混合并行遗传算法的并行性,实现种群之间和种群内部的两级并行;可以进一步加快算法的收敛速度,提高算法的求解

质量。

3.3 多核集群下的混合并行遗传算法模型实现

多核集群下的混合并行遗传算法模型的实现分为物理层和逻辑层两个层次。物理层指具体的并行编程模型实现,逻辑层主要是指混合并行遗传算法的实现。

(1) 物理层实现

物理层实现时为了充分利用多核集群层次结构的特点,将消息传递和共享存储两种并行编程模型相结合,以实现 MPI 和 OpenMP 的混合编程。整个物理层次分为两级结构,上层结构表示节点间的并行,下层结构表示节点内的并行。多核集群中的每个节点对应一个进程,而节点内部按处理核数目分成若干线程,从而实现了进程和线程的两级并行。

进程级并行的实现:采用任务分解的方式将问题分成通信不密集的几个部分,每个部分分配到一个多核节点机(即一个进程)上进行处理,各节点之间按消息传递 MPI 模型并行执行,每隔一定的时间间隔或满足一定的条件各节点间就要交换信息,以使整个任务有效、合理、统一的得到处理。具体的消息传递模型采用 MPI 的 MPICH2 实现。各节点间的通信主要是通过调用 MPI 的基本函数完成。

线程级并行的实现:完成了进程级任务分解后,对每个节点上的任务采用数据分解的方式再次分解,并分配到多核处理器上的不同计算核心上,利用多核编程技术生成多个线程并行执行,节点内的并行编程采用共享存储式的 OpenMP 并行编程模型实现。具体实现时主要通过调用 OpenMP 提供的编译指导语句和运行时库函数对节点内部循环部分进行并行化,并行化之前首先要保证并行执行的部分没有数据相关性。每个进程内只有循环计算的部分是多线程并行求解,而在其它代码段,则只有单个 MPI 进程求解,每个线程基于并行域内的线程数和本身的线程号决定在进程中对哪个部分进行操作。

将两种并行编程模型结合的主要目的是为了充分利用整个多核集群系统的资源,尽量保证负载均衡,实现通信和计算的重叠,从而提高并行程序的性能。若采用单一的 MPI 并行方式,由于它不能同时进行通信和计算,这样对应到多核集群各个计算节点上,则会出现节点内处理核忙闲不均的现象,浪费了计算资源。用混合并行编程模型后,在每个 MPI 进程内部用 OpenMP 并行编程方式生成多个轻量级的线程,这样可选择主线程或指定一个线程进行通信,其它线程执行计算部分,再利用 OpenMP 负载均衡策略尽量保证系统中的各个处理核负载均衡,这样就可将多核处理器每个核心都有效利用起来。同时采用混合并行编程模型可以大大减少通信进程数量,使 MPI 能有效处理多核处理器间的粗粒度通信,而 OpenMP 通过共享存储空间可快速地共享数据,缩短任务的响应时间,较好地解决每个多核处理器节点内部处理核间的交互,在很大程度上降低通信开销,从而可高效地实现进程和线程的两级并行,获得较大的性能提升。

(2) 逻辑层实现

逻辑层对应具体的混合并行遗传算法的实现,本文采用的是“粗粒度-主从式”混合并行遗传算法模型。它包含上下两层,上层由粗粒度并行遗传算法模型构成,该层可以映射到上面物理层的各个多核节点机上分别独立地进行遗传操作;下层由主从式并行遗传算法模型构成,该层可以映射到物理层各个节点机内部,实现具体的遗传操作。

并行遗传算法实现时,首先将整个种群按多核集群节点数目分成相应数量的子种群,然后将这些子种群分配到每一个多核计算节点上。每一个多核节点独立地对一个种群进行遗传操作,经过一定的进化间隔,各个种群之间通过一定的策略进行种群间个体的迁移,以交换种群信息,保持种群的多样性。从而实现遗传算法种群间的粗粒度并行。

多核计算节点内部按主从式并行遗传算法模式进行处理,实现节点内遗传算法的主从式进化,即利用多核编程技术,生成多个并行执行的线程,个体适应度的计算和遗传操作可以分别在不同的线程上并行执行。其中主线程负责主要的遗传操作,而其它线程并行的进行适应度的计算。主线程和其它线程之间的信息交换主要有两方面:一方面主线程按一定的机制将任务分发给其它线程,另一方面其它线程将各自并行计算的结果传送给主线程。

4 混合并行遗传算法的实现

4.1 迁移算子的设计

迁移算子是并行遗传算法的重要组成部分。在并行遗传算法中各种群经过一定的进化间隔后要按一定的策略进行种群间个体的迁移,以交换种群信息,保持种群的多样性。具体的迁移策略即迁移算子的设计是关系到能否有效降低通信代价的关键。

本文迁移算子中引入了“种群池”的概念,打破了传统各种群之间直接通信的模式,大大减少了各种群之间的频繁通信,有效地降低了系统的通信代价。“种群池”的概念及具体实现如下。

“种群池”是一种单独运行在主节点上的数据结构(本文中用结构体数组实现),作为各个计算节点,即各个种群间相互通信的桥梁和纽带,每个子种群可以将优秀个体发送到“种群池”,同时也可以接收“种群池”中来自其它子种群中的优秀“迁移个体”。本文中具体迁移实现过程为:

各个子种群经过一定的进化间隔,将本群体中当前的最优个体传送到主节点中的“种群池”中,主节点在“种群池”中收集了所有子种群发送的优秀个体后,经比较将“种群池”中当前的最好个体以广播的形式发送到各个子种群中,子种群用接收到的优秀个体替换本种群中当前的最差个体,从而实现迁移。

迁移算子具体操作时迁移规模取 $m=1$ 即每次迁移个体的数目为 1;迁移率 $t=10$ 指种群中的个体隔 10 代迁移一次,迁移间隔过小,不利于提高解的质量;迁移策略采用最优个体迁出和最差个体被替换的策略;迁移拓扑决定了各子群体间进行个体迁移的路径,在子种群间有多种迁移方式,如环形拓扑、领域拓扑、完全网络拓扑等^[4]。本文采用的是一种逻辑上的星型拓扑,即迁移只发生在各子种群和主节点上的“种群池”之间。

4.2 混合并行遗传算法描述及流程

(1) 初始化操作

a) 主进程(每个进程对应一个计算节点)调用 MPI 的广播函数 MPI_Bcast() 将遗传算法的初始参数:种群规模 S、最大进化代数 MaxGen、交叉概率 P_c 及变异概率 P_m 以广播的方式发送给其它进程;

b) 从进程按接收到的遗传参数独立的在各个节点内产生初始种群。

(2)各从进程独立并行进化

a)各进程调用 OpenMP 编译指导语句 # pragma omp parallel for 生成多个线程,由多个线程并行执行选择、杂交、变异遗传操作并计算适应度值;

b)主线程保留当前种群中产生的最优个体信息。

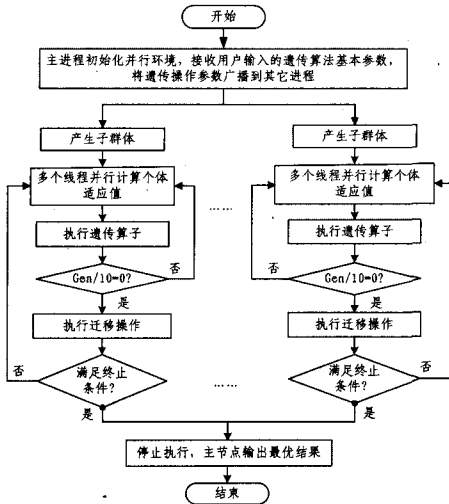


图3 混合并行遗传算法流程图

(3)并行执行迁移操作

a)若间隔代数 $gen \% 10 = 0$,从进程调用 MPI 的收集函数:MPI_Gather()向主节点中的种群池发送当前种群中的最优个体;否则继续执行第(2)步;

b)主节点接收所有从节点发送的最优个体后,从种群池中选出当前的最优个体以广播方式发送给其它节点;

c)各从节点接收主节点发送的优秀个体,以最差个体换出的策略替换本种群中的个体。

(4)判断算法是否结束

若当前进化代数 $gen = MaxGen$,则算法结束,由主进程输出运算结果,否则转到步骤(2)继续执行。

主节点种群池中始终保存着到目前为止所有种群中最好的 N 个个体(N 为节点数目)。这样一直迭代直到满足进化终止条件。最后主节点种群池中的最优个体即为所求得的最优解。

算法的主要执行过程如图3所示。

4.3 算法的性能分析

评价并行算法的标准有很多,其中最通用的一种标准就是计算加速比。加速比完整的表达式为:

$$\psi(n, p) \leq \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n)/p + \kappa(n, p)} \quad (1)$$

式中, $\psi(n, p)$ 表示在 p 个处理器上解决规模为 n 的问题时的加速比, $\sigma(n)$ 表示计算任务中内在的串行部分, $\varphi(n)$ 表示可以并行执行的计算部分, $\kappa(n, p)$ 表示并行开销所需的时间。

根据以上加速比的计算式(1),可以推导出 Amdahl 定律^[14]:

$$\psi \leq \frac{1}{f + (1-f)/p + \kappa(n, p)} \quad (2)$$

式中, ψ 为整个算法在并行机上所能达到的最大加速比, f 为整个算法中串行执行部分所占的比例($0 \leq f \leq 1$)。

Amdahl 定律基于解决固定规模的问题,提供了一种预测用一定数量的处理器并行解决问题时的加速比的上限,从而可以帮助决定一个程序是否值得并行化。

算法的计算负载,用程序中所执行的指令数目来度量,用 I 表示程序中总的指令数目, I_s 表示程序中串行部分的指令数目, I_p 表示程序中并行部分的指令数目。可以用以下公式计算程序中串行部分在整个算法中所占的比例 f 。

$$f = \frac{I_s}{I} = \frac{I_s}{I_s + I_p} \quad (3)$$

对本文具体实现的并行遗传算法程序进行分析,利用式(3)估算出程序中 10% 的程序必须串行执行,主要包括初始化并行环境,主节点广播并行遗传算法参数,接收各节点的最优迁移个体,输出运行结果等操作。程序中完全可以独立并行的双核 PC 数为 4,总的处理器核数为 8。如果用单纯 MPI 并行编程模式则只有进程级并行,每一个进程对应一个节点,根据 Amdahl 定律计算出所能达到的最大加速比理论值为:

$$\psi_1 = \frac{1}{0.1 + (1-0.1)/4} \approx 3.08 \quad (4)$$

采用混合并行编程模式后,每一个进程内部又生成并行执行的两个线程,经分析程序的每个进程部分中又有 80% 的程序可以完全由两个线程并行执行,主要包括具体的选择、交叉、变异遗传操作和个体适应度计算,由 Amdahl 定律计算出混合并行编程模式下所能达到的加速比为:

$$\psi_2 = \frac{1}{0.1 + (1-0.1) * (0.2/4 + 0.8/8)} \approx 4.26 \quad (5)$$

比较两种加速比得出:采用混合并行编程方式比单纯用 MPI 方式所能达到的加速比提高了 38.31%。

通信代价是进一步优化并行遗传算法的瓶颈,而这一代价主要消耗在被选个体在种群间的迁移上。算法中由于引入了种群池,大大减少了种群间的通信。同时迁移是在一定时间间隔内进行的,从而防止了子种群在未能充分进化的情况下将个体信息传播出去,减少了通信代价。由于节点内部以共享存储方式进行通信,其通信代价可以忽略,因此以采用以上策略后算法的整体通信代价大大降低。

5 算法测试与仿真结果分析

5.1 并行遗传算法的性能测试方法

对并行遗传算法来说,它适合固定的种群,即子群体中的数量反比于子种群的规模。为了更好地评价并行遗传算法的性能,要求求解速度和求解精度两个方面同时考虑,文献[15, 16]给出了以下两种评价 PGA 的指标:

(1)为适应度确定一个指标,分别用并行和串行方式执行,直到求出的解大于给定的值,然后用两种方式计算时间比,将其作为评价 PGA 的指标。

(2)设置一个确定的时间,如固定的进化代数,通过 PGA 优化,测试能够得到的最高适应度比利用串行的遗传算法在同样的时间内搜索到的最高适应度值高出多少。

本文先从理论上利用 Amdahl 定律对本混合并行遗传算法在 4 个双核 PC 结点上所能达到的最大加速比进行预测,然后利用以上两种评价指标分别从求解速度和求解质量两个方面对采用的测试函数进行评价,最后验证所预测的加速比。

5.2 测试算例及测试环境

对遗传算法的性能测试,通常采用国际上通用的基准测试函数进行测试^[18],本文采用 Rosenbrock 基准函数:

$$f1(x) = \sum_{i=1}^n 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2, x_i \in [-20, 20] \quad (6)$$

函数 $f1(x)$ 是很难极小化的病态函数, 这个基准测试函数的全局最小值为 0。

为突出本文提出的混合并行遗传算法的有效性, 我们取测试函数的维数为 2, 用传统的串行高斯变异实数遗传算法 (SGA) 和本文提出的混合并行遗传算法 (HPGA) 分别对以上基准函数独立运行 20 次后对最优值取平均值, 然后进行测试对比和性能分析。同时和文献[22]中的免疫克隆选择优化算法 (ICSOA) 对同一函数进行优化后的结果进行对比。

本试验是在实验室的 4 台双核 PC 机组成的集群系统上进行的, 具体的硬件配置如表 1 所列, 软件环境如表 2 所列。遗传算法采用高斯变异实数遗传算法, 最大进化代数取 10000, 种群大小为 1000, 交叉概率为 0.8, 变异概率为 0.06。

表 1 硬件配置

| 节点数 | CPU | 内存 | 交换机 |
|-----|---|-----|---------------------------------------|
| 4 | Intel(R)Core™2 Duo E7400 主频: 2.8GHz, 二级 cache: 3M | 2GB | TP-LINK TL-SF100810/100M 自适应以太网交换机 |

表 2 软件配置

| 操作系统 | 编译环境 | 并行环境 |
|-------------------------|------------------------------|----------------------------------|
| Windows XP Professional | Microsoft Visual Studio 2005 | Mpich2-1.1-win-ia32 OpenMP2.0 |

5.3 测试结果及分析

(1) Rosenbrock 函数的测试结果如表 3—表 5 所列。

表 3 测试函数优化结果

| 算法类型 | 最优值 | 平均值 | 均方差 |
|-------|--------------|--------------|--------------|
| SGA | 6.31566e-012 | 5.50959e-006 | 7.46393e-006 |
| HPGA | 3.02829e-014 | 2.17738e-011 | 3.31827e-011 |
| ICSOA | | 3.31e-007 | 3.30e-007 |

表 4 达到相同精度下 SGA 和 HPGA 所用时间

| 精度 | SGA(seconds) | HPGA(seconds) |
|------|--------------|---------------|
| E-9 | 2.14125 | 0.553485 |
| E-10 | 3.780353 | 0.988512 |
| E-11 | 4.121934 | 1.124465 |
| E-12 | 6.043432 | 1.678623 |
| E-13 | 6.404434 | 1.765653 |
| E-14 | 115.115345 | 29.898834 |

表 5 达到相同精度下 SGA 相对于 HPGA 的加速比

| 精度 | 加速比(Sp) |
|------|---------|
| E-9 | 3.86867 |
| E-10 | 3.82429 |
| E-11 | 3.66568 |
| E-12 | 3.60023 |
| E-13 | 3.62723 |
| E-14 | 3.85016 |

(2) 结果分析

从表 3 的仿真对比中可以看出, 在达到相同进化代数的情况下, 用本文的混合并行遗传算法和传统串行高斯变异实数编码遗传算法求解同一问题相比, 前者所求得的最优解的精度比有较大的提高。同时和文献[22]中的免疫克隆选择优化算法 (ICSOA) 对同一函数所求解相比也有一定的提高。说明本文提出的多核机群下的混合并行遗传算法在处理函数优化问题时, 由于各个子种群分别独立地在多个节点上并行进化, 同时通过迁移操作在各节点之间进行信息交换, 因此保持了种群的多样性, 有效地防止了早熟和局部收敛, 显著地提高了算法的全局搜索能力。

表 4 是对标准测试函数在求得不同精度时两种算法所用的时间统计, 表 5 是对不同精度情况下并行遗传算法相对串行遗传算法的绝对加速比。从表 5 的数据可以看出, 本混合并行遗传算法实际测试所得到的加速比平均达到了前面公式 (5) 中用 Amdahl 定律预测出的理论最大加速比的 86% 以上。因在前面的预测中忽略了并行开销, 所以得出的是一种理想情况下的上限值。这说明在固定负载的情况下, 本混合并行遗传算法的求解性能达到了预期的效果, 验证了本混合并行遗传算法模型的有效性。

结束语 本文在多核 PC 集群系统的基础上, 结合多核 PC 集群物理拓扑和“粗粒度-主从式”并行遗传算法逻辑结构的特点, 将 MPI 和 OpenMP 两种并行编程模型结合起来, 提出了一种具体的混合并行遗传算法实现模型。利用进程和线程两级并行方式实现混合并行遗传算法, 在混合并行遗传算法实现迁移操作时引入了“种群池”的思想, 大大减少了各种群之间的频繁通信, 从而有效地降低了系统的通信代价。从理论和实验两方面对该混合并行遗传算法进行了性能分析。对遗传算法标准测试函数求解的仿真实验表明, 与传统串行遗传算法相比本文混合并行遗传算法实现模型在求解精度和求解速度方面都有很大的提高。同时和单一的 MPI 并行方式相比本文混合模型在加速比上也有较大的提高, 随着问题规模的不断扩大, 如果同时增加处理器核数, 本文混合并行遗传算法将可以达到线性加速比。

参考文献

- [1] Zdeněk Konfrst. Parallel Genetic Algorithms: Advances, Computing Trends, Applications and Perspectives[C]// Proceedings of the 18th International Parallel and Distributed Processing Symposium, 2004
- [2] Xue Sheng-jun, Guo Shao-yong. The analysis and research of parallel genetic algorithm[J]. IEEE xPlore, 2008; 1-4
- [3] Thomaszewskib, Pabsts, Negri B I W. Parallel techniques for physically based simulation on multi-core process or architectures[J]. Computers and Graphics, 2008(32): 25-40
- [4] Cantú-Paz E. A Survey of Parallel Genetic Algorithms[J]. Urbana2 Champaign: Illinois Genetic Algorithms Laboratory, 1996
- [5] 侯广坤, 骆江鹏. 一种理想并行遗传算法模型[J]. 软件学报, 1999, 10(5): 557-559
- [6] 管宇, 徐宝文. 基于模式迁移策略的并行遗传算法[J]. 计算机学报, 2003, 26(3): 294-301
- [7] 赖鑫生, 张明义. 基于渗透原理迁移策略的并行遗传算法[J]. 计算机学报, 2005, 28(7): 1146-1152
- [8] 吴浩扬, 常炳国. 基于模拟退火机制的多种群并行遗传算法[J]. 软件学报, 2000, 11(3): 416-420
- [9] 刘晓平, 安竹林. 基于 MPI 的主从式并行遗传算法框架[J]. 系统仿真学报, 2004, 16(9): 1938-1940
- [10] 刘科研, 李运化. 基于分布式并行遗传算法的电力系统无功优化[J]. 北京航空航天大学学报, 2008, 34(1): 27-30
- [11] 殷新春, 仇亮. 基于主从式并行遗传算法的 S 盒优化算法[J]. 计算机工程与应用, 2008, 44(24): 112-117
- [12] 陈佳, 吴跃. 基于并行遗传算法的企业互操作资源选择[J]. 电子测量与仪器学报, 2009, 23(1): 101-106
- [13] Rabenseifner R, Hager G, Jost G. Hybrid MPI/OpenMP Parallel Programming on Cluster of Multi-core SMP Nodes. Parallel, Distributed and Network-based Processing[C]// The 17th Euro-micro International Conference on Digital Object Identifier.

- [14] Quinn M J. MPI与OpenMP并行程序设计[M]. 陈文光, 武永卫, 等译. 北京: 清华大学出版社, 2004, 10
- [15] Hanyf, Esbensenh, Song L J. Performance improvement of a genetic algorithm for floor planning with parallel computing technology[J]. IEEE, 1997; 642-786
- [16] Calegari P, Guidec F, Kuonen P. Parallel Island2 Based Genetic Algorithm for Radio Network Design[J]. Journal of Parallel Distributed Computing, 1997, 47(1): 68-102
- [17] Arunadevi J, Johnsanjeevkumar A, Sujutha N. Intelligent transport route planning using parallel genetic algorithms and MPI in high performance computing cluster[C]//The 15th International Conference Advanced Computing and Communications. 2007
- [18] 任子武, 伞冶. 实数遗传算法的改进及性能研究[J]. 电子学报, 2007, 35(2): 269-274

- [19] Yussof S, Razali R A, See O H, et al. A Coarse-grained Parallel Genetic Algorithm With Migration for Shortest Path Routing Problem[C]//The 11th IEEE International Conference on High Performance Computing and Communications. 2009
- [20] 汪少敏, 赵猛. 基于多核处理器并发计算软件构架设计与实现[J]. 计算机科学, 2008, 35(7): 283-285
- [21] Dong Li, de Dupinski B R, Nikolopoulos D S. Hybrid MPI/OpenMP power-aware computing[C]//Parallel & Distributed Processing (IPDPS); 2010 IEEE International Symposium on Digital Object Identifier. 2010; 1-12
- [22] Xu Guang-hua, Liu Dan, Liang Lin. Immune Clonal selection optimization method with combining mutation strategies[J]. Journal of Xian Jiao Tong University, 2007, 19(2): 177-181

(上接第164页)

表3 实验结果(AUC) $n=2000000$

| Algo | NS | PG | Hep | Jazz | Email | Cele | PB | USAir |
|------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| CN | 0.9335 | 0.5882 | 0.8900 | 0.9252 | 0.8429 | 0.8779 | 0.9259 | 0.9130 |
| AA | 0.9321 | 0.5877 | 0.8905 | 0.9368 | 0.8420 | 0.9298 | 0.9282 | 0.9301 |
| RA | 0.9366 | 0.5881 | 0.8904 | 0.9425 | 0.8431 | 0.9355 | 0.9301 | 0.9393 |
| IA | 0.9383 | 0.5884 | 0.8905 | 0.9476 | 0.8443 | 0.9373 | 0.9301 | 0.9348 |

表4 实验结果(Precision) $L=100$

| Algo | NS | PG | Hep | Jazz | Email | Cele | PB | USAir |
|------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| CN | 0.8214 | 0.1024 | 0.4101 | 0.8090 | 0.2821 | 0.1947 | 0.4101 | 0.5887 |
| AA | 0.9544 | 0.0689 | 0.3710 | 0.8267 | 0.3069 | 0.2483 | 0.3710 | 0.6073 |
| RA | 0.9585 | 0.0546 | 0.2411 | 0.8178 | 0.2484 | 0.3123 | 0.2411 | 0.6209 |
| IA | 0.9619 | 0.1061 | 0.4628 | 0.8832 | 0.3426 | 0.2887 | 0.4628 | 0.6100 |

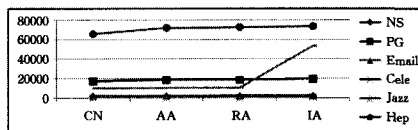


图2 算法运行时间(单位:毫秒)

结束语 随着复杂网络的快速发展,基于网络结构的链接预测方法得到了不断的发展与完善。本文从节点共同邻居之间相互关系的角度提出了一种基于节点相似性的链接预测算法——节点引力算法,该算法是在考虑单个共同邻居的同时,考虑所有共同邻居之间的相互关系。同时,本文将新算法与其他优秀的基于节点相似性的算法同时作用于8种不同的现实网络,在不丢失时间效率的情况下,新算法取得了更好的预测准确率。下一步,我们的工作将致力于研究基于节点相似性的预测算法与网络拓扑结构的关系,并积极研究大规模现实网络的链接预测问题。

参考文献

- [1] Getoor L, Diefl C P. Link mining: a survey[J]. ACM SIGKDD Explorations Newsletter, 2005, 7(2): 3-12
- [2] Lv Lin-yuan. Link Prediction on Complex Networks[J]. Journal of University of Electronic Science and Technology of China, 2010, 9: 5-39
- [3] Sarukkai R R. Link prediction and path analysis using markov chains[J]. Computer Networks, 2000, 33: 377
- [4] Zhu J, Hong J, Hughes J G. Using markov chains for link prediction in adaptive web sites[J]. Computer Science, 2002, 2311: 22
- [5] Popescul A, Ungar L. Statistical relational learning for link prediction[C]//Proc. Workshop on Learning Statistical Models-

- from Relational Data, 2003. New York: ACM Press, 2003: 81
- [6] Clauset A, Moore C, Newman M E J. Hierarchical structure and the prediction of missing links in networks[J]. Nature, 2008, 453: 98-101
- [7] Lichtenwalter R N, Lussier J T, Chawla N V. New Perspectives and Methods in Link Prediction[C]//ACM SIGKDD. 2010
- [8] Liben-Nowell D, Kleinberg J. The LinkPrediction Problem for Social Networks[J]. Journal Am. Soc. Inform. Sci. Technol, 2007, 58: 1019
- [9] Adamic L A, Adar E. Friends and neighbors on the web[J]. Social Networks, 2003, 25: 211
- [10] Zhou Tao, Lv Lin-yuan, Zhang Yi-cheng. Predicting missing links via local information[J]. The European Physical Journal B, 2009, 71: 623
- [11] Hanely J A, Mcneil B J. The meaning and use of the area under a receiver operating characteristic curve[J]. Radiology, 1982, 143: 29-36
- [12] Herlocker J L, Konstann J A, Terveen K, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transaction Information System, 2004, 22(1): 5-53
- [13] Newman M. Network DataSets [OL]. <http://www-personal.umich.edu/~mejn/netdata/>
- [14] Batageli, Mrvar A. Pajek Datasets[OL]. <http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm>
- [15] Watts D J, Strogatz S H. Collective dynamics of small-world networks[J]. Nature, 1998, 393: 440
- [16] Newman M E J. Assortative mixing in networks[J]. Phys. Rev. Lett, 2002, 89: 208701
- [17] Lv Lin-yuan, Zhou Tao. Similarity index based on local paths for link prediction of complex networks[J]. Phys. Rev. E, 2009, 80: 046122