

并行环境下的同步异步 PSO 算法

职为梅¹, 王芳¹, 范明¹, 杨勇²

(1. 郑州大学信息工程学院, 河南 郑州 450052;

2. 黄河水利科学研究院, 水利部堤防安全与病害防治工程技术研究中心, 河南 郑州 450003)

摘要:并行计算能够有效地缩减求解大规模问题的时间。文中在介绍了粒子群算法(Particle Swarm Optimization algorithm)的基础上,对 PSO 算法的同步异步模型进行分析,给出了并行环境下的同步异步 PSO 算法。该并行算法在联想深腾 1800 大型计算机上测试。实验证明 PSO 算法具有较高的并行性,并行算法明显提高了求解的速度。

关键词:PSO 算法;并行计算;并行 PSO 算法;同步;异步

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2009)03-0123-04

Synchronous and Asynchronous PSO Algorithm of Parallel Circumstance

ZHI Wei-mei¹, WANG Fang¹, FAN Ming¹, YANG Yong²

(1. School of Information Engineering, Zhengzhou University, Zhengzhou 450052, China;

2. Research Center on Levee Safety & Disaster Prevention of Ministry of Water Resource, Yellow River Institute of Hydraulic Research, Zhengzhou 450003, China)

Abstract:Parallel computing can curtail the time of solving large-scale problems effectively. Under introducing the biologic character of particle swarm optimization algorithm (PSO) and analyzing synchronous and asynchronous model, presented the synchronous and asynchronous algorithm of particle swarm optimization under parallel circumstance. The parallel algorithm was tested in the cluster of Lenovo Shenton 1800. The experimentations proved that PSO algorithm had higher parallelism, and the parallel arithmetic improved the efficiency obviously.

Key words:PSO algorithm; parallel computing; parallel algorithm of PSO; synchronous; asynchronous

0 引言

进化计算技术是由生物进化规律而演化出的一种搜索和优化计算方法,主要包括遗传算法(Genetic Algorithm, GA)、进化规划(Evolutionary Programming, EP)、进化策略(Evolutionary Strategies, ES)以及遗传编程(Genetic Programming, GP)。同其他的进化算法相似,粒子群优化算法(Particle Swarm Optimization, PSO)也是基于个体的协作与竞争来完成复杂搜索空间中最优解的搜索。PSO 适合于求解连续函数的优化问题,也有将 PSO 用于解决一些离散型优化问题的相关研究,例如用于求解 TSP 问题^[1],任务分配问题等组合优化问题^[2]。PSO 在国内研究刚刚起步,近年来也主要是集中于应用和文献综述^[3]方面。

对于 PSO 改进方面上,先前的算法实现基本上都

集中在同步处理上,即各个粒子的每一次搜索行为都处在同一个迭代步中进行的。因此,粒子在很大程度上失去了独立性,以至于最优粒子的信息不能及时地共享。粒子群算法异步模型,在算法的程序实现上可以采用 Java 的多线程^[4]或是 C++ 的 MFC 多线程技术^[5],把每个粒子看成为一个独立的线程,运行中的粒子充分表现出高度的独立性,而在种群层次上表现为异步性^[4]。并行机环境下的粒子群同步异步算法则又比在单机环境下的算法做了进一步的改进和提高,在算法的程序实现上可以采用利用 MPI 库函数加上 C++ 多进程下的多线程思想或是直接利用内存共享技术,把整个粒子群按照一定规则分给多个进程来同时并行处理。通过实现串行算法到并行算法,粒子群算法的同步和异步算法的效率都得到了较好的提高。

1 PSO 算法

PSO 算法最早是由 Eberhart 和 Kennedy 于 1995 年提出的^[6,7],它的基本概念源于对人工生命(Artifi-

收稿日期:2008-07-02

基金项目:中央级公益性科研院所基本科研业务费专项资金(HKY-JBYW-2007-22)

作者简介:职为梅(1977-),女,讲师,研究方向为数据库数据挖掘。

cial Life) 和鸟群捕食行为的研究。设想这样一个场景:一群鸟在随机搜寻食物,在这个区域里只有一块食物,所有的鸟都不知道食物在哪里,但是它们知道当前的位置离食物还有多远。那么找到食物的最简单有效的策略就是搜寻目前离食物最近的鸟的周围区域^[4]。

PSO算法就从这种生物种群行为特性中得到启发并用于求解优化问题。在PSO中,每个优化问题的潜在解都可以想象成 D 维搜索空间上的一个点,称之为“粒子”(Particle),所有的粒子都有一个被目标函数决定的适应值(Fitness Value)。搜索正是在这样一群随机粒子而组成的一个种群中进行的^[4]。

通过数学描述可以是:假设在一个 D 维的目标搜索空间中,有 m 个代表潜在问题解的粒子组成的一个种群 $S = \{X_1, X_2, \dots, X_m\}$,其中 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, $i = 1, 2, \dots, m$ 表示第 i 个粒子在 D 维解空间的一个矢量点。将 X_i 代入一个与求解问题相关的目标函数可以计算出相应的适应值。用 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, $i = 1, 2, \dots, m$ 记录第 i 个粒子自身搜索到的最好点(所谓最好,是指适应值最好)。而在所有这些粒子中,有一个是最好的,将这个粒子的编号记为 g ,则 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ 就是种群搜索到的最好值,其中 $g \in \{1, 2, \dots, m\}$ 。用 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, $i = 1, 2, \dots, m$ 表示第 i 个粒子的速度。

PSO算法一般是采用下面的公式对粒子进行操作的。

$$V'_i = w * V_i + c_1 * r_1 * (P_i - X_i) + c_2 * r_2 * (P_g - X_i) \quad (1)$$

$$X'_i = X_i + V'_i \quad (2)$$

其中 $i = 1, 2, \dots, m$,学习因子 c_1, c_2 是两个正常数,一般取值为2, r_1, r_2 是均匀分布于 $[0, 1]$ 之间的两个随机数, w 是惯性权因子,有许多的文献对其进行了研究,认为 w 取值依据迭代过程变化有利于算法的改进^[8]。

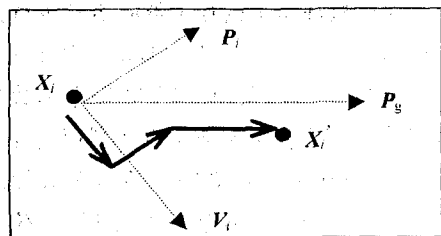


图1 粒子位置的更新

公式(1)的第一部分称为记忆项,表示上次速度大小和方向的影响;公式的第二部分称为自身认知项,是从当前点指向此粒子自身最好点的一个矢量;公式的第三部分称为群体认知项,是一个从当前点指向种群

最好点的一个矢量,反映了粒子间的协同合作。可见,公式(1)是粒子依据先前的速度、自身最好经验,以及群体最好经验这三个因素实现对速度的更新。然后按照公式(2),粒子从当前位置飞向新的位置。这个过程可以用图1^[4]加以形象描述。

2 并行 PSO 算法

2.1 PSO 同步异步算法

2.1.1 同步算法

PSO是在研究种群行为的基础上提出的。种群中的粒子是相互合作的,它们协同搜索,贡献各自的最好点信息,从而形成公共的整体最好点信息,即 P_g ,共同提高整体认知水平,并且这个公共信息成为影响每个粒子下一次飞翔方向的一个重要因素。种群中的粒子又是相互竞争的,其含义有二:首先是各个粒子间的竞争——即贡献自身的最好点试图取代整体的最好点,成为种群中的“胜者”,把它称为“整体竞争”,这与“合作”是事物辩证统一的两个方面。其次是粒子自身的竞争,即粒子自身有一个搜索到的最好点 P_i ,试图摆脱整体认知的控制,成为影响粒子下一次飞翔方向的另一个重要因素。

可见,PSO是一种基于群智能(Swarm Intelligence)的优化技术,粒子间表现出高度的独立性和协同性等生物特征,在现有的算法实现上主要有同步和异步两种模型。

仔细分析在PSO算法的迭代循环步骤,注意到:在同一个迭代步中,根据所有粒子计算得到的适应值而找出的最好点标号 $gBest$,从而确定了这个迭代步的种群最好点 P_g ,这以后所有粒子的更新操作都是针对相同的 P_g 而进行的,这在行为特性上表现为所有的粒子都向着相同的一个最好目标同步移动。可见,在任何一次迭代循环中,所有的粒子都以共同的整体认知水平进行了一次完整的搜索,可以把这种实现称为同步模型(Synchronous Model)^[4]。

2.1.2 异步算法

PSO的同步处理模型并不十分合理。因为PSO来自生物行为模拟,正如鸟的觅食,每只鸟都是独立的个体,它的每一次觅食动作都不会等待所有的鸟进行了一次觅食动作后再做反应的。实际上,在鸟群协同觅食的每一个时刻,任何一只鸟若是先发现了好的食物位置,就可以通过鸣叫等形式立即通知所有鸟,使之及时成为种群的整体认知。所以要是考虑有着这样一种鲜明个体独立行为、即时种群通讯行为的处理模式,这就是该算法的异步模型^[5]。

该处理模型如下:

(1)一个 PSO 种群由多个独立的粒子组成,即粒子具有自身的最好适应值、点位置矢量、速度矢量等特征,并且对于搜索中的任意时刻,每个粒子行为都是并行的。

(2)粒子间共享整体最好点信息,这个共享信息具体包括整体最小适应值、相应的点位置 P_g 以及相应的粒子标号 $gBest$ 。

(3)在搜索中,当任意一个粒子发现其计算的适应值小于共享信息中的最小适应值时,则会立即更新共享信息^[4]。

图2表示了这种模型结构,其中“粒子”是个独立的搜索单元,所具备的功能包括:计算自己的适应值,与共享信息的通讯,更新速度和位置等。

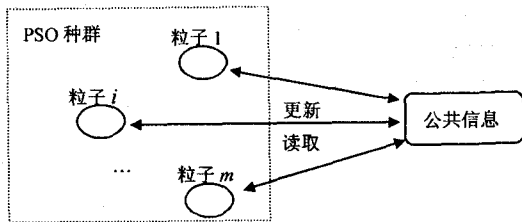


图2 异步 PSO 模型结构图

2.1.3 分析比较

由此可以得到在粒子群算法的具体实现上,其同步模型和异步模型的不同之处就在于:同步模型是在所有的粒子都搜索一遍后得到自身的最优值,然后从这些值中比较得到该次搜索得到的整体最优值,从而对共享的整体最优解进行更新。而异步模型则是充分体现了粒子之间的竞争性,在一次搜索中,只要粒子搜索到的自身最优解比现在共享的整体最优值好,就对整体最优值进行更新。

并行与单机环境下的算法的不同在于:并行算法是把所有的粒子按照一定规则分成块,然后分配给并行机的各个节点,把将每个粒子的行为看成为一个独立的处理过程,运行中的粒子充分表现出高度的独立性,并且通过进程之间的通讯来更新种群最优信息。各个进程同时处理自己被分配到的数据信息,同步算法是在所有粒子计算完一遍后各进程进行一步同步操

作得到种群最优信息,异步算法则是通过内存共享和 P-V 操作来充分实现粒子的竞争和并行性。并行算法是在操作系统层次上利用多进程来大大提高了算法的效率。

2.2 并行 PSO 算法的主要技术

2.2.1 进程通信

并程序通信方式通常分为数据并行模型、消息传递模型、共享内存模型。并行 PSO 算法采用消息传递模型,在各个并行从进程之间不需要消息传递。需要进行消息传递的情况通常有以下几种^[9]:

(1)适应值的传递:适应值由从进程传回主进程,主进程比较并更新 f_{best} , f_{best}^i 优于 f_{best} , 则 $f_{best} = f_{best}^i$, 并发送 f_{best} 到所有从进程。

(2)终止信号的传递:程序结束时,主进程将终止信号传递给各个从进程,从而彻底终止程序运行。

2.2.2 进程同步

为了同步所有进程,规定消息传递的顺序如下:

(1)主进程等待每个 P_j 发来的消息,直到接收 n 个消息。

(2)在(1)中, P_j 发送消息后,则等待主进程消息。

(3)当(1)的工作完成后,主进程比较并更新 f_{best} , 然后向每个从进程 P_j 发送消息。主进程进入工作(1)。

(4)在(3)的工作完成后, P_j 进入工作状态。

2.2.3 共享内存

在实现并行算法的过程中,开辟了共享内存,在这块内存中,保存了整体最优值这个重要的需要更新的数据,通过内存共享,便可以使得各个从进程中的粒子实时地认知最新的整体最优值信息并对自我信息进行更新。

3 算法实验测试

为验证算法的有效性,应用 PSO 的这两种模型求解三个经典测试函数的最优值问题,这三个测试函数的最优值都是 0。表 1 列出其特征及其初始化范围。

表 1 三个测试函数及初始化范围

测试函数	函数表达式	初始化范围	函数说明
Sphere	$f_1(x) = \sum_{i=1}^n (x_i^2)$	$[-5.12, 5.12]$	简单的单峰函数,在 $x_i = 0, i = 1, 2, \dots, n$ 时有极小值 0
Rosenbrock	$f_2(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-10, 10]$	单峰函数,其中二维的 Rosenbrock 函数是一个非凸函数,在 (1,1) 处达到极小值
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	多峰函数,全局极小是在 $x_i = 0, i = 1, 2, \dots, n$, 其局部极小点是 $x_i = \pm k\pi\sqrt{i}, i = 1, 2, \dots, n, k = 0, 1, 2, \dots$

3.1 单机实验结果

已经研究出的单机环境下该算法的实验结果如下:

在两种模型,运行的参数保持一致,即:取 PSO 种群规模为 $m = 30$,学习因子 $c_1 = c_2 = 2.0$,惯性权值 $w = 1$ 随在迭代过程中线性递减,变化范围是 $1.1 \sim 0.4$,最大迭代次数取为 6000。

测试工作的物理运行环境:主处理器 Intel Celeron 400MHz(主频 66MHz),内存 256M SDRAM,操作系统为 Windows 2000 Professional^[1]。

三个测试函数的计算结果列于表 2^[4],其中每个测试函数执行时间 T_1, T_2 是经过 30 次运算取均值得到。

表 2 两种 PSO 处理模型的计算结果比较

测试函数	维数	终止条件 (精度要求*, 最大迭代数)	PSO 同步模型 执行时间 T_1 (ms)	PSO 异步模型 执行时间 T_2 (ms)
Sphere	15*	0.01; 6000	2475.9	1141.867
Rosenbrock	10	5.0; 6000	2389.533	1682.533
Griewank	10	0.1; 6000	3857.167	2585.1

* 精度要求数值的选择与测试函数本身、维数等因素有关。

* 考虑到 Sphere 函数较简单,所以选择计算维数为 15,以提高复杂度。

3.2 并行机实验结果

在两种模型,运行的参数保持一致,即:取 PSO 种群规模为 $m = 30$,学习因子 $c_1 = c_2 = 2.0$,惯性权值 $w = 1$ 随在迭代过程中线性递减,变化范围是 $1.1 \sim 0.4$,最大迭代次数取为 6000。

测试工作的物理运行环境:主处理器 Intel(R) Pentium(R) 4 CPU 2.93GHz,内存 256M SDRAM,操作系统为 Windows XP Professional。登陆到 Lenovo 深腾 1800 集群系统的并行环境下进行实验。

三个测试函数的计算结果列于表 3,其中每个测试函数执行时间 T_1, T_2 是经过 50 次运算取均值得到的平均结果。

表 3 并行机环境下两种 PSO 处理模型的计算结果比较

测试函数	维数	进程 数目	终止条件 (精度要求, 最大迭代数)	PSO 同步 模型执行 时间(ms)	PSO 异步 模型执行 时间(ms)	得到适 应值精 度
Sphere	15	10/20	$1e-3$; 6000	16.160	13.731	0.001
Rosenbrock	10	4	$1e-1$; 6000	112.345	102.978	1.0
Griewank	10	10	$1e-3$; 6000	17.833	14.874	0.0001

* 精度要求数值的选择与测试函数本身、维数等因素有关。

* 并行机环境下的实验结果为与单机实验进行对比,某些数据

采用相近或一致。

* 值的精度和进程数目选择有关,并不一定进程数目越多效果就越好,要根据实际情况选择。

通过实现结果的对比可以发现并行环境下的算法的效率比单机环境下有了明显的提高:(1)执行时间提高了几十甚至上百倍;(2)适应值的精度也有了明显的提高。

4 结束语

在分析粒子群优化算法的生物特性的基础上提出了并行 PSO 的同步异步算法思想,并在并行平台下实现了同步和异步粒子群算法:把将每个粒子的行为看成为一个独立的处理过程,发布到各个技术节点中,运行中的粒子充分表现出高度的独立性,并且通过进程之间的通讯来更新种群最优信息。这将对于研究 PSO 算法提供了一种新思路。

通过三个经典的测试函数运算表明:在并行平台下所实现同步和异步粒子群算法在时间效率和精度性能上很有优势。

参考文献:

- [1] Clerc M. Discrete Particle Swarm Optimization[EB/OL]. Illustrated by the Traveling Salesman Problem. 2000. <http://www.mauriceclerc.net>.
- [2] 曾建潮,介静,崔志华.微粒群算法[M].北京:科学出版社,2004.
- [3] 李爱国,覃征,鲍复民,等.粒子群优化算法[J].计算机工程与应用,2002(21):1-3.
- [4] 李建勇,俞欢军,张丽平,等.基于 Java 多线程技术的粒子群优化算法异步模型[J].计算机工程,2004,30(22):134-136.
- [5] 罗建宏,张忠能.并行仿真的粒子群优化算法异步模式研究[J].计算机仿真,2005,22(6):76-78.
- [6] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proc. IEEE int'l conf. on neural networks. Piscataway, NJ: IEEE service center, 1995:1942-1948.
- [7] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]//Proceedings of the sixth international symposium on micro machine and human science. Piscataway, NJ: IEEE service center, 1995:39-43.
- [8] 谢晓峰,张文俊,杨之廉.微粒群算法综述[J].控制与决策,2003,18(2):129-134.
- [9] 张蕾,杨波.并行粒子群优化算法的设计和实现[J].通信学报,2005,26(1A):290-292.