

基于分布式网络并行环境的惩罚函数法

范波芹¹,索丽生²

(1.浙江省水利厅咨询中心,浙江 杭州 310009;2.中华人民共和国水利部,北京 100000)

摘要 基于 PVM 分布式编程环境,将传统的惩罚函数法加以改进,提出惩罚函数并行算法的概念.该算法不但具有编程方便、可移植性强和能有效地减少计算时间等优点,而且通讯量少,对传统的惩罚函数法改动少,只需有一网络和 PVM 并行编程环境就能实现.

关键词 网络并行环境;分布式;惩罚函数法

中图分类号 TP311 **文献标识码** A **文章编号** 1000-198X(2002)02-0116-03

1 并行环境下的惩罚函数法^[1]

传统惩罚函数法的基本思想是将约束函数通过赋以一个或几个可以调整的参数(即罚因子)与目标函数一起构成一个无约束条件的新目标函数.由于是依次计算目标函数值,这种方法本身没有并行性.为此,本文对传统的惩罚函数法进行改进,以便能在并行环境下使用,从而达到缩短计算时间的目的.

并行环境下惩罚函数法的主要步骤是:

- a. 读入初始点 $X^{(0)}$,控制精度 ϵ ,初始罚因子 $r^{(0)}$,自变量类型以及约束类型;
- b. 按照自变量类型和约束类型确定各种参数;
- c. 进行通讯,使得 P 台机器上都有初始参数值;
- d. 每台机器上针对各自的罚因子进行无约束最优化计算,即第 1 台机器以 $r^{(1)} = r^{(0)}$ 为罚因子,第 2 台机器以 $r^{(2)} = r^{(0)}/10^{(1/3)}$ 为罚因子, P 台机器以 $r^{(P)} = r^{(0)}(10^{-1/3})^{P-1}$ 为罚因子优化,找出相应于 $r^{(i)}$ 的极小化问题的解 $X^*(r^{(i)})$;
- e. 进行通讯,每台机器将计算结果传回主进程,然后处于等待状态;
- f. 主进程对结果进行比较,若任意相邻进程产生的结果满足精度要求,则结束任务,输出第 P 台机器的计算结果;若精度不能满足要求,主进程则调用外推,推算出 $X^*(r^{(P+1)})$,然后以推算出的 $X^*(r^{(P+1)})$ 代替 $X^{(0)}$,以 $r^{(P)}$ 代替 $r^{(0)}$ 转 d.

2 PVM 环境下惩罚函数法并行算法的实现

PVM(Parallel Virtual Machine)是一个能用来进行并行程序设计的软件环境^[2],基于 PVM 的网络主从式编程模式如图 1 所示.主从式(Master-Slave)编程模式适用于粗粒度并行计算,因此在设计算法时通讯与运算尽量重叠.本文所给出的惩罚函数法的并行计算是程序级的并行,因此特别适合主从式编程模式.其问题的求解由主进程和从进程共同完成.进程的生成、管理及相互间的通信等并行过程通过调用 PVM 库函数共同实现,不需进行通讯的部分与串行程序一样.

2.1 Master 算法描述

- a. 读入需产生的子进程数 n ;
- b. 产生 n 个子进程;
- c. 参与并行环境下混合惩罚函数法的 c 步运算;
- d. 参与并行环境下混合惩罚函数法的 f 步运算.

2.2 Slave 算法描述

执行前面给出的并行环境下的惩罚函数法的 d 和 e 步. 在网络中的每台微机上装入 PVM 程序和 Slave 程序的执行文件, 其中 1 台装入 Master 程序的执行文件, 运行 Master 程序就可启动所有的 Slave 程序, 执行并行计算. Master 程序综合最终的结果并将其输出. PVM 作为网络并行编程平台, 其操作系统为 Windows 95, Fortran 编程环境采用 Powerstation, 使用以太网上的奔腾—100, 以此作为并行计算的硬件和软件环境.

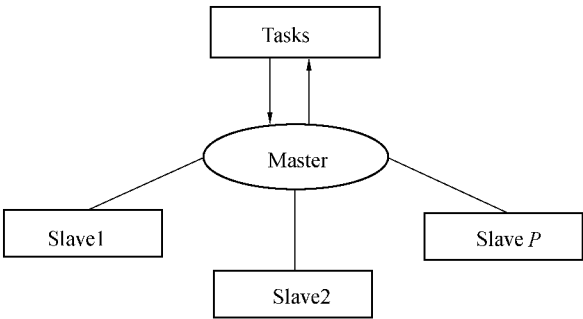


图 1 主从式模式
Fig.1 Master-slave model

3 算 例

为了考证文中给出的并行环境下的惩罚函数法的并行效果, 本文采用 2 个简单数学极小值问题来验证.

- a. 求函数 $f(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 1)^2$ 的极小值, 满足约束条件
- $$x_1 - 2x_2 + 1 = 0 \quad -0.25x_1^2 - x_2^2 + 1 \geq 0$$
- b. 求函数 $f(x_1, x_2) = 60 - 10x_1 - 4x_2 + x_1^2 + x_2^2 - x_1x_2$ 的极小值, 并满足约束条件
- $$-x_1 - x_2 + 16 \geq 0 \quad 0 \leq x_1 \leq 9 \quad 0 \leq x_2 \leq 9$$

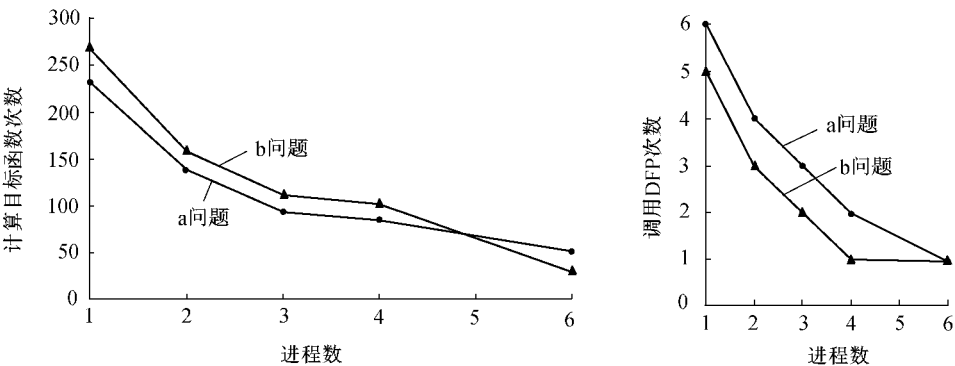


图 2 进程数与计算次数的关系
Fig.2 Relation between process times and calculation times

为检验该并行环境下的惩罚函数法的并行效果, 求解时分别采用了 1 个进程、2 个进程、3 个进程、4 个进程、6 个进程. a 问题优化结果为 $x_1 = 0.823, x_2 = 0.911, f(x_1, x_2) = 1.393$; b 问题优化结果为 $x_1 = 8.0, x_2 = 6.0$. 由图 2 可知, 每个进程计算目标函数的次数和调用 DFP (Davidon Fletcher Powell 变尺度法) 进行无约束循环的次数几乎是随着所使用的进程数的增加成倍减少, 若多台机器并行计算, 每台机器上有一个进程, 则并行计算的时间也应成倍地减少. 同时也发现, 进程增加到一定程度, 计算目标函数的次数和调用 DFP 的次数对进程的增加不是很敏感. 因此, 当优化问题不大时, 进程数目不必取得很大, 随优化问题的加大, 可适当地增加进程数目, 以达到缩短计算时间、提高计算效率的目的.

4 结 论

本文首次研究了在网络和 PVM 分布式并行环境下的惩罚函数法. 该算法具有编程方便、可移植性强和能有效地减少计算时间等优点, 并且其通讯量少, 对传统的惩罚函数法改动少, 只需有一网络和 PVM 并行编程环境就能实现, 可以为一般的工程人员所接受. 同时若将该算法用于结构优化, 也具有不必划分子结构的优点, 而将原来每次循环过程中的结构分析计算分别由各个进程完成, 可有效地减少计算时间. 因此, 本文提出的惩罚函数并行优化算法在结构优化中也将有良好的应用前景.

参考文献:

[1] 范波芹. 潮汐电站厂房浮运结构的浮运稳定和结构优化分析[D]. 南京: 河海大学, 2000.

[2]孙家昶,张林波,迟学斌,等.网络并行计算与分布式编程环境[M].北京:科学出版社,1996.10~15.

Penalty Function Method Based on Distributed Network Parallel Environment

FAN Bo-qin¹, SUO Li-sheng²

(1. Water Conservancy Department of Zhejiang Province, Hangzhou 310009, China;

2. Ministry of Water Resources, Beijing 100053, China)

Abstract: The conventional penalty function method is improved, and the concept of parallel arithmetic for the penalty function method is proposed. The arithmetic is easy to be programmed and transplanted, and has the merit of time saving. With a small quantity of communication and little modification of the conventional penalty function method, the arithmetic can be realized with only a network and a PVM parallel environment.

Key words: network parallel environment; distributed; penalty function method

《河海大学学报(哲学社会科学版)》征订启事

(CN32-1521/C, ISSN1671-4970, 季刊, 自办发行)

本刊以马列主义、毛泽东思想和邓小平理论为指导,坚持实事求是的思想路线,贯彻“双百”方针,弘扬时代主旋律,理论联系实际,为社会主义物质文明与精神文明建设服务。

本刊为哲学与社会科学类学术期刊,主要刊登哲学、政治学、文学、史学、语言学、法学、社会学、心理学、经济学、管理学、高等教育学等社会科学方面的研究成果、学术论文、综述等学术性文章。本刊可供上述有关专业的研究人员、管理人员和大专院校师生阅读与参考。本刊为中国学术期刊(光盘版)入编期刊与中华期刊网入编期刊。

本刊由河海大学主办,每季末出版,国内外公开发行,每期定价5元,每期邮费1元,全年订费24元。欢迎广大读者直接向编辑部订阅。联系地址:南京市西康路1号《河海大学学报(哲学社会科学版)》编辑部,邮政编码210098,联系电话(025)8786376。