广西科学院学报
Journal of Guangxi Academy of Sciences
2006,22(4):309～311
Vol. 22,No. 4 November 2006

基于协同优化方法的分布式协同演化系统的设计与实现*

Design and Implement of A Distributed Co-evolution System Based on Collaborative Optimization

陈秋莲,李陶深,黄毅然
CHEN Qiu-lian,LI Tao-shen,HUANG Yi-ran

(广西大学计算机与电子信息学院,广西南宁 530004)
(School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:基于遗传算法的协同优化方法的基本思想,在C#平台下设计与实现了一个分布式协同演化系统。该系统具有长基因群体分解、分配任务给子服务器、综合、演化、产生新的群体、短基因群体在局部空间上演化、数据输入、输出等功能。

关键词:协同演化 协同优化 遗传算法 系统级优化
中图法分类号:TP311 **文献标识码**:A **文章编号**:1002-7378(2006)04-0309-03

Abstract: A distributed co-evolution system is developed in the platform of C # according to the basic knowledge of collaborative optimization of genetic algorithm. There are long gene groups and short gene groups in the system. The long gene groups can be decomposed and assign tasks to sub-servers, and finally create new groups. The short gene groups evolve in local rooms, and perform input and output of data.

Key words: co-evolution, collaborative optimization, genetic algorithm, optimization in system level

基于遗传算法的协同优化方法是一种分布式、多级的优化方法。其主要思想是把复杂系统的多学科设计问题分解为各学科优化设计问题,然后用某种策略来协调各学科的设计结果[1]。协同优化的基本框架主要由两部分组成:系统级优化和多个子系统优化。子系统优化器负责与本学科有关的设计变量优化,优化目标是使该子系统优化方案与系统级的目标方案之间的差异最小。系统级优化的任务是分配和协调各个子系统间的设计活动,是使系统总体目标最优。在并行或分布式系统上进行演化计算,不仅可以提高问题的速度和解的质量,甚至可以获得超线性加速比。本文基于遗传算法的协同优化方法的基本思想,设计与实现了一个分布式协同演化系统,该系统可以加快遗传算法演化计算的收敛速度,提高搜索效率和解的质量。

1 系统设计思想

分布式协同演化系统的基本思想是问题分解,分而治之。将长基因的大群体划分为一些短基因的子群体,每个子群体自成为一个演化单位,有自己的选择和演化操作。当满足指定代数时,子群体个体迁移,即子群体的个体从子服务器迁移到主服务器。迁移量的多少通常由两个参数来确定:迁移代频和迁移率。

基于协同优化方法的协同演化系统中的系统级优化器和子系统优化器都采用基于遗传算法的协同演化策略[2]进行全局演化寻优和局部演化寻优,系统级优化器和多个子优化器同处于一个分布式的局域网环境下,共同协调与协作完成整个遗传演化任务。通过系统级优化和子系统优化之间的多次遗传演化迭代,最终找到一个使各子系统间能够达成一

收稿日期:2006-07-17
作者简介:陈秋莲(1974-),女,广西扶绥人,讲师,硕士,主要从事遗传优化设计、CAD研究工作。
*本文得到国家自然科学基金项目(59868001)及广西大学科研基金(X061001)的资助。

致的最优设计[3]。

2 系统功能

分布式协同演化系统直接利用局域网络环境条件，在客户机/多服务器模式下完成。其中服务器分为主服务器和子服务器。主服务器对应于系统级优化器，它负责将长基因群体的分解，分配任务给子服务器，并对各子服务器返回结果进行综合，演化，产生新的群体再分解和分配。子服务器作为子优化器，负责分解后的短基因群体在局部空间上的演化；客户端运行输入、输出模块程序，为用户提供输入、输出界面。各部分的功能如图1所示。

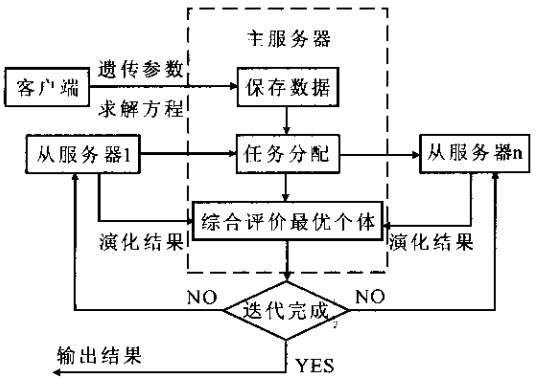

图1 数据流程

3 系统实现

我们实现的协同演化系统采用的遗传算法是实数编码，轮盘赌选择，算术杂交，均匀变异算法[4]。此外为了更好地处理从客户端输入的优化方程和约束条件，增加了一个表达式分析函数，采用算符优先算法和词法分析完成方程的分析。再使用C#中的Socket来实现信息的传递。通过Socket类来实现Berkeley套接字接口，以Socket类创建Internet传输服务的托管版本。一旦创建了Socket，该Socket可通过Bind方法绑定到特定的终结点，并通过Connect方法绑定到该终结点所建立的连接上，使用Send或SendTo方法将数据发送到Socket；使用Receive或ReceiveFrom方法从Socket中读取数据。当使用完Socket后，可用Shutdown方法禁用Socket，并用Close方法关闭Socket。为了方便地对Socket进行操作，我们利用NetworkStream，通过其提供的方法StreamReader和StreamWriter进行字符的输入、输出。

3.1 客户端实现

客户端主要完成输入、输出、连接服务器及关闭连接的操作。需要提供的输入参数有：服务器端口、优化方程、约束条件、方程中的变量个数、优化精度。设置输入参数后，就可以连接主服务器，由主服务器分配任务完成分布式协同演化算法。

客户端通过NetworkStream的Write方法来实现数据的传输，NetworkStream实现通过网络套接字发送和接收数据的标准.NET框架流机制。NetworkStream支持对网络数据流的同步和异步访问。这样可将变量个数，演化方程、演化精度和约束条件分别通过Socket发送到主服务器端。发送数据的关键代码如下：

```
Byte[] sendByte= new Byte[1024];
richTextBox2.AppendText(aimString);
NetworkStream netStream=new
NetworkStream(connectSock);
sendByte=System.Text.Encoding.
BigEndianUnicode.GetBytes
(aimString.ToCharArray());
        netStream.Write (sendByte, 0,
sendByte.Length);
netStream.Flush();
```

客户端在发送数据以后，就一直处于等待状态，等待服务器返回来的演化结果。当结果送来后，以文本文件方式保存在当前目录下，这时可关闭连接。

3.2 服务器端实现

服务器端由主服务器和从服务器组成，它们的工作流程分别为：(1)主服务器的工作流程为①主服务器启动，等待客户机及从服务器连接；②在客户机连接后，读取用户输入信息。等待各个从服务器均连接；③进行任务分配，发出请求，启动从服务器进行协同演化；④主服务器收到从服务器的演化结果后整合，计算方案的适应值。判断适应值是否满足条件，如果满足则输出结果。如果不满足条件，则问题空间演化，再重复执行③。(2)从服务器的工作流程为①从服务器连接到主服务器，等待主服务器的调用请求；②主服务器发出请求后，接受演化优化方程和其他参数取值；③在本区域内进行遗传演化；④将演化结果返回给主服务器。

服务器实现使用NetworkStream实现通过网络套接字接收数据，首先建立一个IPEndPoint类，IPEndPoint类包含应用程序连接到主机上的服务所需的主机和端口信息。通过组合服务的主机IP地址和端口号，IPEndPoint类形成到服务的连接点。然后创建SOCKET捆绑，监听，确定连接之后就通过NetworkStream实现通过网络套接字接收数

据。其中接收数据的关键代码如下:

```
MyServer = new IPEndPoint (myIP, Int32. Parse (textBox2. Text));
sockclient=new Socket(AddressFamily. InterNetwork, SocketType. Stream, ProtocolType. Tcp);
sockclient. Bind(MyServer);
sockclient. Listen(50);
textBox3. AppendText ("主机" + textBox1. Text +"端口" +textBox2. Text+"开始监听 . . . . . \r\n");
while (true) {
  accSockclient=sockclient. Accept();
  if (accSockclient. Connected) {
    textBox3. AppendText("与客户建立连接");
    Thread threadclient=new Thread(new ThreadStart(roundclient));
    threadclient. Start();
  }
}
private void roundclient()
{
while(true) {
 Byte[] Rec=new Byte[1024];
 NetworkStream netStreamc = new workStream (accSockclient);
 netStreamc. Read(Rec,0,Rec. Length);
  string Recmessage = System. Text. Encoding. BigEndianUnicode. GetString(Rec);
cstring=Recmessage;
richTextBox1. AppendText (Recmessage + "\r\n");
}
```

4 应用分析

主服务器与从服务器及主服务器与客户端之间的数据通信是通过 SOCKET 连接,由 NetworkStream 类传送和接收。发送前先把各部分数据打包成 String 格式,数据间用@分开。客户端的数据传输格式为:"0"+@参数个数+@整数精度+@小数精度+@方程#$+@各个参数约束条件。主从服务器间的数据传输格式为:标志位+@基因 1+@基因 2+@基因 3+@基因 4+@基因 5+@结果。接收端则按标志位把各个数据从数据包中分离出来。

例如:求函数 y=(x1 * x1)-(x1 * x2) +x3,在变量 x1∈(1.0 150.12),x2∈(25.0 75.12),x3∈(10.0 50.12)范围的最大值。取种群规模和最优个体群规模分别为 50 和 5,杂交率为 0.8,变异率为 0.15,迁移代频是每 10 代迁移一次,迁移率是 0.01%。进行协同演化寻优测试。主服务器以函数本身作为目标函数,约束条件就是三个变量的取值范围。将演化任务分成三部分:(x1 * x1),-(x1 * x2),x3,将三个子函数及其相应的变量取值范围传送给从服务器,由从服务器完成子函数在其取值区域内的遗传优化。优化所得的前 5 个最优结果返回给主服务器进行整合和总体评估,由评估结果来调整从服务器的下一演化过程。

实例结果表明(见图 2),随着代数的增加,最佳适应值逐步增长,平均适应值则有成波浪状的上升趋势。当迭代 100 次后,其结果为:x1 = 149.971, x2 = 25.000, x3 = 50.080,最佳适应值 = 18792.073。

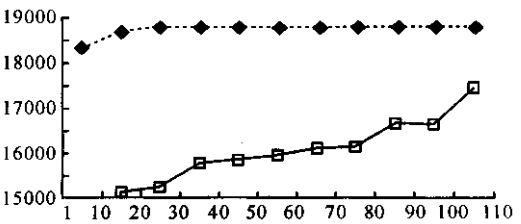

图 2 协同演化优化结果

—◆—:最佳适应值; —□—:平均适应值

说明了协同优化方法将原问题按结构进行分解,可将高维、复杂耦合的优化设计问题按结构分解为若干个低维、相对独立的子问题,缩短染色体的长度,加速了算法的收敛速度。在相同条件下,分布式协同演化计算所得的解质量要比单机遗传算法要好。如果使用更复杂和计算量更大的目标函数,分布式遗传算法在演化时间上更体现出它的优越性。

参考文献:

[1] 潘正君,康立山,陈毓屏. 演化计算[M]. 北京:清华大学出版社,1994.

[2] MAHER M L,POON J. Modeling design exploration as co-evolution[J]. Microcomputers in Civil Engineering, 1996,11(3):195-209.

[3] 陈秋莲,李陶深,吴恒,等. 基于遗传算法的基坑支护协同演化处理模型[J]. 计算机应用,2004,24(10):139-140.

[4] 余雄庆,薛飞,穆雪峰,等. 用遗传算法提高协同优化方法的可靠性[J]. 中国机械工程,2003,14(21):1808-1811.

(责任编辑:凌汉恩 邓大玉)