

文章编号: 1001-0920(2003)05-0630-03

## 基于 GA 的系统分解协调算法 Agent 实现

于江涛, 钱积新

(浙江大学 系统工程研究所, 浙江 杭州 310027)

**摘 要:** 为解决实际工程问题, 将 Agent 的设计思想引入分解协调算法, 实现了大系统的分解和生产单元之间的协作, 并采用 GA 算法进行参数协调。改进后的分解协调算法具有较好的适用性。

**关键词:** 分解协调; Agent; 优化; 遗传算法

**中图分类号:** TP18      **文献标识码:** A

## GA based agent implementation of system decomposition and coordination algorithm

YU Jiang-tao, QIAN Ji-xin

(Institute of System Engineering, Zhejiang University, Hangzhou 310027, China)

**Abstract:** The theory of agent is introduced to optimization model design. A large scale system is decomposed into several sub-systems which are described by agent models. Their coordination values are modified by a genetic algorithm. Optimization of overall system is implemented by sub-system agents cooperation. A case study shows that this modified algorithm meets the real-time requirements of chemical process industry.

**Key words:** Decomposition and coordination; Agent; Optimization, Genetic algorithm

### 1 引 言

大规模复杂系统, 如石油化工企业、冶金工业等, 因工艺流程复杂, 系统参数的维数较多, 给系统模型的建立、分析和优化计算带来很多困难。不论是执行效率, 还是优化的鲁棒性, 系统的整体优化都存在不少问题, 而且随着系统规模的增大, 参数的维数呈爆炸性增长也是不可避免的。因此, 将大系统分解为规模较小的系统是一种必然趋势。

分解协调算法是复杂系统优化的一种良好手段, 但从实际工程应用角度看, 该算法仍存在很大的局限性。本文将 Agent 的设计思想引入分解协调算法, 实现了大系统的分解和生产单元之间的协作, 并采用 GA 算法进行参数协调, 使改进后的分解协调算法具有较好的适用性。

### 2 基于 GA 的分解协调算法 Agent 实现

#### 2.1 系统分解与协调

20 世纪 60 年代中期, Aris 等人首先提出将大系统分解为多个子系统, 并在不同等级上对子系统进行优化和协调计算, 以求得大系统的最优解<sup>[1]</sup>。其中最常用的是两级划分, 首先将系统划分为相互分离但相互之间有一定联系的  $N$  个子系统, 若系统的目标函数  $J$  满足“加性可分离”的形式, 则可表示为各子系统的性能指标之和, 于是整个系统的 Lagrange 函数便分解为  $N$  个子系统的较为简单的 Lagrange 函数, 从而得到各子系统的优化模型。Lagrange 乘法优化求解结果为协调参数  $\beta$  的函数  $x_i(\beta)$ ,  $y_i(\beta)$ ,  $u_i(\beta)$ , 根据子系统之间联接关系, 采用梯度法调整

收稿日期: 2002-03-19; 修回日期: 2002-06-13。

作者简介: 于江涛(1976—), 男, 山东临沂人, 博士生, 从事多智能体系统、智能决策与系统优化的研究; 钱积新(1939—), 男, 浙江嵊县人, 教授, 博士生导师, 从事过程控制、决策与系统优化等研究。

协调参数  $\beta$  值, 将调整后的  $\beta$  送回各子系统, 再次执行优化和对比调整, 重复迭代直至满足整体最优。

虽然分解协调的概念清楚, 条理性好, 但对于大多数系统, 只有当协调参数  $\beta$  的迭代初值选在最优化附近时才有可能收敛, 因此从工程应用角度讲并不实用。其他形式的分解和协调, 如模型协调法, 只有当输出变量  $Y_i$  的维数  $m_{y_i}$  和输入变量  $X_i$  的维数  $m_{x_i}$  满足  $m_{y_i} = m_{x_i}$  时才可用, 所以也有很大的局限性。从近期国内外研究发展趋势可以看出, 单纯采用分解协调的算法已经逐渐被其他优化方法所替代, 但分解协调的基本思想仍广泛应用于各种优化计算<sup>[2]</sup>。

### 2.2 Agent 的引入

系统优化的关键在于各子系统单元之间协调方法的有效性。源于分布式人工智能 (DAI) 领域的 Agent 思想, 在群体协作方面已取得很多研究成果, 并在许多领域中得到广泛的应用。

Agent 被定义为具有自身和环境知识、独立运行实现特定功能、能够相互协作达到共同目标的功能个体。Agent 的基本特征是自主性、反应性、社会性和适应性, 能以特定的方式响应特定的环境。作为独立的实体, Agent 的行为应是完全主动和自发的。1987 年, Bratman 提出了 Agent 的 BDI 模型, 将 Agent 用 Belief, Desire 和 Intention 三种精神状态来描述, 分别代表 Agent 所具备的知识、能力和目标。Agent 的能力建立在所具备的知识基础上, 知识的完备性直接影响到 Agent 的能力和目标的实现<sup>[3]</sup>。分解子系统可用 BDI 模型描述如下:

**知识:** 个体装置和生产环境的状态, 包括所有的输入、输出变量, 系统的公共变量, 以及所有的约束条件;

**能力:** 子系统自身优化求解运算, 与生产环境交互, 与其他生产单元协作;

**目标:** 在保证自身局部优化的同时, 实现系统整体优化。

在这个模型中, 单个装置的优化操作在 Agent 内部实现, 不受外界干扰。每个 Agent 单独执行自身的优化操作, 其结果通过 Agent 的外部接口传递给其他 Agent, 经过信息交换和结果比较, 各自修正优化参数, 返回 Agent 再次进行优化。经过多次分布式的同步迭代计算, 最终达到满意的整体优化结果。

### 2.3 GA 的适用性

在传统的分解协调算法中, 协调参数的修正采用负梯度法或共轭梯度法。对于单个参数变量来说,

梯度法的收敛方向一致, 调整步长可变, 收敛精度高, 是一种很好的修正方法。但对于多个参数或参数向量的修正, 特别在维数较大且具有耦合关系的情况下, 梯度法的收敛性能会变得很差, 容易出现振荡, 这也是传统分解协调算法无法实用的重要原因所在。针对上述情况, 为避免振荡并达到满意的收敛结果, 本文对子系统 Agent 之间的协调参数  $\beta$  采用 GA 算法进行修正。

参数  $\beta$  没有实际工程意义, 因而难以确定其初值。GA 的初始种群是随机选取的, 可以覆盖很大范围。当  $\beta$  维数增长时, GA 的性能不会受很大影响。从搜索结果的精度看, 虽然 GA 的收敛结果与整体优化结果相比要差一些, 但对于大多数工业系统而言, 在整体优化结果无法达到的情况下, 采用次优结果也是令人满意的。GA 属于带有指导性的随机搜索算法, 收敛速度不占优势, 但对于化工流程来说, 得到最优或可行解是首要的, 优化的实时性要求并不高, GA 的收敛速度对于以小时计的优化周期完全能够满足要求。

## 3 应用实例

具有 3 个生产单元 (如精馏塔) 的工业流程系统如图 1 所示。

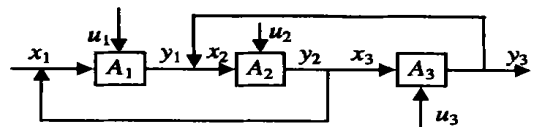


图 1 工业流程系统模型

系统的状态方程及联接关系如下

$$\begin{cases} y_1 = x_1 + u_1 - 2 \\ y_2 = 0.5x_2 + 0.5u_2 - 2, \\ y_3 = x_3 + 2u_3 - 1 \end{cases} \quad \begin{cases} x_1 = y_2 \\ x_2 = y_1 + y_3 \\ x_3 = y_2 \end{cases}$$

求目标函数  $J = \frac{1}{2}(x_1^2 + u_1^2 + x_2^2 + u_2^2 + x_3^2 + u_3^2)$  为最小时的  $u_1, u_2, u_3$

### 3.1 Agent 分解模型

首先按生产单元进行划分, 将系统的 Lagrange 函数分解为 3 个生产单元的 Lagrange 子函数之和, 得到各子系统的优化模型。用 Lagrange 乘子法求得子系统局部最优解  $x_i, u_i, y_i (i = 1, 2, 3)$  均为  $\beta$  的函数。个体单元优化完成后, 再按图 2 所示的 Agent 功能实现模型进行信息交换和整体协调。

### 3.2 协调参数的 GA 修正

#### 3.2.1 初始种群的选取

由于协调参数  $\beta$  没有实际工程意义, 因此其初

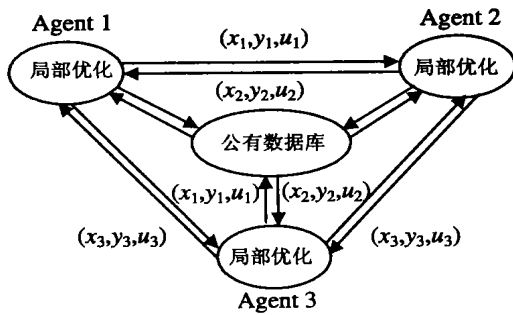


图2 Agent功能实现模型

值只能凭工程经验选取或随机选取。在本例中, 如果没有工程经验取值范围, 初始种群的选取范围可以适当扩大, 以保证覆盖全局最优解或至少次优解, 而这一点是梯度法所无法达到的。这里不妨选取  $\epsilon$ ,  $\epsilon_2, \dots, \epsilon_{100}$ ,  $\epsilon = \{\beta_{i,1}, \beta_{i,2}, \beta_{i,3}\}$  表示个体数目  $N = 100$  的初始种群, 其中  $\beta_{i,1}, \beta_{i,2}, \beta_{i,3}$  为大小在  $\pm 1\ 000$  之间的随机浮点数。

分解协调算法优化的目标是在各子系统实现局部优化的基础上, 尽可能满足各子系统之间的关联约束。在本例中, 根据系统的关联约束, 确定如下的适应度函数

$$\min f = (y_2 - x_1)^2 + (y_1 + y_3 - x_2)^2 + (y_2 - x_3)^2$$

### 3.2.2 交叉和变异

GA 搜索的目标是确定使适应度函数  $f$  值最小的协调参数向量  $\epsilon$ 。在每次迭代过程中, 按  $f$  值从小到大将 100 个  $\epsilon$  排序, 选择  $f$  值较小的前 50 个个体保留至下一代。然后由这 50 个父代进行交叉操作, 成对选取  $\epsilon = \{\beta_{i,1}, \beta_{i,2}, \beta_{i,3}\}$  和  $\epsilon = \{\beta_{i,1}, \beta_{i,2}, \beta_{i,3}\}$  在随机位置进行交叉互换, 共产生 50 个新个体。新个体经过小概率变异后加入父代, 重新组成数目  $N = 100$  的种群, 进入下一次进化迭代, 直到  $f$  值满足精度要求 ( $< 0.01$ ) 为止。本例中, 考虑到种群数目和搜索范围的大小, 经过多次仿真实验, 确定变异概率在  $0.2\% \sim 0.4\%$  之间为宜。由于初始种群是随机产生的, 所以每次优化过程的收敛速度和收敛结果均有不同的差异, 但误差都在可以接受的合理范围内。

### 3.3 仿真结果

4组优化过程的仿真结果由表1给出。虽然每

次仿真的初始值是不同的随机值, 但经过 GA 迭代总能很快地收敛到最优值附近, 且仿真结果均在合理的误差范围内。

表1 仿真优化结果

仿真过程	迭代次数	优化结果		
		$u_1$	$u_2$	$u_3$
a	1 062	1.078	1.765	2.122
b	2 010	1.077	1.835	2.209
c	1 213	1.066	1.844	2.140
d	748	1.040	1.736	2.107

在奔腾 133 计算机上运行仿真程序, 收敛时间一般为几秒到几十秒, 所以本算法能满足生产系统的实时性要求。GA 搜索的特点是能以很快的速度收敛到最优点附近, 对于化工生产流程级的优化来说, 给出一个最优区域是可以令人满意的。

## 4 结 语

Agent 的思想和方法在许多领域得到广泛的应用, 而在工业控制系统中引入 Agent 来解决常规控制和优化方法难以解决的问题则是一个新的尝试<sup>[4]</sup>。本文采用 Agent 的模型方法, 结合现代优化算法对常规的系统分解协调算法进行有效的改进, 弥补了纯数学方法的缺陷, 增强了算法的实用性。随着大工业系统由集中控制向分散控制的发展, 基于 Agent 的控制和优化方法将会越来越多地应用于工业系统。

### 参考文献 (References):

- [1] Shen Jingzhu. *Process System Optimization* [M]. Beijing: Tsinghua University Press, 1994.
- [2] Kwang-Hyun Cho, Jong-Tae Lim. Multiagent supervisory control for antifault propagation in serial production systems [J]. *IEEE Trans on Industrial Electronics*, 2001, 48(2): 460-466.
- [3] Vancza J, Markus A. An agent model for incentive-based production scheduling [J]. *Computers in Industry*, 2000, 43: 173-187.
- [4] Stefan Bussmann, Klaus Schild, Daimler Chrysler. Self-organizing manufacturing control: An industrial application of agent technology [A]. *Research & Technology 3A ItrM oabit 96A* [C]. Berlin, 1996: 10559.