

一种基于多 Agent 协同的准并行遗传算法

江 瑞¹, 罗予频¹, 胡东成¹, 司徒国业²

(1. 清华大学自动化系, 北京 100084; 2. 香港科技大学物理系, 香港九龙清水湾)

摘 要: 提出了一种基于多 Agent 协同操作的准并行遗传算法结构. 该算法由若干运行简单遗传算法的计算单元组成, 每个单元也就是独立的计算 Agent. 算法依照资源分配向量为各单元分配不同的计算资源, 并根据个体迁移矩阵驱动它们进行个体交换. 从多 Agent 系统的观点看, 资源的分配体现了算法对各 Agent 的协调, 个体的迁移则体现了 Agent 之间的协作. 该算法很容易在串行计算机上实现, 此时各个计算单元具有微观上串行、宏观上并行的准并行关系. 对二维准并行算法动态性能的分析表明: 由于统筹考虑了各计算单元间的协同关系, 算法能够更充分有效地利用有限的计算资源, 在解决不同的优化问题时表现出了很高的性能.

关键词: 遗传算法; 准并行; 多 Agent 计算系统; 协调; 协作

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2002) 10-1490-06

A Multi-Agent Cooperating Approach to Quasi-Parallel Genetic Algorithms

JIANG Rui¹, LUO Yu-pin¹, HU Dong-cheng¹, Szeto Kwok-Yip²

(1. Department of Automation, Tsinghua University, Beijing 100084, China;

2. Department of Physics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China)

Abstract: This paper describes a kind of parallel genetic algorithm that is based on the idea of multi-agent cooperation. The algorithm consists of several computing units, in each of which a simple genetic algorithm is maintained, thus each computing unit can be regarded as an independent autonomous agent. The algorithm allocates computing resources to each unit according to the resource-allocating vector and carries through exchange of individuals between units according to the individual-migrating matrix. From the viewpoint of multi-agent system, the allocation of computing resource represents the coordination between agents, while the migration of individuals represents the collaboration between them. The algorithm can be implemented easily in a serial computer and it has the quasi-parallel feature in this case. The analyses to such a quasi-parallel genetic algorithm in two dimension show that since the cooperation between computing agents is taken into account in the algorithm, the computing resources can be utilized in a more effective way and thus better performances are presented when the algorithm deals with different kinds of optimizing problems.

Key words: genetic algorithms; quasi-parallel; multi-agent computing system; coordination; collaboration

1 引言

遗传算法 (Genetic Algorithm, GA) 是一类基于遗传机制和自然选择原理的随机搜索算法. 通过使用简单的编码技术来表现复杂的现象并使用多个个体构成的种群对解空间并行地搜索, 遗传算法能够在不要求函数连续、可导、单峰等性质的情况下实用地找到问题的近似全局最优解^[1,2]. 这些优点使得遗传算法在机器学习、模式识别、图像处理、时间序列预测等领域获得了越来越广泛的应用^[3~6].

随着计算机网络, 特别是 Internet 技术的日益成熟和普及, 对空间上分布和时间上并行的计算方法的研究早已成为一个热点^[7~10]. 遗传算法的种群由多个个体组成, 这使得在两个算法间进行个体交换很容易实现. 因此, 并行地运行多个

遗传算法并让它们之间进行个体交换以组成粗粒度并行遗传算法, 已是一种广为接受的遗传算法并行策略^[2,7,8]. 这方面的研究集中在子算法在逻辑上的拓扑结构和它们在单位时间内交换个体的数量. 研究成果表明, 不同的拓扑结构将直接影响并行算法的效率; 而个体的交换能够给种群带来新的基因, 这对于提高算法的性能很有帮助^[10]. 从多智能体系统 (Multi-Agent System, MAS) 的观点来看^[11,12], 子算法是具有独立计算能力的 Agent, 它们之间的个体迁移实质上是 Agent 之间的协作 (Collaboration). 这种协作可以使整个系统的性能得到提升. 然而, 这些研究总是假定各 Agent 使用相同的计算资源并具有相等的计算能力, 并没有考虑当系统资源不平均分配而使各 Agent 具有不相等计算能力的情况下并行算法的特性. 与

其它随机搜索算法类似,遗传算法对同一问题的多次求解结果往往表现出相当大的分散性和不确定性,这通常用算法得出优化结果时运行时间的概率分布来表征^[13]. Huberman 把随机算法运行时间的数学期望和方差与经济学中效用 (Utility) 和风险 (Risk) 的概念进行类比. 基于效用和风险并存这一基本的经济学原理,他使用一种通用的方法在串行计算机上组合随机算法的两个运行实例,以获得算法运行时间数学期望和方差之间的折衷^[14]. 从多智能体系统的观点看,算法运行实例就是独立的 Agent. 这种算法组合方法实质上是通过协调 (Coordination) 各 Agent 所使用的计算资源来获得系统性能的提升.

基于以上认识,本文提出了一种基于对计算资源进行再分配的准并行遗传算法结构. 该算法在串行计算机上实现时具有微观上串行宏观上并行的准并行关系. 这一算法驱动多个具有相同结构和参数的子算法,依照计算资源分配向量为它们分配不同的计算资源,并根据个体迁移矩阵执行子算法间的个体迁移. 子算法实质上就是具有独立计算能力的 Agent,计算资源的分配体现了对它们的协调;个体的迁移则体现了它们之间的协作. 综合地看,该算法统筹考虑了子算法之间的协同 (Cooperation) 关系,因而具有更大的灵活性,在解决不同问题时表现出很高的性能.

2 算法结构及准并行实现

2.1 算法结构

本文提出的准并行遗传算法 (Quasi-Parallel Genetic Algorithm, QPGA) 由若干独立计算单元组成,每个单元运行一个简单遗传算法. 从 MAS 的观点看,每个计算单元都是独立的 Agent,整个算法就是一个多 Agent 计算系统,由如下的 5 元组定义:

$$QPGA = (M, SubGA, \quad, \quad, T)$$

这里, M 是子算法个数, $SubGA$ 是子遗传算法, \quad 是计算资源分配向量, \quad 是个体迁移矩阵, T 是算法终止条件,它们的意义如下.

算法维数:即准并行算法 QPGA 中子算法的个数 M .

子算法:子算法 $SubGA$ 可以是任何一种简单遗传算法 (SGA),如最优保持遗传算法^[1],稳态遗传算法^[2]. 子算法用 $A_i, i = 1, 2, \dots, M$ 表示,它们在结构和参数上是完全相同的.

计算资源分配向量:设总的计算资源为 R ,由 M 个子算法 $A_i, i = 1, 2, \dots, M$ 共享, A_i 在单位时间内分配到的计算资源为 R_i ,令 $\quad = R_i / R$,易知 $\quad \geq 0$, $\sum_{i=1}^M \quad = 1, i = 1, 2, \dots, M$. 计算资源分配方案可以用如下的计算资源分配向量表示:

$$\quad = (\quad_1, \quad_2, \dots, \quad_M)$$

由于子算法结构和参数上的一致性,分配到的计算资源将直接决定其计算能力. 在一般情况下, $\quad_i \neq \quad_j, \forall i \neq j$, 此时算法具有非对称的性质.

个体迁移矩阵:两子算法 A_i 和 $A_j, i, j = 1, 2, \dots, M$ 之间交换个体的操作 $A_i \leftrightarrow A_j$ 表示 ($i = j$ 时不进行交流). 单位时间内迁移的个体数记为 $\quad(A_i \leftrightarrow A_j)$ 或简写成 \quad_{ij} ,它与子算法种群内个体数量的比值记为 \quad_{ij} . 这样子算法间的个体迁移方

案可以用如下的 M 阶方阵表示.

$$= \begin{pmatrix} \quad_{11} & \quad_{12} & \dots & \quad_{1M} \\ \quad_{21} & \quad_{22} & \dots & \quad_{2M} \\ \dots & \dots & \ddots & \dots \\ \quad_{M1} & \quad_{M2} & \dots & \quad_{MM} \end{pmatrix}$$

显然, \quad 是一个对称非负的随机矩阵,即 $\quad_{ij} = \quad_{ji}, \quad_{ij} \geq 0$ 及 $\sum_{j=1}^M \quad_{ij} = 1$ 对任意 $i, j = 1, 2, \dots, M$ 成立.

停止准则:用以决定算法何时停止运行. 通常使用最大误差限准则或最大运行时间准则.

整个算法的并行执行过程用图 1 表示.

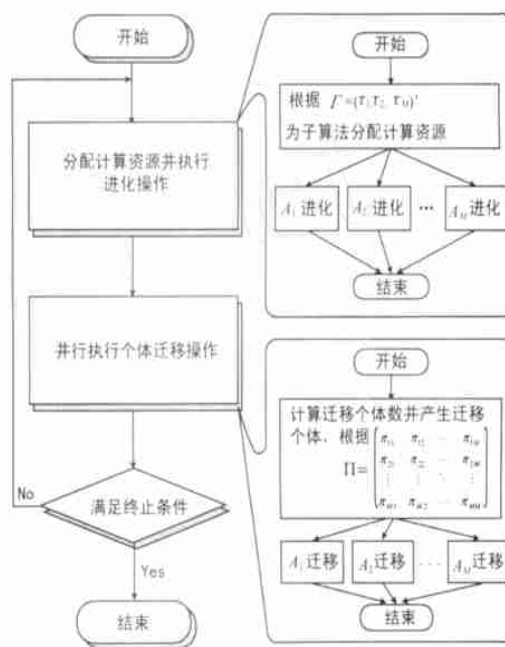


图 1 QPGA 执行流程

准并行遗传算法中存在进化和个体迁移两类操作. 在依照 \quad 为每个子算法分配计算资源后, QPGA 让子算法并行地进化, 之后根据 \quad 执行个体迁移并检测停止准则以确定算法是否结束. 从 MAS 的观点看, 为每个 Agent 分配不同的计算资源意味着算法对各 Agent 所使用的资源进行协调; 而在 Agent 之间交换个体意味着它们之间存在协作. 综合起来看, 这一算法结构体现了对独立计算 Agent 的协同.

2.2 准并行实现及个体迁移拓扑结构

QPGA 可以在串行计算机上实现. 不失一般性, 设计算资源仅限于处理机时间, 先给出如下定义.

定义 1 处理机单位时间. 子算法的一致性决定了它们进行一次进化所需处理机时间是相等的. 这一时间称为处理机单位时间, 它是 QPGA 分配处理机时间的最小单位.

子算法 $A_i, i = 1, 2, \dots, M$ 进行一次进化迭代的操作用事件 $E(A_i)$ 表示, 这一事件在处理机单位时间内发生的概率记为 $Pr[E(A_i)]$ 或简写成 p_i , 则 $p_i = \quad_i, i = 1, 2, \dots, M$. 当处理机时间完全可分配时, 使用如下策略为子算法分配处理机时间.

Step 1: 计算 $i = \sum_{k=1}^i p_k, i = 1, 2, \dots, M$. 显然, 这里有 $0 \leq i_1 \leq i_2 \leq \dots \leq i_M = 1$ 成立;

Step 2: 产生一个在 $(0, 1)$ 内均匀分布的随机数 r , 令 $m = \min(i, r), \forall i > 0, i = 1, 2, \dots, M$;

Step 3: 对 A_m 执行一次进化操作.

在该策略下, 各子算法本质上并不是并行的, 但它们所使用的处理机时间的数学期望与并行时是一样的 ($p_i = i, i = 1, 2, \dots, M$), 而且由于它们对处理机的使用是轮换进行的, 因此它们之间是一种微观上串行和宏观上并行的关系, 本文称之为准并行 (Quasi-Parallel) 关系, 此时的并行算法称为准并行遗传算法 (Quasi-Parallel Genetic Algorithm, QPGA).

子算法在逻辑上被安排成一定的拓扑结构, 以决定 QPGA 的个体迁移矩阵. 例如, 当子算法组成一维环状结构^[10]时, 个体迁移仅发生在 A_i 和 A_{i+1} 以及 A_i 和 A_{i-1} 之间 (注意 $A_{(M+1)} = A_1$ 及 $A_0 = A_M, i = 1, 2, \dots, M$). 令迁移的个体数与种群内个体数的比值为 α , 则此时的个体迁移矩阵可以简化为:

$$= \begin{pmatrix} 1-\alpha & \alpha & 0 & \dots & 0 \\ 0 & 1-\alpha & \alpha & \dots & 0 \\ \dots & \dots & \dots & \ddots & \dots \\ 0 & \dots & \dots & 1-\alpha & \alpha \end{pmatrix} \quad (1)$$

这是一个稀疏矩阵, 非零元素仅出现在 $i, i \pm 1$ 及 $i, i \pm M$ 处. 这里, $i, i \pm M = 0$ 及 $0 = M, i = 1, 2, \dots, M$. 显然, $0 \leq \alpha \leq 0.5$ 成立.

3 准并行算法的动态性能

本节将首先介绍基于算法运行概率分布和统计规律的算法评价体系; 在此基础上分析二维 QPGA 的动态性能, 即算法在不同的计算资源分配方案和不同的个体迁移强度下的性能特点, 最后给出确定二维 QPGA 参数的一般方法.

3.1 算法评价体系

虽然运行时间的算术平均值被广泛使用来衡量随机算法的性能, 但它是不全面的, 因为它不能描述随机搜索算法运行时的不确定性^[14]. 本文因此提出了一组基于算法运行时间概率分布的评价指标, 包括平均成功率 R_{AS} , 平均首次代数 N_{AFP} , 平均首次代数方差 D_{AFP} 和平均首次代数信息熵 S_{AFP} .

定义 2 在给定最大迭代次数 N_{max} 下算法的一次运行称为一次试验. 如果在一次试验的第 n 次迭代中满足:

$$\begin{cases} |F^* - F_{Best}^n| < \epsilon \cdot F^*; F^* > 0 \\ |F_{Best}^n| < \epsilon; F^* = 0 \end{cases} \quad (2)$$

则称该次试验成功, 否则称该次试验失败. 对于一次成功的试验, 算法首次满足式 (2) 的进化代数 n 称为算法在本次试验中的首次代数 N_{FP} . 这里, F^* 表示全局最优解, F_{Best}^n 表示第 n 进化代的最优解.

定义 3 设对算法进行了 M 次独立试验, 其中 M_s 次成功, 则称 $R_{AS} = M_s / M$ 为算法在多次试验中的平均成功率. 平均成功率 R_{AS} 是对算法在规定时间内找到问题全局最优解的概率 P_s 的估计.

考虑连续时间情况下算法的性能. 以算法运行时间为横坐标, 以特定时间点算法计数值为纵坐标, 作算法在特定时间点得出问题解的分布直方图 $H(T)$, 进而求出算法运行时间的概率分布 $P(T) = H(T) / \int_0^\infty H(T) dT$.

定义 4 平均首次代数 N_{AFP} , 平均首次代数方差 D_{AFP} 和平均首次代数信息熵 S_{AFP} 定义为

$$N_{AFP} = \int_0^\infty P(T) \cdot T dT \quad (3)$$

$$D_{AFP} = \int_0^\infty P(T) \cdot (T - N_{AFP})^2 dT \quad (4)$$

$$S_{AFP} = - \int_0^\infty P(T) \log[P(T)] dT \quad (5)$$

N_{AFP} 即算法求解时间的数学期望值. D_{AFP} 是它的方差. S_{AFP} 用来衡量算法运行的分散程度.

遗传算法是一个迭代过程, 每次迭代所使用的处理机时间是一个常量, 因此使用基于迭代次数的指标来衡量其性能是更为适当的. 为此, 作算法在特定迭代次数下求出问题解的分布直方图 $H(N)$, 进而得出算法在特定迭代次数下求出问题解的概率分布 $P(N) = H(N) / \sum_{N=0}^\infty H(N)$.

定义 5 平均首次代数 N_{AFP} , 平均首次代数方差 D_{AFP} 和平均首次代数信息熵 S_{AFP} 定义为

$$N_{AFP} = \sum_{N=0}^\infty P(N) \cdot N \quad (6)$$

$$D_{AFP} = \sum_{N=0}^\infty P(N) \cdot (N - N_{AFP})^2 \quad (7)$$

$$S_{AFP} = - \sum_{N=0}^\infty P(N) \log[P(N)] \quad (8)$$

一般取 $N = 1$ 即可. N_{AFP} 即算法求解迭代次数的数学期望值. D_{AFP} 是它的方差. S_{AFP} 用来衡量算法运行的分散程度.

3.2 二维算法的实现

以下将通过二维 QPGA 对标准测试函数 Schaffer 函数^[15]寻优时的特点分析本文提出的算法在二维情况下的动态性能. Schaffer 函数定义为

$$S(x) = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001 \cdot (x_1^2 + x_2^2)]^2}$$

其中, $-100 \leq x_{1,2} \leq 100$. 它的全局极大值点为 $(0, 0)$, 而在距其大约 10^4 的范围内有无限多个次全局极大点. 该函数剧烈振荡, 而且其全局最优点被次全局最优点包围, 因而对它的寻优很困难.

子算法: 使用带重新启动策略的稳态遗传算法^[2]. 该算法在每一进化代通过某种繁殖策略从 $P(t)$ 中选择 N 个个体形成交配池, 对其中个体施以交叉和变异操作得到种群 $Q(t)$ 后, 在 $P(t)$ 和 $Q(t)$ 共 $2N$ 个个体中通过某种选择策略选出 N 个形成 $P(t+1)$.

1). 如果在指定迭代次数内算法最优适应值没有变化, 则用随

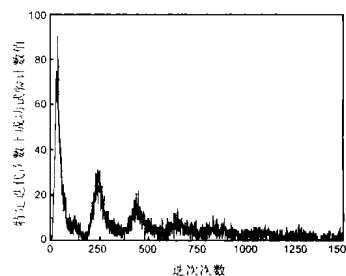


图 2 带重新启动策略的稳态遗传算法首次代数分布直方图

机产生的种群代替 $P(t)$. 该算法在求解函数优化问题时首次时间分布的典型直方图见图 2.

处理机因子: $M=2$ 时, $\alpha = (\alpha_1, \alpha_2)$. 考虑到 $\alpha_1 + \alpha_2 = 1$, 令 $\alpha = \alpha_1$, 则两个子算法分配到的处理机时间为 α 和 $1 - \alpha$. 称为处理机因子.

个体迁移因子: $M=2$ 时, 令个体迁移因子 $\beta = 2$, 则个体迁移矩阵为

$$= \begin{pmatrix} 1 - \beta & \beta \\ \beta & 1 - \beta \end{pmatrix}$$

迁移个体选取策略: 即如何从子算法中选择个体进行迁移. 可以使用随机选择, 最优选择, Roulette Wheel 选择等^[1,2]. 本文使用 Roulette Wheel 选择.

停止准则: 给定最大误差限 ϵ 和最大迭代次数 N , 如果算法满足式 (2) 或是迭代次数大于 N , 则停止算法的运行.

对称性: $M=2$ 时, 在给定子算法和停止准则下, QPGA 是处理机因子 α 和个体迁移因子 β 的函数, 令 $QPGA = QPGA(\alpha, \beta)$, 此时算法有如下的对称性:

$$QPGA(\alpha, \beta) = QPGA(1 - \alpha, \beta) \quad (9)$$

$$QPGA(\alpha, 1) = QPGA(0.5, 0) \quad (10)$$

前者成立是由于两子算法是完全相同的; 后者成立说明如果两个子算法在单位时间内交换所有的个体, 就等价于两个子算法均分处理机时间.

3.3 平均成功率及平均首次代数信息熵随处理机因子和个体迁移因子变化的关系

对于算法取种群规模为 100 个个体, 交叉概率 $P_c = 0.8$, 变异概率 $P_m = 0.1$, 最大误差限 $\epsilon = 0.0001$ 和最大迭代次数 $N = 2000$. 令处理机因子 α 在 $[0, 0.5]$ 之间, 个体迁移因子 β 在 $[0, 1.0]$ 之间以步长 0.02 变化. 在每组 (α, β) 下进行 10000 次独立试验, 得到平均成功率 R_{AS} 随 (α, β) 变化的关系, 如图 3 所示.

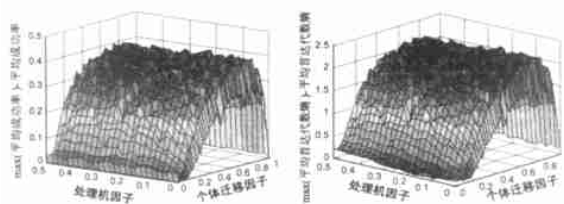


图 3 平均成功率随处理机因子和个体迁移因子变化的关系

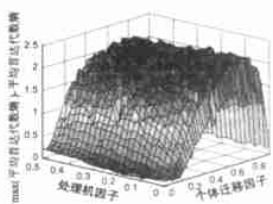


图 4 平均首次代数信息熵随处理机因子和个体迁移因子的变化

为方便观察, 图中以 $R_{AS} = \max(R_{AS}) - R_{AS}$ 代替了 R_{AS} . 从图中可以得出如下结论:

(1) 在相同的个体迁移因子 β 下, 算法的平均成功率随处理机因子 α 的不同仅在小范围内波动. 这说明 QPGA 本身对于 α 来说是鲁棒的.

(2) 在相同的 β 下, R_{AS} 随 α 的不同而呈现谷底和峰值. 这是因为: 当子算法之间有少量个体迁移时, 迁移个体为整个种群带来了新的遗传基因, 因此各个子算法都能够对问题解空间进行更有效的搜索, 算法的性能因此得以提高; 当子

算法之间有大量个体迁移时, 各子种群将趋向于对解空间中相同的区域进行搜索, 这很可能引起算法的不成熟收敛, 使子算法陷于一些局部极小值点之中, 表现在 QPGA 的平均成功率下降. 此外, 从图中还可以看到, 由于式 (10) 成立, R_{AS} 在 $\alpha = 1.0$ 和任意 β 下的值等于 $\alpha = 0.5, \beta = 0$ 时的值.

类似地, 可以得到平均首次代数信息熵 S_{AFP} 随 α 和 β 的变化关系, 如图 4 所示.

为方便观察, 图中用 $S_{AFP} = \max(S_{AFP}) - S_{AFP}$ 代替了 S_{AFP} . 如图所示, 平均首次代数信息熵是对算法首次代数分散程度的有效的定量度量. 在相同的 β 下, S_{AFP} 随处理机因子 α 的不同而在小范围内波动, 这说明 QPGA 首次代数的分散程度对 α 而言是鲁棒的. 而在相同的 α 下, S_{AFP} 随 β 呈现单峰形态, 而且 β 较大时数值急剧增大, 这还是由于大量个体迁移造成子算法趋于搜索解空间相同区域的结果.

总之, 从 R_{AS} 和 S_{AFP} 随 (α, β) 的变化看: (1) QPGA 对处理机时间的分配是鲁棒的; (2) 在子算法之间进行一定的个体迁移将对算法的性能产生正面影响, 而大量的个体迁移是不可取的. 实验结果表明, 个体迁移因子 β 一般而言不应大于 0.1.

3.4 平均首次代数及其方差随处理机因子和个体迁移因子变化的关系

用与 3.3 节中类似的方法得到 QPGA 的平均首次代数 N_{AFP} 及其方差 D_{AFP} 随 (α, β) 变化的关系, 如图 5 和图 6 所示.

图中再一次显示了较大的 β 将使各子算法趋向于搜索解空间中同样的区域, 并可能导致不成熟收敛的发生, 直接表现为在这些情况下 QPGA 的 N_{AFP} 较大. 而当仅有少量个体发生迁移时, 在不同的 α 下, 算法的性能都将获得大幅度的提升, 表现为此时算法能够获得较小的 N_{AFP} . 值得注意的是, 在相同的 α 下, 通过将有限的计算资源非对称地分配给各个子算法, 也能够提高算法的性能, 表现在这些情况下 N_{AFP} 在 0 处出现极小值.

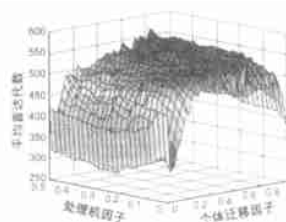


图 5 平均首次代数随处理机因子和个体迁移因子变化的关系

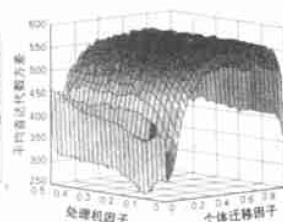


图 6 平均首次代数方差随处理机因子和个体迁移因子变化的关系

D_{AFP} 随 (α, β) 变化的趋势基本上与 N_{AFP} 一致. 在较小的 β 下, 通过非对称地分配有限的计算资源给不同的子算法或者是通过子算法之间的个体迁移都能够提高 QPGA 的性能. 然而, 它们提高算法性能的机理却很不一样:

(1) 通过非对称地分配有限的计算资源给不同的子算法可以提高组合算法的性能. 这显然是由于子算法本身在求解问题时存在的分散性和不确定性造成的. 从 MAS 的有关理论来理解, 发生这一现象的原因是由于系统协调了多个 Agent

所使用的计算资源,从而获得了各 Agent 之间的协同运作,表现在算法性能的提升.

(2)通过子算法之间的个体迁移来提高组合算法的性能则是由于迁移的个体给予算法带来了新的遗传基因,使得它有可能对解空间进行全面地勘探并因而发现问题的全局最优解.从 MAS 的有关理论来看,这种 Agent 之间的互通信息和相互协作导致了各个 Agent 的共同进化,进而获得了各 Agent 的协同运作,算法性能因而得到了提升.

3.5 处理机因子和个体迁移因子的确定

设算法在迭代次数为 N 时满足式(2)的概率为 $P(N)$, $N = 0, 1, 2, \dots$, 则其在迭代次数大于等于 N 时满足式(2)的概率为 $T \geq N P(T)$. 这样,当式(1)中 $= 0$, 即为单位阵时,PGA 在迭代次数为 N 时满足式(2)的概率为:

$$P(\cdot, N) = \prod_{k=1}^M P(T) - \prod_{k=1}^M P(T) \quad (11)$$

特别地,当 $M=2$ 时,上式可以简化为:

$$P(\cdot, N) = [P(T)] \cdot [P(T)] - [P(T)] \cdot [P(T)] \quad (12)$$

把 $P(\cdot, N)$ 代入式(6)和(7),得出不同 \cdot 下的平均首次时间 $N_{AFP}(\cdot)$ 及其方差 $D_{AFP}(\cdot)$,对于如图 1 所示概率分布的算法, $N_{AFP}(\cdot)$ 和 $D_{AFP}(\cdot)$ 如图 7 所示.

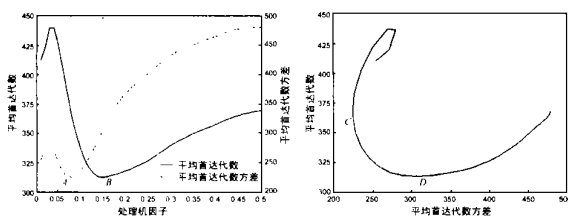


图 7 平均首次代数及其方差随处理机因子变化的关系 图 8 平均首次代数随平均首次代数方差变化的关系

以 \cdot 为参数,作 $N_{AFP}(\cdot)$ 相对与 $D_{AFP}(\cdot)$ 的变化曲线,如图 8 所示.

为使 QPGA 具有较小的 N_{AFP} ,应取值在图 7 中 B 点附近;为使其具有较小的 D_{AFP} , \cdot 值应该取在 A 点附近.而为使其在具有较小 N_{AFP} 的同时保持较小的 D_{AFP} , \cdot 应该介于 A 和 B 之间.此时 D_{AFP} 随 N_{AFP} 的变化将总在图 8 的 CD 段.或者说,当 QPGA 运行于这一区段时,它将具有较小的 N_{AFP} 、较小的 D_{AFP} 、或同时具备两者,因此, \cdot 值的选取应使 QPGA 尽量运行于这一区段内.在确定了 \cdot 后,考虑到在特定 \cdot 下 N_{AFP} 和 D_{AFP} 相对于 \cdot 的变化都有一个极小值点,该极小值点一般出现在 \cdot 较小的时候,依照这一点可以确定 \cdot 的取值.

4 求解组合优化问题

本节将讨论前述准并行遗传算法在求解旅行商 (Traveling Salesman Problem, TSP) 问题时的性能. TSP 问题定义为:给定点集 $P = \{p_1, p_2, \dots, p_N\}$ 及距离矩阵 $D = \{d_{ij}\}$, $i, j = 1, 2, \dots, N$, 求 $\{1, 2, \dots, N\}$ 的一个排列 $\{k_1, k_2, \dots, k_N\}$, 使 $\sum_{i=1}^N d_{k_i, k_{i+1}}$

取最小值(注意 $k_{N+1} = k_1$). TSP 问题是经典的 NP 难度问题,有大量学者对其进行研究^[1,2,16,17],其中有不少采用基于遗传算法的方法^[1,2,17]. 本文以通常使用的算法结构作为子算法构建求解 TSP 问题的准并行遗传算法,并对随机点序列问题和经典问题进行了试验.

4.1 随机点序列问题

在随机点序列问题中, $p_i, i = 1, 2, \dots, N$ 为在欧氏平面的区域 A 内服从二维概率密度函数 $\mu(x, y)$ 的独立分布的点.最短路径则由如下极限给出^[16].

$$\lim_{N \rightarrow \infty} \frac{l^*}{\sqrt{N}} = \sqrt{2} \int_A \sqrt{\mu(x, y)} dx dy$$

这里, l^* 是一个常数.特别地,当 $\mu(x, y)$ 为二维均匀分布时,有 $\lim_{N \rightarrow \infty} \frac{l^*}{\sqrt{N}} = 0.749$. 取种群规模为 100, 交叉概率 $P_c = 0.8$, 变异概率 $P_m = 0.05$, 取算法求解结果与 l^* 误差小于 0.01 时的迭代次数为首达次数.考虑算法之间没有个体迁移时的情况,令 $r_i = \frac{1}{M} \sum_{k=i}^M r_k, i = 1, 2, \dots, M$. 对 200 个点的随机序列进行 10000 次独立试验,求出 1 至 4 维时 QPGA 求解问题的最优平均首次代数及对应 r 值,如表 1 所示.

表 1

维数	r_1	r_2	r_3	r_4	N_{AFP}^*	提高
1	1.00	—	—	—	555.82	0.00 %
2	0.08	1.00	—	—	429.96	22.6 %
3	0.76	0.50	1.00	—	318.86	42.6 %
4	0.58	0.48	0.50	1.00	247.10	55.5 %

表中数据说明,通过协调多个 Agent 所使用的处理机时间可以提高整个算法的性能,并且这一提高随着算法维数的增加而增大.但由于在很高维数下难以得到计算资源分配向量,而且随着维数的增高,算法整体性能的提高效果将会降低,因此 QPGA 的维数没有必要太高.

4.2 经典问题

China144 城市问题是一个经典的 TSP 问题,有许多文章对此进行讨论,使用基于路径分割的分布式并行遗传算法求得的该问题的最优解为 30354km^[17],这是已知的最优解. 本文使用这一算法及其最优解作为参考,取算法求解结果与上述最优解误差小于 0.01 时的迭代次数为首达时间,求出三维 QPGA 在没有个体迁移的情况下求解该问题时 R_{AFP} 随 $\cdot = 1 / (\cdot_1 + \cdot_2 + \cdot_3)$ 及 $\cdot = 2 / (\cdot_2 + \cdot_3)$ 的变化情况,如图 9 所示. 这里, $R_{AFP} = N_{AFP} / \min(N_{AFP}) - 1$.

如图所示,首先, $\cdot < 0.5$ 和 $\cdot > 0.5$ 时 R_{AFP} 呈对称状况,这是显然的.其次,在最优情况下,用公式 $1 - \min(N_{AFP}) / \max(N_{AFP})$ 可以求得,三维 QPGA 能给算法带来 45 % 以上的性能提高.在该最优情况下,令 $\cdot = 0.02$,即个体迁移矩阵为

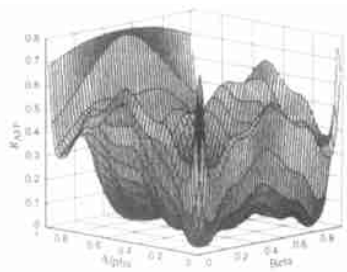


图 9 三维准并行遗传算法的特性

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + 0.02 \begin{pmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{pmatrix}$$

在这一情况下, QPGA 的性能与独立运行的算法相比, 提高了近一倍。

5 结论

本文提出了一种基于多 Agent 协同的并行遗传算法结构。该结构由若干计算单元组成, 每个计算单元实际上就是一个运行简单遗传算法的独立的计算 Agent。算法依照计算资源分配向量为它们分配不同的计算资源, 这体现了系统对各个 Agent 所使用资源的协调。同时, 算法还根据个体迁移矩阵在各个计算单元的遗传算法之间进行个体的迁移, 这体现了各个 Agent 之间的协作。综合起来看, 这一算法完全是在对独立计算 Agent 协同操作的基础上构建的。

当在计算资源有限可分的情况下(典型情况即为在只有一个 CPU 的串行计算机上)实现该并行算法时, 计算资源分配向量将决定各 Agent 在每一处理机时间片内运行的概率。由于多个 Agent 是轮换运行的, 它们使用的计算资源的数学期望值就等于计算资源分配向量中对应的分量, 因此各 Agent 实际上是一种微观上串行宏观上并行的准并行关系。本文使用结构和参数完全一致的子算法, 这样算法的对称性将仅由计算资源分配向量决定, 一般情况下是一种非对称结构。各 Agent 在逻辑上组成一定的拓扑结构, 这决定了控制它们交换个体的个体迁移矩阵。例如在一维环状结构下的个体迁移矩阵为一个准对角的稀疏矩阵。一般情况下, 任意两个 Agent 之间的通信强度(子算法之间个体迁移的数量)都是一样的, 因此个体迁移矩阵将完全由其结构和个体迁移因子来表示。

为衡量算法的性能, 本文提出了一组基于算法运行概率分布和统计规律的指标体系, 包括平均成功率、平均首次代数、平均首次代数方差和平均首次代数信息熵等。在此基础上, 本文详细分析了准并行算法在二维情况下的动态特性。结果表明: 在处理机时间一定的情况下, 通过为各 Agent 分配不同的计算时间和通过各 Agent 之间个体的迁移, 都能够提高算法的性能。从多 Agent 系统的角度来看, 前者是由于系统对 Agent 进行协调的结果, 后者则是各 Agent 之间协作的结果。这说明通过对多个计算 Agent 的协同, 能够在有限的计算资源下取得更好的计算效果。本文还对经典的组合优化问题进行了实验, 结果同样说明了这一点。

参考文献:

- [1] Goldberg. Genetic algorithms in search, optimization, and machine learning [M]. Addison-Wesley, Reading, MA, 1989.
- [2] 潘正君, 康立山, 陈毓屏. 演化计算 [M]. 北京: 清华大学出版社, 广西: 广西科学技术出版社, 1998.
- [3] Dongcheng Hu, Rui Jiang, Yupin Luo. An adaptive classifier system tree for extending genetics-based machine learning in dynamic environment [A]. Proceedings of AROB 99 [C]. Japan: AROB, 1999.
- [4] 孟庆春, 周长久, 纪洪波等. 带有对称编码的遗传算法在动力系统中的应用 [J]. 电子学报, 1999, 27(2): 59 - 63.
- [5] 张立华, 徐文立. 基于遗传算法的点模式匹配方法 [J]. 电子学报, 2000, 28(10): 36 - 40.
- [6] K Y Szeto, K H Cheung. Multiple time series prediction using genetic algorithms optimizer [A]. Proc. of the 1st Symposium on Intelligent Data Engineering and Learning [C]. Hong Kong, IDEAL, 1998.
- [7] R Tanese. Distributed genetic algorithms [A]. Proc. of 3rd International Conference on Genetic Algorithm, Lawrence Erlbaum Associates [C]. Hillsdale, NJ: ICGALEA1989. 434 - 439.
- [8] Muhlenbein H, Schomish M, Born J. The parallel genetic algorithm as function optimizer [J]. Parallel Computing, 1991, 17: 619 - 632.
- [9] J H Nang. A simple parallelizing scheme of genetic algorithm on distributed-memory multiprocessors [J]. International Journal of High Speed Computing, 1994, 6(3): 451 - 474.
- [10] K Y Szeto, K H Cheung, S P Li. Effects of dimensionality on parallel genetic algorithms [A]. Proc. of the 4th Int'l Conf. on Information System, Analysis and Synthesis [C]. Orlando, Florida, USA: 1998.
- [11] 石纯一, 王克宏, 王学军等. 分布式人工智能进展 [J]. 模式识别与人工智能, 1995, 8(增刊): 72 - 92.
- [12] Nicholas R Jennings, Katia Sycara, Michael Wooldridge. A Roadmap of Agent Research and Development [J]. Autonomous Agents and Multi-Agent Systems, 1998, 1: 7 - 38.
- [13] R Motwani, P Raghavan. Randomized Algorithms [M]. Cambridge University Press, Cambridge, 1995.
- [14] Bernardo A Huberman, Rajan M Lukose, Tad Hogg. An economics approach to hard computational problems [J]. Science, 1997, 275(3): 51 - 54.
- [15] Schaffer J D, Garuana R A, Eshelman L J, et al. A study of control parameters affecting online performance of genetic algorithms for function optimization [A]. Proc. of the 3rd Int'l. Conf. on Genetic Algorithms [C]. Los Altos, 1989.
- [16] J Beardwood, J H Halton, J M Hammersley. The shortest path through many points [A]. Proc. Cambridge Philos. Soc [C]. Cambridge: 1959.
- [17] Rui Jiang, K Y Szeto, Yupin Luo, Dongcheng Hu. A path-splitting scheme based distributed parallel genetic algorithm for large traveling salesman problems [A]. Proc. of Conference on Intelligent Information Processing [C]. Beijing: World Computer Congress, 2000.

作者简介:



江 瑞 男, 1975 年 1 月生, 2001 年毕业于清华大学自动化系, 获工学博士学位, 研究方向为遗传算法的理论及应用、进化计算方法的理论及应用、分布式并行计算方法、人工神经网络、数据挖掘和知识发现等。



罗予频 男, 1959 年 12 月出生, 1990 年毕业于日本名古屋工业大学, 获博士学位, 现在清华大学自动化系副教授, 研究兴趣为进化计算、与通信网络相关的图论、分布计算及容错。