

基于函数分解的可伸缩宏进化算法*

谢 涛** 陈火旺 张育林

国防科学技术大学计算机学院, 长沙 410073

摘要 基于函数可分解性提出可伸缩的宏进化算法. 宏进化算法以子函数的独立进化为竞争式进化内环, 而以子函数间的协调进化为合作式进化外环, 算法采用多父本杂交算子. 从数学上初步分析了该算法收敛性与函数分解粒度数之间的指数级关系, 指出宏进化算法的效率高于传统进化算法. 数值仿真与理论分析结果一致, 表明宏进化算法的效率主要由函数的分解粒度决定, 是一种可伸缩的进化算法, 具有实际工程应用价值.

关键词 宏进化算法 可伸缩性 函数分解 多父本杂交 复杂性

进化计算是一类可用来求解复杂优化与解空间搜索问题的仿生过程类算法, 被广泛认为是可成功地用于科学与工程各领域的一种自组织、自适应人工智能技术. 优化目标函数的复杂性可由维数与变量之间的非线性耦合关系描述. 然而, 维数越高, 变量间耦合度越强, 进化算法陷入局部最优域的概率越大. 为了降低函数的维数, 可以把复杂函数分解成较小的子函数; 为了降低变量之间的非线性耦合度, 可使各子函数间的耦合变量数最小化, 使子函数间尽量解耦.

传统进化算法的缺点集中表现在种群多样性与算法收敛性之间的矛盾. 提高种群多样性有利于解的全局最优性, 但同时也降低了算法的收敛性. 宏进化算法是多物种间以及物种与环境间竞争与合作过程的一种算法模型^[1,2]. 总目标函数分解后, 如果把对各子函数的独立进化视为竞争过程, 则对原目标函数的进化可以看成合作与协调过程. 总目标的综合进化压力与各子目标的局部进化压力构成一种宏进化过程, 种群在宏进化过程中保持一种动态的综合进化压力. 有机地结合宏进化算法中的“竞争与合作”过程, 可以实现算法的搜索效率与效果之间的动态平衡, 从而提高复杂函数的优化效率与效果.

1 进化算法中的函数分解

一般地, 函数可分解为若干子函数; 进一步, 如果各子函数间无耦合变量, 则称此类函数为可分离的. 进化算法的收敛性取决于评价准则的惟一性, 因此, 必须使得各子函数与原函数之间保持一种单调关系.

2000-08-23 收稿, 2000-11-20 收修改稿

* 国家自然科学基金(批准号: 69903010, 69785002)与博士后科学基金资助项目

** Email: taoxie@nudt.edu.cn

有效分解:假定函数 $g(x)$ 可分解成 k 个子函数 $g_i(x^i)$, 即 $g(x) = q(g_1(x^1), g_2(x^2), \dots, g_k(x^k))$, 则称这种分解方式对于进化算法是有效分解, 如果 $g_i(x^i)$ 与 $g(x)$ 之间满足严格单调关系(即固定 x/x^i 时, 由 x^i 的变化导致 $g_i(x^i)$ 的变化与 $g(x)$ 的变化之间的关系不是单调降就是单调升). 其中 $x \in D \subseteq R^n$, 并且 $x^i \subset x$, $\bigcup_{i=1, \dots, k} x^i = x$, x/x^i 即为由 x 中除去 x^i 的变量部分, $i = 1, \dots, k$. 进一步, 函数 $g(x) = \sum_{i=1}^k g_i(x^i)$ 称为有效和式分解. 和式分解是基本有效分解方式, 通过对数等运算的转换, 其他有效分解一般可转化为和式分解.

最小耦合分解:使 $\sum_{i=1}^k \sum_{j=1, i \neq j}^k |x^i \cap x^j|$ 最小的有效分解称为最小耦合分解, 其中 $|x^i|$ 表示分量 x^i 中变量的个数. 最小耦合分解使分解后各子函数间的耦合变量数最少. 对于不可分离的函数, 并行独立优化各子函数必然破坏各子函数间的相关性, 即对于一个子函数最优化的耦合变量可能破坏另一子函数的最优性. 因此, 必须以总目标函数作为各子函数间的协调函数.

2 可伸缩宏进化算法

以函数 $g(x)$ 的最小耦合有效和式分解为例, 说明算法的原理.

$$g(x) = q(g_1(x^1), g_2(x^2), \dots, g_k(x^k)) = \sum_{i=1}^k g_i(x^i).$$

有关函数 $g(x)$ 的分解说明见上节. 设 Ω 为 $g(x)$ 的种群集合, $\Omega = P \cup \Theta$, 其中 P 为父本集合, Θ 为子代集合; 相应地, 设 Ω_i 为子函数 $g_i(x^i)$ 的种群集合, $\Omega_i = P_i \cup \Theta_i$, P_i 与 Θ_i 类似定义. 设 $|P| = N$, $|\Theta| = M$; $|P_i| = \lfloor \frac{N}{k} \rfloor$, $|\Theta_i| = \lfloor \frac{M}{k} \rfloor$, $i = 1, 2, \dots, k$.

算法由 k 个子函数 $g_i(x^i)$ 的竞争式局部进化层与函数 $g(x)$ 的综合协调进化层构成, 竞争层与协调层之间由多父本杂交算子衔接. 各子种群 P_i 首先以相应子函数 $g_i(x^i)$ 分别从 P 中优选出 $\lfloor \frac{N}{k} \rfloor$ 个个体, 并按传统进化算法分别进化 m_1 代. 然后, 分别从各子群体 P_i 中按轮盘赌规则优选父本参与多父本杂交, 并基于 $g(x)$ 从杂交子代 Θ' 与 P 的并集 $(P \cup \Theta')$ 中优选出 N 个个体, 作为协调进化层的父本集合 P' , 再以函数 $g(x)$ 对 P' 中个体按传统进化算法进化 m_2 代, 形成下一代总目标父本集合 P , 协调进化层的变异算子作用在各耦合变量上. 这样, 一个宏进化周期结束. 如此重复宏进化过程, 根据函数的特点, 算法中竞争与合作的相对强度可分别由参数 m_1 与 m_2 控制. 当 $m_1 = m_2 = 0$ 时, 宏进化算法竞争层与协调层合二为一, 即竞争层各子函数只参与相应 $\lfloor \frac{N}{k} \rfloor$ 个个体的优选, 而协调进化层取消. 一个简化的宏进化周期就是, 先从当前总目标种群 P 中分别依每一子函数选择 $\lfloor \frac{N}{k} \rfloor$ 个父本个体并分区按序排列, 形成多目标的混合种群. 然后, 多父本杂交算子均匀随机地从不同子群体区域按轮盘赌规则选择杂交父本, 形成杂交子代集合. 最后, 对杂交子代进行变异, 变异后代再用总目标选择, 以形成下一代总目标种群. 对于可分离函数, 可取 $m_1 = m_2 = 0$, 此时宏进化算法等价于对各子函数的独立并行求解——松弛法.

多父本杂交算子: 设个体 $I = (x, \sigma)$, 其中 $x = (x_1, x_2, \dots, x_n)$, $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$, 即演化

策略(ES)表示方案^[3]. 定义 $x^{i(0)} = x^i / (x^{i-1,i} \cup x^{i,i+1})$, $x^{i,i+1} = x^i \cap x^{i+1}$, $x^{i-1,i}$ 类似定义, $x^{i(0)}$ 即为 x^i 中仅属于子函数 $g_i(x^i)$ 的变量集; 则有 $x = (x_1, x_2, \dots, x_n) = (x^{1(0)} x^{1,2} x^{2(0)} x^{2,3} \dots x^{k-1,k} x^{k(0)})$. 多父本杂交是从各子种群 P_i 中按轮盘赌规则分别取一个体 $x_i \in P_i$, $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n}) = (x_i^{1(0)} x_i^{1,2} x_i^{2(0)} x_i^{2,3} \dots x_i^{k-1,k} x_i^{k(0)})$, $i = 1, 2, \dots, k$; 多父本杂交算子 R_k 可定义为 $R_k(x_1 x_2 \dots x_{k-1} x_k) = (x_1^{1(0)} \bar{x}_{1,2}^{1,2} x_2^{2(0)} \dots \bar{x}_{k-1,k}^{k-1,k} x_k^{k(0)})$, 其中, $\bar{x}_{i,i+1}^{i,i+1} = \alpha \cdot x_i^{i,i+1} + (1 - \alpha) \cdot x_{i+1}^{i,i+1}$, 均匀随机变量 $\alpha \in [0, 1]$. 变异方差向量 σ 具有类似 x 的杂交过程.

宏进化算法基本步骤

第 1 步: 初始化宏进化种群 $P(0)$;

第 2 步: 基于子函数 $g_i(x^i)$ 值对第 j 代宏进化总父本集合 $P(j)$ 排序, 选取前 $\lfloor \frac{N}{k} \rfloor$ 个作为子函数 $g_i(x^i)$ 的初始种群 $P_i(0)$, 并按子函数 $g_i(x^i)$ 局部进化 m_1 代得到 $P_i(m_1)$, 其中 $i = 1, \dots, k$;

第 3 步: 按轮盘赌规则从各子种群 $P_i(m_1)$ 中选取父本进行多父本杂交, 生成 $P(j)$ 的杂交后代集合 $\Theta(j)$, 使得 $|\Theta(j)| = M$;

第 4 步: 基于函数 $g(x)$ 值从并集 $P(j) \cup \Theta(j)$ 中选取 N 个最优个体形成集合 $P^*(j+1)$; 然后基于函数 $g(x)$ 对 $P^*(j+1)$ 中个体进化 m_2 代, 得到第 $(j+1)$ 代宏进化父本集合 $P(j+1)$;

第 5 步: 终止条件满足否? 满足, 则终止; 否则, 转第 2 步继续宏进化过程.

3 算法性能分析

宏进化算法当 $m_1 = m_2 = 0$ 时, 多父本杂交算子与比例选择的混合作用具有与遗传算法中模式定理相媲美的性质, 这一性质可用如下定理描述. 基于函数分解的宏进化算法中, 高于平均适应值的解的分布密度的递增指数高于传统进化算法, 并与函数分解粒度数成指数级关系.

为了简化分析, 假定种群规模无限大. 于是, 解的概率分布函数变化即为种群分布密度变化^[4].

假定目标函数 $g(x)$ 可分离, 即 $g(x) = \sum_{i=1}^k g_i(x^i)$, 其中 $g_i(x^i) > 0$, $x^i \in D_i = \text{space}(x^i)$, $\bigotimes_{i=1}^k D_i = D = \text{space}(x)$, $D_i \cap D_j = \phi$ (即 $x^i \cap x^j = \phi$), $x \in D \subseteq R^n$, $i \neq j$, $i, j = 1, \dots, k$.

设 $f^t(x)$ 与 $f_i^t(x^i)$ 分别是第 t 代种群中个体 x 与子个体 x^i 的分布函数 ($i = 1, \dots, k$), 则必有 $f^t(x) = \prod_{i=1}^k f_i^t(x^i)$, $f_i^t(x^i) = \int_{D/D_i} f^t(x) dx^1 \dots dx^{i-1} dx^{i+1} \dots dx^k$, D/D_i 表示解的定义空间 D 中除去子解 x^i 的子空间 D_i 的剩余空间. 假定进化算法适应值函数为原目标函数, 则第 t 代种群个体适应值的均值为 $\bar{g}_t(x) = \int_{x \in D} g(x) f^t(x) dx = \sum_{i=1}^k \bar{g}_{t,i}$, $\bar{g}_{t,i} = \int_{x \in D} g_i(x^i) f^t(x) dx = \int_{x^i \in D_i \subseteq D} g_i(x^i) f_i^t(x^i) dx^i$ 为第 t 代种群相对各子目标函数的平均子适应值.

传统连续类进化算法中, 比例选择策略对种群分布的影响可表示为 $f^{t+1}(x) = \frac{g(x)}{\bar{g}_t} f^t(x) =$

$\frac{\sum_{i=1}^k g_i(x^i)}{\sum_{i=1}^k \bar{g}_{t,i}} f^t(x)$, 即种群密度变化指数为 $\frac{\sum_{i=1}^k g_i(x^i)}{\sum_{i=1}^k \bar{g}_{t,i}}$; 宏进化算法中, 比例选择与多父本杂交算

子对种群分布的混合影响可表示为 $f^{t+1}(x) = \prod_{i=1}^k \frac{g_i(x^i)}{\bar{g}_{t,i}} f_i^t(x^i) = f^t(x) \prod_{i=1}^k \frac{g_i(x^i)}{\bar{g}_{t,i}}$, 即宏进化

算法中种群密度变化指数为 $\prod_{i=1}^k \frac{g_i(x^i)}{\bar{g}_{t,i}}$. 于是, 定理的证明转化为: 当 $g_i(x^i) > 0$ 且 $g_i(x^i) \geq$

$$\bar{g}_{t,i} \text{ 时 } (i = 1, \dots, k), \prod_{i=1}^k \frac{g_i(x^i)}{\bar{g}_{t,i}} \geq \frac{\sum_{i=1}^k g_i(x^i)}{\sum_{i=1}^k \bar{g}_{t,i}}.$$

此处证明从略. 此外, 假定 $\gamma = \min\left\{\frac{g_i(x^i)}{\bar{g}_{t,i}}, i = 1, \dots, k\right\}$, 则有 $\prod_{i=1}^k \frac{g_i(x^i)}{\bar{g}_{t,i}} \geq \gamma^k$. 此定理说明采用多父本杂交算子的宏进化算法中, 解的密度变化指数与分解粒度数 k 成指数关系. k 越大, 高于平均适应值的解的密度递增指数越高. 因此, 基于函数分解的宏进化算法的求解效率高于传统进化算法, 且分解粒度愈小 (k 越大), 算法求解效率愈高. 此结论可推广至 $m_1 \neq 0$ 和/或 $m_2 \neq 0$; 对于有序相关类弱耦合函数, 此结论的一般形式也是应该满足的, 并可通过数值计算验证.

4 数值仿真分析

宏进化算法效率可用典型的高维复杂函数优化进行数值分析.

$g(x) = \sum_{i=0}^{n-1} g_i(x_i, x_{i+1})$, $x \in \mathbb{R}^{n+1}$ 其中基本子函数为

$$g_i(x_i, x_{i+1}) = 2 - \exp\left(-\left(\left(x_i + \frac{1}{i+1}\right)^2 + \left(x_{i+1} - \frac{1}{i+1}\right)^2\right)\right) - \exp\left(-\left(\left(x_i - \frac{1}{i+1}\right)^2 + \left(x_{i+1} + \frac{1}{i+1}\right)^2\right)\right)$$

$q(x) = \sum_{i=0}^{n-1} q_i(x_i, x_{i+1})$, $x \in \mathbb{R}^{n+1}$, 其中基本子函数为

$$q_i(x_i, x_{i+1}) = \frac{1 - \exp\left(-\left(\left(x_i + \frac{1}{i+1}\right)^2 + \left(x_{i+1} - \frac{1}{i+1}\right)^2\right)\right)}{1 - \exp\left(-\left(\left(x_i - \frac{1}{i+1}\right)^2 + \left(x_{i+1} + \frac{1}{i+1}\right)^2\right)\right)}$$

函数 $g(x)$ 与 $q(x)$ 是可扩展的有序耦合关系函数, 存在惟一全局最优解, 其复杂性均随维数升高而增大, 且函数 $q(x)$ 的结构比 $g(x)$ 复杂得多. 如果采用梯度下降类解析算法, 此类函数的高维优化是几乎不可求解的; 即使采用随机启发式搜索算法, 也因易陷入众多局部最优域而过早收敛, 如模拟退火算法与传统的进化算法等.

仿真分析采用简单宏进化算法, 即 $m_1 = m_2 = 0$. 简单宏进化算法每代访问总目标函数次

数(计算代价)与传统进化算法相同,因而仿真结果可用每代最优解的收敛曲线比较. 连续函数优化一般采用演化策略,因此,采用演化策略变异算子的简单宏进化算法简称宏演化算法(MES). 数值分析的目的是,分析宏演化算法收敛性与函数分解粒度之间的关系,并比较宏演化算法与传统演化策略的收敛性随函数复杂性增长时的变化情况.

基于函数的不同分解粒度,图 1 与图 2 对宏演化算法收敛性与函数分解粒度之间的关系进行了数值分析,并与传统演化策略作了比较. 取种群父本个体数为 $N = 41$,子代个体数为 100,算法采用精英保留策略. 取 $n = 100$,函数可按序均匀分解成 5 等份(5×20), 10 等份(10×10)和 20 等份(20×5),此即为函数的最小耦合分解,如 $g(x)$ 的 5 等份最小耦合分解可表示为 $g(x) = \sum_{i=0}^4 \sum_{j=20 \times i}^{20 \times (i+1)-1} g_i(x_j, x_{j+1})$. 传统演化策略可视为不作分解的宏演化算法(1×100). 仿真结果表明,宏演化算法的收敛性随函数分解粒度降低(k 增大)而提高,且远高于传统演化策略.

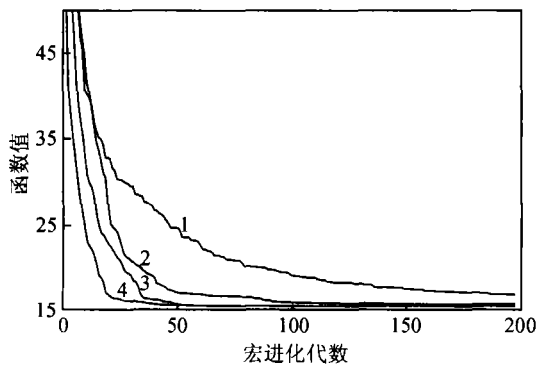


图 1 函数 $g(x)$ 分解粒度与收敛性关系
1——ES, 2——MES(5×20), 3——MES(10×10),
4——MES(20×5)

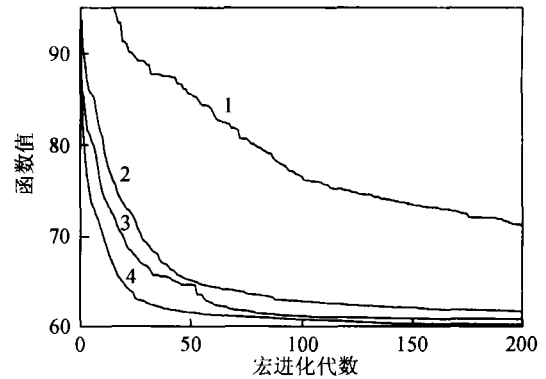


图 2 函数 $q(x)$ 分解粒度与收敛性关系
1——ES, 2——MES(5×20), 3——MES(10×10),
4——MES(20×5)

图 3 与图 4 是宏演化算法与传统演化策略可伸缩性能的比较仿真分析结果. 两函数的基本子函数个数分别取 50, 100 与 200, 即 $n = 50, 100$ 与 200, 相应种群父本规模亦随函数维数成

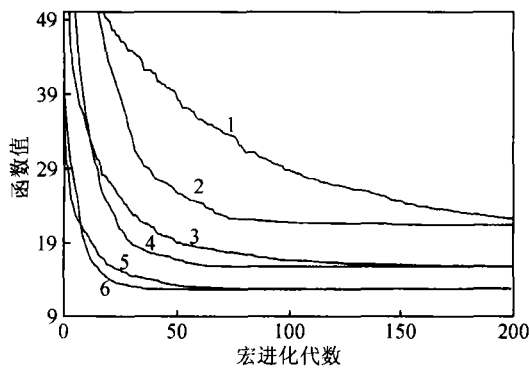


图 3 基于函数 $g(x)$ 的可伸缩性比较
1——ES(200), 2——MES(20×10), 3——ES(100), 4——
MES(10×10), 5——ES(50), 6——MES(5×10)

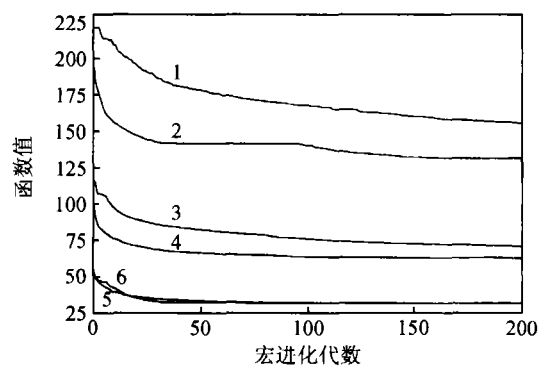


图 4 基于函数 $q(x)$ 的可伸缩性比较
1——ES(200), 2——MES(20×10), 3——ES(100), 4——
MES(10×10), 5——ES(50), 6——MES(5×10)

倍增加,分别取 $N = 11, 21$ 与 41 ,子代规模均取 100 ,每一个体变异最多不超过 3 个变量,算法亦采用精英保留策略.对 3 种不同维数的函数进行相同粒度的最小耦合分解,使得分解子函数均含 10 个相邻基本子函数.图 3 与图 4 表明,宏进化算法的收敛性随函数维数的增长变化不明显,而传统演化策略则有明显下降,且函数愈复杂,宏进化算法的效率愈明显.此外,用网络计算多任务调度问题^[5]进行仿真分析,也得到了类似结果.

5 结论与问题

宏进化算法中子种群间既竞争又合作的进化过程是实现种群多样性与算法收敛性之间动态平衡的一种有效方式.宏进化算法的收敛性主要由函数的分解粒度决定.对于同一函数,宏进化算法的收敛性与分解粒度成指数级关系;对于可扩展函数,具有同一分解粒度的宏进化算法的收敛性基本不变.理论分析与数值分析结果一致,均表明宏进化算法的收敛性高于传统进化算法.宏进化算法可以克服复杂函数的维数灾难,并可最大程度地降低由非线性耦合关系所产生的复杂性,是一种可伸缩的进化算法.

宏进化算法子种群规模的均分原则只是一种简化处理,搜索过程中应根据子种群对总目标的贡献按比例分配,这是一种自适应的子种群规模分配策略,值得进一步研究.此外,实际问题往往存在约束条件,如背包问题等,约束条件会使宏进化算法的实现复杂化.目标分解如何与约束条件相结合,以使子目标间的“强强联合”对约束条件的破坏性最小,也是值得进一步研究的问题.

参 考 文 献

- 1 Hillis W D. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D*, 1990, 42: 228
- 2 Ikegami T, et al. Computing symbiosis-emergence of symbiotic behavior through evolution. *Physica D*, 1990, 42: 235
- 3 Back T, et al. A survey of evolution strategies. In: Below R K, et al. eds. Proc of the 4th ICGA, San Diego, 1991. San Mateo: Morgan Kauffman Publishers, 1991, 2
- 4 Qi X F, et al. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. I: Basic properties of selection and mutation. II: Analysis of the diversification role of crossover. *IEEE Trans on Neural Networks*, 1994, 5(1): 102
- 5 Zhong Q X, et al. Task matching and scheduling based on coevolutionary computational model. In: Proceedings of the 3rd World Congress on Intelligent Control and Automation, HeFei, China, 2000. HeFei: Publisher of National University of Science and Technology, 2000, 1910