第 34 卷 第 8 期 计 算 机 工 程 2008 年 4 月
Vol.34 No.8 Computer Engineering April 2008
·人工智能及识别技术· 文章编号：1000—3428(2008)08—0221—02 文献标识码：A 中图分类号：TP18

自适应小生境遗传算法在系统级综合中的应用

陈云峰，段成华
(中国科学院研究生院信息科学与工程学院，北京 100049)

摘 要：多目标遗传算法是解决 SoC 系统级综合问题的有效途径之一，但现有的遗传算法只能求得非劣解集前沿的一部分，局部搜索能力差，收敛速度较慢。该文通过结合小生境技术，根据种群往代的多样性信息，自适应地确定子种群的规模和交叉、变异的概率，提出一种自适应小生境遗传算法，有效提高解集的覆盖率，加快收敛速度。以视频编解码的系统级综合为例，证明该算法可以较快地产生较多非劣解。

关键词：系统级综合；自适应；小生境技术；多目标优化

Application of Self-adaptive Niching Genetic Algorithm in System Level Synthesis

CHEN Yun-feng, DUAN Cheng-hua
(School of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, Beijing 100049)

【Abstract】 Multi-objective genetic algorithms are effective in solving SoC system level synthesis. These available algorithms can only attain part of the whole Pareto front, because of the worse local searching ability, the convergence speed is slow. In order to overcome these disadvantages, an updated multi-objective genetic algorithm, self-adaptive niching genetic algorithm, is proposed, which integrates the niching technique, self-adaptively adjusts the sub-populations' size and the probabilities of crossover and mutation by using the previous population's diversity information. It can effectively increase the solutions coverage and improve the convergence speed. Using the updated algorithm, this paper succeeds in optimizing the synthesis of video codec. Result indicates that the new algorithm can rapidly acquire more Pareto optimal solutions and proves the superiority of the algorithm.

【Key words】 System Level Synthesis(SLS); self-adaptive; niching; multi-objective optimization

1 概述

硬件系统级综合(System Level Synthesis, SLS)是SoC设计中的重要环节。目前对SLS尚无公认的定义，但一般认为其必须具备如下特征[1]：(1)SLS被认为是从行为描述(功能对象的粒度是：tasks, procedures, processes)到结构描述(结构对象是通用或专用处理器，如：ASICs, busses和memories)的映射；(2)类似于行为级综合，一般包括分配、绑定和调度等任务。

作为SoC设计技术中的一个热点问题，系统级综合问题已得到广泛的研究[2]。在众多的实现途径中，比较重要的有基于整数线性规划的算法、模拟退火算法和遗传算法[2]。规划算法在系统级综合的应用中有很高的求解精度，但由于其本身是一个NP完全问题[2]，因此在处理系统的规模增大时时间开销难以令人满意。作为近似算法，模拟退火算法能够有效克服局部的极小值陷阱，但是其收敛速度相当慢[1]。文献[1]将遗传算法应用于系统级综合中，以一个采用H.263 标准的图像压缩系统为例，获得了满意的结果。基于图理论，它把SLS问题建模为任务级的数据流图规范到异构软/硬件体系结构的映射。

在进行系统级综合时，设计者总是希望目标系统的性能尽可能好，比如在使用成本较小的情况下，同时希望系统的响应时间、功耗又尽量小。这些相互矛盾的指标在设计时必须进行权衡处理，构成了一个典型的多目标优化问题。如果能求得该问题的非劣解集，决策者就能对多个指标间的关系有完整的认识，从而能更好地决策和折中。为此，人们开始采用多目标遗传算法来处理多目标优化问题，把系统级综合问题看作一个需要求解的多目标函数，采用基于群体的搜索策略，逐代进化，可以在一次运行过程中生成多个非劣解。为了提高系统级综合问题解集的质量，本文对多目标遗传算法进行了改进，为了增大种群的多样性，引入小生境技术，自适应地修改交叉率和变异率，提出了自适应小生境遗传算法(Self-adaptive Niching Genetic Algorithm, SNGA)。

2 自适应小生境遗传算法

一个多目标问题可以表示如下[3] (不失一般性，假设各个目标皆求极小值)，其中，决策向量 $\boldsymbol{x}\in R^m$；目标向量 $\boldsymbol{y}\in R^n$，最小化 $\boldsymbol{y}=f(\boldsymbol{x})=(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\cdots,f_n(\boldsymbol{x}))$；约束条件为 $g_i(\boldsymbol{x})\le 0$，$i\in\{1,2,\cdots,k\}$。

定义 1 Pareto优于：若目标向量解 $\boldsymbol{u}=(u_1,u_2,\cdots,u_n)$部分小于 $\boldsymbol{v}=(v_1,v_2,\cdots,v_n)$，即 $\forall i\in(1,2,\cdots,n),\ u_i\le v_i\wedge\exists i\in(1,2,\cdots,n),\ u_i<v_i$，则称 $\boldsymbol{u}$ Pareto优于 $\boldsymbol{v}$，记为 $\boldsymbol{u}\succ\boldsymbol{v}$。

定义 2 Pareto最优解：决策变量 $x_u\in R^m$ 称为多目标问题的Pareto最优解(也称非劣解)，当且仅当不存在决策变量 $x_v\in R^m$，

作者简介：陈云峰(1982－)，男，硕士研究生，主研方向：SoC 系统级设计；段成华，教授
收稿日期：2007-05-23 **E-mail：**chenyunf@mails.gucas.ac.cn

使得相应的目标向量解 $\boldsymbol{v}=f(x_v)=(v_1,v_2,\cdots,v_n)$ Pareto优于 $\boldsymbol{u}=f(x_u)=(u_1,u_2,\cdots,u_n)$，即 $\neg\exists x_v, \boldsymbol{v}\succ\boldsymbol{u}$。

定义 3 Pareto 解集：所有 Pareto 最优解的集合，即 $P=\{\boldsymbol{x}\in R^m \mid \neg\exists \boldsymbol{x}'\in R^m : f(\boldsymbol{x}')\succ f(\boldsymbol{x})\}$。

定义 4 Pareto 前沿：$PF=\{f(\boldsymbol{x}) \mid \boldsymbol{x}\in P\}$。

可以看出，这里的 Pareto 前沿就是决策者所需的可能满意解。但是，由于多目标优化问题的 Pareto 解集往往是一个集合(一片区域)，问题的关键是更多、更均匀地找出此集合中的元素，让决策者有足够的选择余地。

2.1 适应度函数

遗传算法中适应度函数决定了群体的进化方向。与单目标优化不同，多目标优化处理的是一向量函数空间，它的适应度函数应体现出一个点在目标函数域中的好坏。每一个群体中的非劣点应具有同样的重要性。为了得到最优解，仍采用文献[1]中的 Pareto-ranking 技术，取

$$F(\boldsymbol{x})=\sum_{i=1,2,\cdots,n,\boldsymbol{x}\neq x_i}\begin{cases}1 & f(x_i)\succ f(\boldsymbol{x})\\ 0 & \text{其他}\end{cases}$$

即个体 $\boldsymbol{x}$ 的适应度由群体中 Pareto 优于 $\boldsymbol{x}$ 的个体数决定。可以看到，每一代群体中的 Pareto 最优解的适应度值都为 0。

2.2 小生境的自适应形成

多目标遗传算法普遍采用小生境技术以维持种群的多样性。本文计算当前种群中每个个体的目标值和适应度值，轮流选择多目标优化问题的一个目标，根据该目标值对种群进行排序；将排序种群划分成若干子种群，子种群规模随着大种群的多样性的变化而自适应变化，各子种群独立完成遗传操作。设大种群规模为 N，子种群规模为 K，则

$$K=\begin{cases}2 & S\geq\delta\\ f(S) & S<\delta\end{cases}$$

其中，$f(S)$是关于S的一个函数，可根据问题的特征预先设置；S为大种群个体的Spacing距离[4]；δ为一常数。

当大种群个体多样性降低时，S 就变大，当 S 大于阈值 δ 时，子种群规模就降低到最低限度 2，以刺激种群多样性的提高，而子种群间是相对隔离的。在生物学中，隔离是保持物种多样性的重要条件之一。在小生境遗传算法对多目标函数的优化中，隔离使得各子种群在解空间内搜索各可能存在非劣解点的区域，并不断缩小其可信域(promising area)。每一代进化完成后将子种群合并，然后重新计算个体适应度值，获取本代 Pareto 最优解集，再按照 Spacing 距离重新划分子种群。

2.3 小生境中的遗传算子

在遗传算法的操作算子中，选择算子起到启发进化方向的作用，交叉算子起到全局搜索的作用，而变异算子通常被认为是一种背景操作或辅助操作，它能够以大于 0 的概率找回丢失的优良基因。

在小生境技术中，本文采用(μ+λ)选择机制，它被认为是遗传算法的几种流行选择机制中选择压力最高的一种[5]。当在种群中进行随机配对的交叉操作时，(μ+λ)选择机制能产生最快的局部收敛速度。(μ+λ)选择策略是指在μ个父代个体和由这μ个个体交叉产生的λ个子个体中选择μ个最佳个体。

交叉和变异算子能够探索新的可行解空间，使种群中具有多样性，同时又使种群个体趋于最优解。交叉和变异的具体实现依赖于所选用的染色体表示方式，因此，仍然采用文献[1]的算子，对 α 采用均匀交叉和按位变异，对各 lists 采用基于顺序的交叉和随机置换变异。

2.4 自适应遗传算法参数

交叉和变异相互配合、相互竞争能使进化过程具有全局的和局部的均衡搜索能力，因此，交叉率和变异率的确定是提高遗传算法性能的关键[5]。已有很多文献讨论了遗传算法控制参数的自适应调节问题，最有代表性的是文献[6]提出的根据适应度值调节个体交叉率、变异率的方法，本文对其作修改用以自适应调节子群体的交叉、变异概率：

$$Pc=\begin{cases}k_1(N-f_{\text{avg}})/(N-f') & f'<f_{\text{avg}}\\ k_3 & f'\geq f_{\text{avg}}\end{cases}\quad(1)$$

$$Pm=\begin{cases}k_2(N-f_{\text{avg}})/(N-f) & f<f_{\text{avg}}\\ k_4 & f\geq f_{\text{avg}}\end{cases}\quad(2)$$

其中，Pc表示子群体的交叉率；Pm表示子群体中个体的变异率；N表示种群规模；f_{avg}表示种群平均适应度值；f'表示子种群的平均适应度值；f表示要变异的个体适应度值；k_1, k_2, k_3, k_4 是在 0~1 之间取值的常数，k_3 和 k_4 较大。

从式(1)可以看出，如果子群体较差，其平均适应度值大于种群平均适应度值，对其就给予较大的交叉率 k_3；如果子群体较优良，其平均适应度值小于种群平均适应度值，则依据其优良程度赋予此子群体相应的交叉率，适应度值越接近 0，交叉率就越小。变异率的设置与其相仿，只是对应地为每个个体进行设置。

2.5 算法描述

SNGA 的算法描述如下：

(1)初始化群体

随机生成每个个体的染色体；

(2)解码

求得每个个体的性能指标O_i，$i\in\{1, 2,\cdots, m\}$，m为目标数；在本代群体内，求取每个个体的适应度值*fitness*；

(3)$i := 1$；

(4)若 $i\leq m$ 转下一步；否则转(3)；

(5)规划子群体

种群按性能指标O_i的值进行排序，计算大种群Spacing距离S，若 $S<\delta$则设置子种群规模$K=f(S)$，否则$K=2$；依次选取相邻K个个体形成子种群，并对交叉率和变异率按式(1)、式(2)作相应变化；

(6)交叉、变异

子种群内个体进行交叉和随机变异，并进行(μ+λ)选择；

(7)合并成大群体，按照 *fitness* 求取本代群体的 Pareto 最优解集；

(8)若达到最大代数，输出 Pareto 最优解集，算法结束；否则 $i := i + 1$，转(4)。

3 仿真算例

系统级综合通常面临多个相互矛盾的优化目标，为作对比分析，本文用文献[1]中的图像压缩系统的系统级综合为例，以降低使用成本和减小系统执行周期为目标做仿真实验，与原文采用限制锦标赛选择(RTS)的遗传算法所得结果进行比较。

统一设定所有参数：群体规模 N=100，最大代数 G=200，参数 δ=30, $f(S)=N/S$, k_1=0.5, k_2=0.05, k_3=0.7, k_4=0.2。将两种算法分别产生的 Pareto 前沿放在一个集合中再进行 Pareto 排序，产生一个新的 Pareto 前沿，称其为该算例的 Pareto 前沿参照集，分别计算 2 种算法产生的解集在参照集上解的个数。2 种算法采用同样的初始群体运行 10 次，实验结果见表 1，覆盖率一列中，分母是运行一次产生的 Pareto 最优解的个数，分子是这些解在参照集上的个数。

按照文献[4]对非劣解集的覆盖率评价标准，平均覆盖率如下：

(下转第 225 页)