

改进的用于半导体器件综合系统的遗传算法

谢晓锋, 张文俊, 杨之廉
(清华大学微电子学研究所, 北京 100084)

摘 要: 本文对应用于器件综合系统的遗传算法 GENOCOP 进行了改进. 将实数设计空间根据参数的工艺精度影响转换为整型空间, 并加入适应性复合算子利用已经得到的点来扩展和开发准可行空间. 使其保持有效搜索到可行解的特性的同时, 在同等的算法设置下, 提高了对可行空间的覆盖率 (约 2.87 倍), 可以帮助设计人员更有效的设计可工作的器件.

关键词: 遗传算法; 器件综合; 针对带约束数值型优化问题的遗传算法

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2002) 02-0184-03

Improved Genetic Algorithm for Semiconductor Device Comprehensive System

XIE Xiao-feng, ZHANG Wen-jun, YANG Zhi-lian

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: The genetic algorithm, called GENOCOP, used in the semiconductor device comprehensive system, is improved. The real-valued design space is transformed into integer-valued design space according to the effects of the technical precision of the processing parameters. An adaptive combined operator is introduced to extend and exploit the quasi-feasible space by utilizing the searched points. While keeping the advantage to find the solutions, the coverage of the solution space is improved (about 2.87 times) with same setting. This will help the designers to design workable device more efficiently.

Key words: genetic algorithm; device decomposition; GENOCOP (Genetic algorithm for Numerical Optimization of Constrained Problems)

1 引言

在半导体器件设计中, 通常需要根据所希望得到的电学性能来利用器件模拟器设计器件. 传统所用的方法是首先根据经验产生一个初始器件模板, 然后对其中一些参数设计实验 (DoE)^[1], 并模拟出一组不同参数器件的性能, 然后利用得到的变化趋势对参数在设计空间 S_D 中手工进行调整, 直到得到可行结果. 这种设计方法有如下问题: (1) 在遇到可行空间 S_F 远小于设计空间 S_D 的情形或特性空间非常复杂时, 通常需多次调整, 需要设计人员有较好的判断能力, 而且不一定能得到所需的结果. (2) 对器件设计, 由于每个设计参数都受工艺精度的影响, 要得到可工作的器件, 要求得到的参数有一个可行范围而不是仅仅一个点.

器件综合方法学根据所希望的性能要求在设计空间 S_D 中寻找符合特性要求的可行空间 S_F , 是器件/工艺总体综合方法的重要组成部分^[2]. 很多时候 S_F 非常复杂, 不可能用函数形式进行描述. 在先前的工作中^[3], 我们构造了一个器件综合系统, 利用搜索算法调用器件模拟器对器件设计空间 S_D 进行搜索来得到一组可行点. 为保证搜索的全局性, 其中的搜索算法我们选用一种遗传算法 GENOCOP^[4], 并应用于综合

FIBMOS 器件.

可行空间的覆盖率 R_C 对搜索算法来说是非常基本的质量标准^[5]. 大的覆盖率有更大余地来选择合适的器件, 使综合系统实用. 本文首先简要分析 GENOCOP 算法, 然后着重于改进使其更加适应器件综合系统的要求. 对设计空间考虑了参数的工艺精度影响转换为整型空间, 并增加算子来利用所有已经搜索过的点尤其是可行点的信息, 使其能快速搜索到分布合理的可行点提高对 S_F 的覆盖率.

2 GENOCOP 系统

GENOCOP 是一种实数型遗传算法, 其个体由设计参量直接表示. 它首先在设计空间中随机产生所需的初始种群, 然后再根据个体的适应度选择父代, 进行遗传操作产生的子代个体, 提取出行为特性作为目标值和所需特性进行评价得到个体的适应度, 随后根据等级选择机制来选择生存个体. 如果演化没达到结束条件, 则继续进行下一代演化. 在遗传算法结束后, 输出所有的可行点.

该算法采用了七种遗传操作: 三种交叉运算, 包括简单算术 (simple arithmetical) 交叉、整体算术 (whole arithmetical) 交叉和启发式 (heuristic) 交叉; 四种突变运算, 包括一致 (uniform) 突

变、边界(boundary)突变,非一致(non-uniform)突变和整体非一致(whole non-uniform)突变.用交叉的开发能力寻找可行点,用突变的探索能力来增加全局性.

只要解空间非空,随着演化代数的增加,以上遗传算子在概率意义上能搜索到任何模式^[6].然而由于核心的开发策略使用的是基于较高适应度个体的交叉算子,在工作于实数空间得到的解经常过分集中,导致对解空间的覆盖效率较低.

3 算法改进

定义一个“准可行空间” S_{FQ} :即由所有可行点的连线形成的所有不包含有非可行点的超多面体的集合.如图 1 所示灰色部分.

定义“准非可行空间” S_{DQ} :由所有不可行点(S_D 的边界点没被搜索之前,作为不可行点对待)的连线形成的所有不包含有可行点的超多面体的集合.

图 1 准可行空间 S_{FQ} 示意图

设计者可利用准可行区获得信息来选择合适参数设计器件.为了使准可行空间更好地反映实际的可行空间,需要得到更多的解并使得解分布比较合理.为此采用如下方法:

3.1 整型设计空间

为了防止得到过分集中的解,对每个设计变量按一定步长划成网格,只对每个格点进行搜索,这样所有的个体之间有一定距离.每个设计变量将使用整数表示,离散后的搜索空间将变成整型空间.

设计变量的步长的选择非常重要,如果太小,就起不到防止解过于集中的作用;如果太大,就会略过很多有用的点,甚至无法搜索到可行点.

把步长选择为工艺精度的数量级将比较合适,这样得到的可行点在容许误差内不会集中.而且如果可行点处于 S_{FQ} 的内部,则由此设计参数选择的器件有很大的可能满足可工作的要求.

3.2 算子改进

尽管大的种群尺寸和代数有助于得到更多的解,但由于为提取个体行为特性所需进行的器件模拟比较耗时,所以希望总的搜索点数尽量少.

为此可利用已经获得的点尤其是可行点的信息来获得比交叉算子更高的开发能力.增加如下算子增强搜索可行点的能力:

(1) 解空间扩展算子 定义一个可行点和一个不可行点为一对相异点,在它们的连线上进行搜索可以得到可行区的边界.本算子利用所有已得到的点来得到 S_{FQ} 边界上的点和与之最近的 S_{DQ} 边界上的点,新点将产生在最有可能得到边界可行点的区域,即设计空间 S_D 中位于 S_{DQ} 和 S_{FQ} 两种空间之间的区域 S_E .得到的边界可行点可用将来 S_{FQ} 扩展到 S_F 的边界.

假设在已经搜索过的点中有 m ($m > 0$) 个可行点和 n ($n > 0$) 个不可行点.计算所有相异点对的距离,所有离某个不可

行点最近的可行点为 S_{FQ} 的边界点,离该点最近的不可行点即为最近的 S_{DQ} 的边界点,这两点称为相异边界点对.优先搜索稀疏区域,即选距离最长的相异边界点对进行整体算术交叉:

$$y^* = y_2 - n \cdot (y_2 - y_1)$$

(1) 图 2 解空间扩展算子搜索区域示意图

其中 y_1, y_2 为选中的相异点对, n 在 0 到 1 之间,一般选择在 0.5 左右, y^* 新个体大致位于 S_E 中.

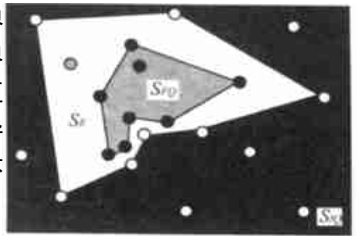
在程序实现中,可以利用过往的计算信息减少计算量.使用一个数据结构 AlienPoints 保存相异点对的信息,其中包括相异点的位置 P_{AL} ,和相异点的距离 P_{AD} ,以及是否可作为父代使用的标志 P_{AU} .为减小内存消耗,使用一维数组 V_F 来保存可行点及其最近相异点,一维数组 V_I 来保存不可行点及其最近相异点,以避免使用一个 $m \times n$ 矩阵保存所有相异点数据.每有新个体,只计算新增加的一组相异点,和保存的相应信息作比较,如果结果有更新则存入 V_F 和 V_I 中的相应位置,并对有更新的点的 P_{AU} 标志复位.由 V_I 的相异点位置信息确定 V_F 中的边界可行点,再得到这些点(不包括 P_{AU} 置位的点)中距离最长的相异点对,如果该对相异点之间的距离均小于某预定搜索精度值(最小为 2)时,本算子能力耗尽,否则进行算术交叉得到新点,并将标志 P_{AU} 置位.这样可在得到的点无效时,根据 P_{AU} 防止重复使用该对点作为父代.

(2) 解空间开发算子 本算子用来开发准可行空间 S_{FQ} ,在其中较为稀疏的区域搜索来得到分布较为合理的点以更为精确的描述实际的复杂可行空间.

在稀疏区域定点的方法如下:设有 n 个 ($n > 1$) 可行点,首先求解出不同可行点的中点的位置,对每个中点求最大半径(该中点到所有可行点的最小距离),取半径最大的一个中点作为新个体.

在程序实现中使用一个数据结构 MidPoint 保存中点的信息,其中包括中点的坐标 P_{MC} ,当前的最大半径 P_{MR} .所有的中点将被存入一个数组 V_{PM} ,在第 $n+1$ 个可行点产生时,对 V_{PM} 中的原有元素的 P_{MR} 和到新点的距离作比较得到新的 P_{MR} .然后将新可行点与先前的 n 个点求中点,计算这些点的 P_{MR} ,并加入到 M_{PM} 中.使用时,从 M_{PM} 中得到 P_{MR} 最大的中点作为新个体.为了减小 V_{PM} 的规模,其中所有 P_{MR} 小于预设稀疏长度 L_S (最小为 1)的中点和所有已被作为个体使用过的中点将被从 V_{PM} 中删除.一旦 V_{PM} 为空,则其开发能力耗尽.

(3) 适应性复合算子 适应性复合求解算子将以上两个算子(1)和(2)和一致突变算子复合使用,它可以根据各阶段的需要在探索和开发之间适应性变化.在我们改进的系统中替换掉原来单纯的一致突变算子,如以上两个算子没有得到新搜索点的能力时,则使用一致突变算子(这时和 GENOCOP 在算子上没有差别).早期使用大的突变率可以改善解的质



量^[7],但一旦产生的点的分布使得有开发解的可能,以上两个算子(1)和(2)可以加快解的开发扩展,在开发能力耗尽后,重新成为一致突变算子来探索新空间。

3.3 防重复计算策略

设计空间离散成整型后,总的个体数目将大为减少。这样,个体发生重复的几率变得很大,尤其是在演化了一定的代数以后。为防止重复计算,所有被搜索过的个体将被存入数据库,用来测试所有新个体是否为重复个体。如是,则使用适应性复合算子来重新生成子个体,直到确实为新个体为止。

4 说明与验证

设计如下实验来比较原有 GENOCOP 算法和改进的算法。采用相同的初始个体(无可行点)和算法设置(种群大小为 50,代数为 20,需 450 次计算)执行各 11 次,去掉最坏的一次后取平均,比较对可行空间的覆盖率 R_c 。 R_c 的计算通过将可行区按预定步长(为便于比较,设置成和划分整型空间时的步长一样)划分后,计算可行点占据的格子数和可行区总格子数的比值。例子

为设计一个 MOS 器件,其中的设计参数分别为沟长 L_{eff} 在 $0.35 \sim 0.45\mu\text{m}$,栅氧厚度 T_{ox} 在 $0.005 \sim 0.01\mu\text{m}$ 。在其中搜索合适的点得到所希望的驱动电流 I_{on} 和关断电流 I_{off} 。可接受的相对误差设置为 0.01。测试了不同大小的可行空间。

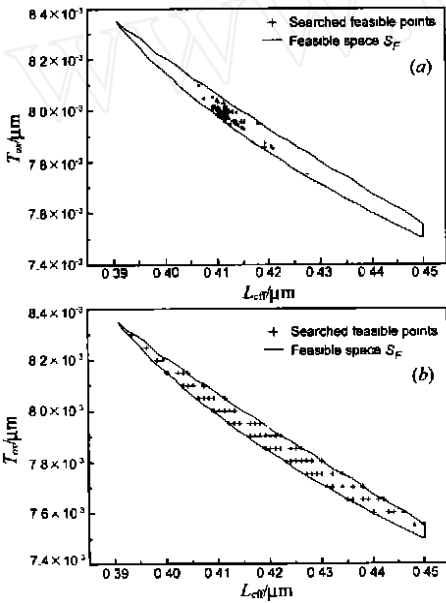


图 3 两种算法得到的结果比较。(a) GENOCOP 的结果 ($R_c=21\%$); (b) 改进的算法的结果 ($R_c=72\%$)

$I_{on} = 2e - 3A$, $I_{off} < 4.3E - 13A$, 有 $S_F/S_D = 0.98\%$

得到的两种算法的解空间覆盖率 R_c 见表 1, 可以看到, 改进的算法的平均 R_c 是 GENOCOP 的 2.89 倍, 有能力得到较全面的准可行空间。

表 1 解空间覆盖率 R_c 的比较 ($S_F/S_D = 0.98\%$)

选用算法	GENOCOP	改进的算法
最高 R_c	38 %	85 %
最低 R_c	11 %	16 %
平均 R_c	19.4 %	56 %

图 3 给出了两个算法中来自 10 次运算中解空间覆盖率排第三的结果。其中每个参数方向按 200 等分划分。“+”号数

据为算法得到的可行点,“-”内的区域为作为参考的可行区 S_F 。可看到 GENOCOP 的解过分集中,覆盖率 R_c 很小。而改进后的结果能更好地反映可行空间。

(2) $I_{on} = 2e - 3A$, $I_{off} < 4.04E - 13A$, 有 $S_F/S_D = 0.014\%$

得到的两种算法的解空间覆盖率 R_c 见表 2。可以看到, 在 S_F/S_D 较小时, 改进的算法仍然有效, 这里的平均 R_c 是 GENOCOP 的 2.87 倍。

表 2 解空间覆盖率 R_c 的比较 ($S_F/S_D = 0.014\%$)

选用算法	GENOCOP	改进的算法
最高 R_c	28 %	55 %
最低 R_c	7 %	16 %
平均 R_c	12.4 %	35.6 %

5 结论

本文改进了遗传算法 GENOCOP, 考虑了器件参数受工艺精度影响的特点将设计空间转换为整型空间, 并加入适应性复合算子来合理的开发和探索准可行空间。实验证明其保持 GENOCOP 有效快速搜索到可行解的特性的同时, 能有效的提高解空间的覆盖率 R_c , 有能力得到更为全面的准可行空间。在器件设计时, 用户对一些重要参数有更大的选择余地。比如在对特性要求不高时, 可以选择可行区间靠中间的设计点获得较大的工艺设计余量; 或者受工艺精度限制较小时, 选择具有较好的特性的设计点。从而使得 MOS 器件综合系统更加实用。

参考文献:

- [1] P Mbens, M Tack, H Van hove, M Vermandel, D Bolognesi. Development of an optimised 40V pDMOS device by use of a TCAD design of experiment methodology [A]. SISPAD [C], 2000:276 - 279.
- [2] Harold H Hosack, et al. Recent advances in process synthesis for semiconductor devices [J]. IEEE ED, 1998, 45 (3): 626 - 633.
- [3] Li Zhao, Xie Xiaofeng, Zhang Wenjun, Yang Zhilian. Realization of device decomposition for technology synthesis with the genetic algorithm [A]. Conference ICDA2000 of the 16th IFIP WCC2000 [C], August 2000:355 - 359.
- [4] Z Michalewicz. Genetic algorithms + Data structures = Evolution programs [M]. Springer-Verlag, 1994.
- [5] R Wehrens, E Pretsch, L M C Buydens. The quality of optimisation by genetic algorithms [J]. Analytica Chimica Acta, 1999, 388:265 - 271.
- [6] 任庆生, 叶中行, 曾进, 戚飞虎. 遗传算法中常用算子的分析 [J]. 电子学报, 2000, 28(5): 113 - 114.
- [7] B M Kim, YB Kim, C H Oh. A study on the convergence of genetic algorithms [J]. Computers ind. Engng, 1997, 33:581 - 588.

作者简介:



谢晓锋 男, 1975 年生于湖南。清华大学微电子所博士。当前主要研究半导体器件综合的设计和实现。