# 文献与学习总结

# Efficient topology optimization in MATLAB using 88 lines of code

Ole Sigmund提出一种高效的拓扑优化方法，这是基于99行拓扑优化代码而改进的。原始代码已通过密度滤波器扩展，并且主要通过预分配阵列和矢量化循环来实现效率的显著提高。对于具有7500个元素的基准示例，获得了因子为100的速度改进。此外，代码的长度已经减少到仅仅88行。这些改进是在不牺牲代码可读性的情况下完成的。本文还讨论了基本代码的简单扩展，包括最近的基于PDE和黑白投影滤波方法。

# A simple and compact Python code for complex 3D topology optimization

作者提出了一种用于一般三维拓扑优化的100行Python代码。该代码采用ABAQUS脚本接口，提供先进的有限元分析（FEA）的方便访问。它采用双向进化结构优化（BESO）方法求解体积约束下的柔度最小化问题。源代码由控制迭代过程的主程序和实现输入模型准备的五个独立函数FEA组成。独立于网格的滤波器和BESO算法。该代码从模型数据库(.cae文件)读取初始设计，该数据库可以是Abaqus/CAE中生成的任意3D几何图形，或者从各种广泛使用的CAD建模包进行转换。文中还给出了多个负载情况和非线性的扩展的代码。文章表示通过进一步的扩展，该代码可以解决结构工程、机械工程和建筑实践中复杂的三维概念设计问题。

Topology Optimization Design of 3D Continuum Structure with Reserved Hole Based on Variable Density Method

作者建立了三维连续体结构拓扑优化柔度最小的目标函数，以搜索受预定体积约束的最优材料分布。基于改进的SIMP模型和新的灵敏度滤波技术，推导了三维有限元分析的基本迭代方程，并用优化准则法求解。以上程序均采用MATLAB语言编写，并对拓扑优化设计实例进行了说明。通过观察各种指标，包括柔度、最大位移和密度指数，对含预留孔的三维连续体结构进行了反复试验。分析了网格、惩罚因子和滤波器半径对拓扑结果的影响。计算结果表明，网格数越细或越粗，柔度、最大位移和密度指数越大。当滤波半径大于1.0时，拓扑形状不再表现为棋盘问题，说明本文提出的灵敏度滤波方法是有效的。惩罚因子应该是整数，因为迭代步骤在非整数时大大增加。上述改进的变密度方法为今后更复杂的三维连续体结构的拓扑优化设计提供了技术路线。

Continuous Optimization of Adaptive Quadtree Structures

# 自适应四叉树结构的连续优化

本文提出一种新颖的连续优化方法来应对四叉树优化的离散问题。优化目的在于形成具有高机械刚度的四叉树结构，四叉树的边界被认为承载机械负载的结构元素。我们将四叉树优化看做一个材料连续分布的问题。离散化的设计变量被四叉树分层的多级的连续变量代替。在离散化四叉树优化中，如果一个单元的复习单元更新的话，那么这个单元就符合更新的要求。我们提出对连续多级设计变量的这种依赖关系进行连续模拟，并将其集成到迭代优化过程中。结果表明连续优化的四叉树比传统模式和启发式优化同行表现出更高的刚度。我们将自适应结构用于3D打印零件的轻量化填充，尤其是统一的几何模型的零件。

连续四叉树优化

一个矩形设计区域，首次平均划分方格，每个方格的边界单元密度均为1（见文中图2（b）），进行优化，将实心方格再次平均划分，进行优化，以此类推，进行优化，最后得到符合要求刚度的图形。

88行拓扑优化是把这个矩形区域一次性划分若干元素，依次进行优化，也可以说划分的元素越多，优化越精确，得到的图形是圆滑的，而自适应四叉树逐层划分，逐层优化，得到的图形是一个个方框组成，有空白，又实心，作者也提出该方法适用于模型填充。

# The making of SPINdle

本文介绍了SPINdle的设计和实现，一个基于开源Java的可废弃逻辑推理器，能够对可废弃的逻辑理论进行有效和可扩展的推理。包括可废弃逻辑的标准和模态扩展。它可作为独立的理论证明者使用，并且可以作为可废弃的逻辑规则引擎嵌入到任何应用程序中。它允许用户或代理在给定的知识库或其他应用程序即时生成的理论上发出查询，并自动生成其结果的结论。

# 1、SPINdle最重要的特征：

1. 它支持所有可废止逻辑的规则类型，如事实，严格规则，可废止规则，失败，优越。
2. 它支持带有模态运算转换符的模态可逆逻辑。
3. 它支持否定和冲突（互斥）文字。
4. 可以使用XML和纯文本表示一个理论，这个理论和它的扩展也是用XML表示。
5. 视觉理论编辑器用于编辑标准可废止逻辑理论。

# 2、可废止逻辑

可废止理论D包括三个部分，F表示事实陈述，R表示规则，描述事实与结论之间的关系，规则中有三个规则，即严格规则，可废止规则和失败。严格规则事实是结论，可废止规则是可用相反证据打破这一规则，失败即得不出结论。优于关系（>）用于确定规则之间的优先性，即一个规则要优于另一个规则，先执行优先级高的规则。

2.1模态可废止逻辑

模态逻辑应包括两部分：命题基本的基础逻辑结构和模态运算符的逻辑行为。

# 3、实现

SPINdle系统框架：规则解析器、理论规范化器和推理机。规则解析器用于将保存的理论文档转换成可用于下一模块的数据。加载一个理论到SPINdle，用户就可以根据需求更改这一理论。I/O界面会出现一些方法帮助用户在数据库中加载和保存理论。理论也可使用XML导出代理通信。

3.1推理过程

整个推理过程有两相：预处理相--将描述的技术理论转换为没有优势关系和失败的等效理论，有助于简化推理引擎中的推理过程。理论规范化模块包括三个线性转换：一是转换为常规形式，一是用于清空优势关系，一是用于清空失败推理；还将具有多头的规则转换具有单头的等效规则集。预计转化理论将在他们转化的理论语言中产生相同的结论集。结论生成相--基于一系列（理论）转换，允许我们确定文字是否可证明（及其推导的强度），并逐步减少和简化理论。

最后，结论要么输出给用户，要么保存在理论中数据库供以后使用。

1. 结论

我们提出了SPINdle，一个可在标准可废止逻辑和模态可废止逻辑上推理的推理器。SPINdle的未来发展方向包括算法和技术改进，其中包括支持从Web直接导入和处理的接口OWL/RDF数据和RDF模式本体。也包括时间可废止逻辑的推理。

# 规则驱动的体结构演化设计

本文提出一种规则驱动的体结构演化方法。首先定义模型中每个体素及其邻域为一个智能主体，将模型内部结构装配问题转化为一个多智能主题系统的问题，然后将物理约束和人工策略定义为设计规则，通过引入可废止逻辑表达不完备的知识和约束，最后通过群体涌现行为生成全局结构。实验结果表明该方法能够在相同规则的控制下根据不同的边界条件演化出有针对性的结构，也可以在相同条件下根据不同的规则得到相异的设计。

文章指出下一步的研究将会拓展演化规则，通过建立面向实际工程问题领域知识的规则库，测试不完全领域知识形成的规则集合作用于设计过程中产生的冲突问题，以进一步检测方法的能力和扩展性。同时继续完善局部结构编辑方法，以实现在不同尺度上进行结构优化，并通过与现有拓扑优化方法对比和理论分析，阐明复杂交互行为对算法的收敛性产生的影响。

Simp方法中密度更新使用OC方法，而现在将数学计算方法改为演化算法，来逐步更新每个元素的密度值，以此降低数学计算。比较数学计算与演化算法各个性能，有收敛性能的比较......