策略逻辑：

◆研究背景

优化问题是金融中基础、不可避免的问题，从均值方差的二次规划开始，

优化问题已经深入到金融领域的方方面面，从大类资产配置到组合优化、从理

论模型下的效用最大化再到实战模型的参数优化，都用到优化技术。而很多优

化问题较为复杂，非凸、不连续、不可导、高维、随机、约束过多等问题给数

值计算带来困扰，本文提出次优理论并且介绍差分进化算法，通过展示差分进

化算法的良好效果，希望给广大投资者的量化建模带来一丝启示。本文方法对

组合优化、大类配置、FOF 组合构建、智能投顾等领域都会有所帮助。

◆次优战胜最优

作者在长期建模的经验中斗胆提出金融次优理论，其实金融没有次优理

论，运筹学也没有，只有福利经济学中有过次优理论的描述，文中提出的次优

理论和经济学中的次优理论有一点类似，我们把这个概念借来。其实很简单：

当期优化的最优解不一定是下一期的最优，而样本内的次优在样本外可能战胜

样本内的最优。所以，从金融投资角度看，优化问题下的最优解不一定是我们

想要的，因为我们的目标是获得较好的样本外收益表现。

◆差分进化算法

差分进化 Differential Evolution（DE）由 Storn 等人于 1995 年提出，和其它

演化算法一样，DE 是一种模拟生物进化的随机模型，通过反复迭代，使得那

些适应环境的个体被保存了下来。但相比于遗传算法，DE 保留了基于种群的

全局搜索策略，采用实数编码、基于差分的简单变异操作和一对一的竞争生存

策略，降低了遗传操作的复杂性。同时，DE 特有的记忆能力使其可以动态跟

踪当前的搜索情况，以调整其搜索策略，具有较强的全局收敛能力和稳健性，

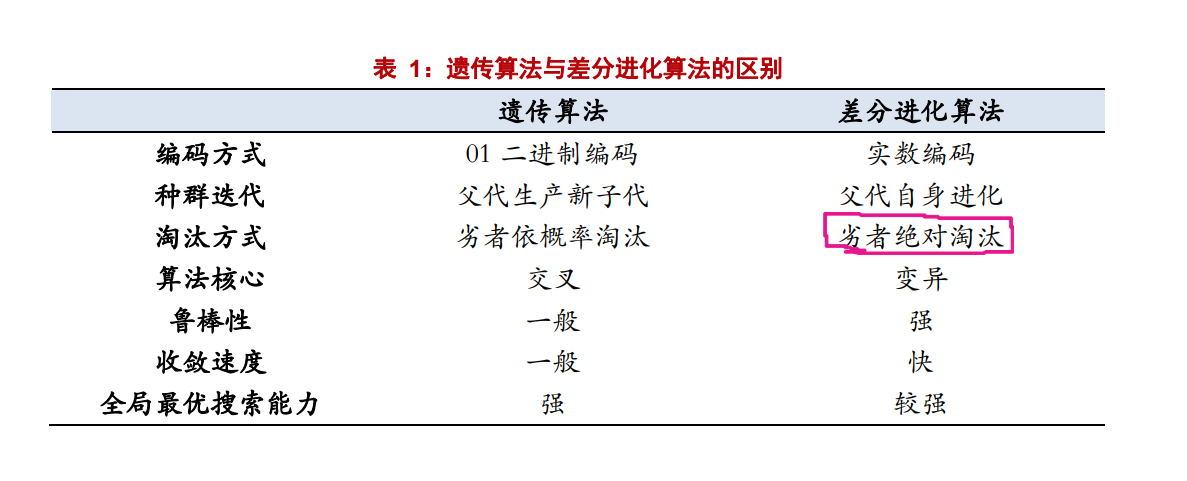
且不需要借助问题的特征信息，适于求解一些利用常规的数学规划方法所无法

求解的复杂优化问题。

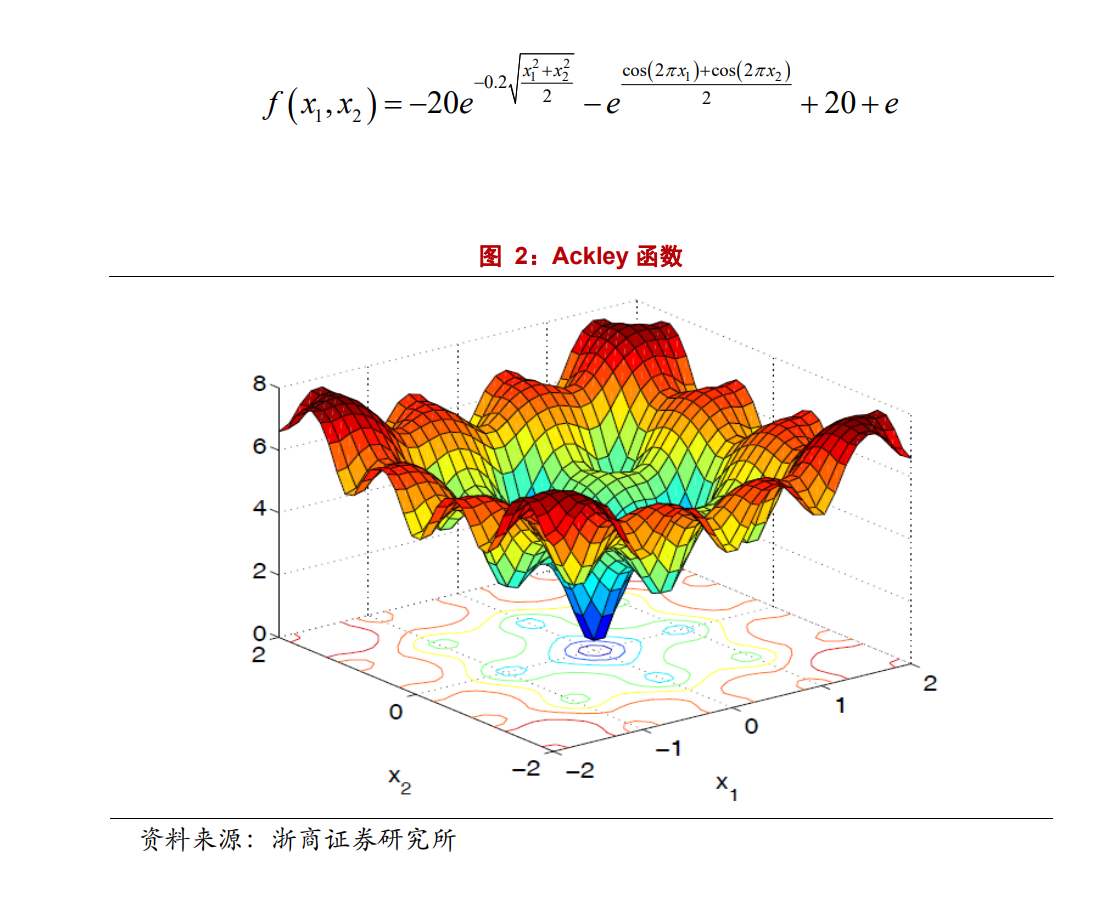
代码模型及简要：

1. DE\_algorithm：编写差分进化算法

差分进化 Differential Evolution（DE）由 Storn 等人于 1995 年提出，和其它演化算法一样，DE 是一种模拟生 物进化的随机模型，通过反复迭代，使得那些适应环境的个体被保存了下来。但相比于进化算法，DE 保留了基 于种群的全局搜索策略，采用实数编码、基于差分的简单变异操作和一对一的竞争生存策略，降低了遗传操作的 复杂性。同时，DE 特有的记忆能力使其可以动态跟踪当前的搜索情况，以调整其搜索策略，具有较强的全局收 敛能力和鲁棒性，且不需要借助问题的特征信息，适于求解一些利用常规的数学规划方法所无法求解的复杂环境 中的优化问题。 目前，DE 已经在人工神经元网络、化工、电力、电磁、机械设计、机器人、信号处理、生物信息、经济学、 现代农业、食品安全、环境保护和运筹学等许多领域得到了应用。例如：在神经网络训练中，有研究将差分进化 方法作为前馈神经网络的候选全局优化方法，用于验证已达到的最优值，以及用于开发不一定提供梯度信息的正 则项和非常规传递函数；在电磁学中，差分进化被用于解决以非凸函数和连续空间为特征、涉及大量未知数的优 化问题，有研究提出了电磁学中与天线合成和逆散射有关的基于差分进化的方法；在电力系统中，差分进化算法 用于解决电力系统中经济负荷分配（ELD）问题，与其他现有技术相比，基于差分进化算法的方案更有效；在机 器学习中，差分进化算法在大型未标记数据集自动聚类中取得了应用，与大多数现有的聚类技术相比，基于差分 进化所提出的算法不需要先验知识就可以对数据进行分类，并通过与两种最新开发的分区聚类技术和一种流行 的分层聚类算法进行比较可以证明基于 DE 的新方法的优越性。

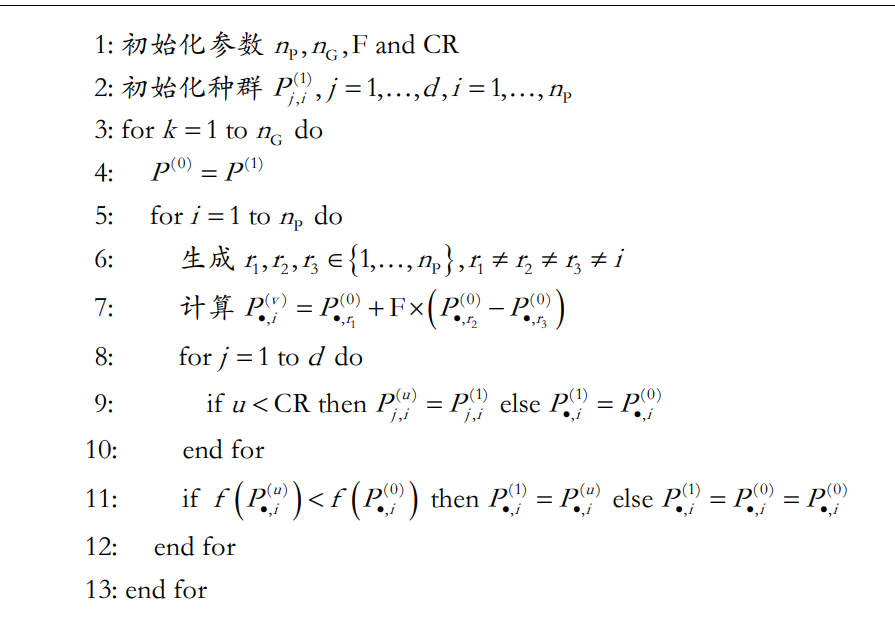


差分进化算法（Differential Evolution algorithm)是的一种针对连续函数、基于种群的启发式优化技术。该算 法通过加、减和交叉来更新解向量的种群，然后在原始和更新的种群中选择最合适的解。我们通过最小化 Ackley 函数来说明算法的工作原理（Ackley,1987）。



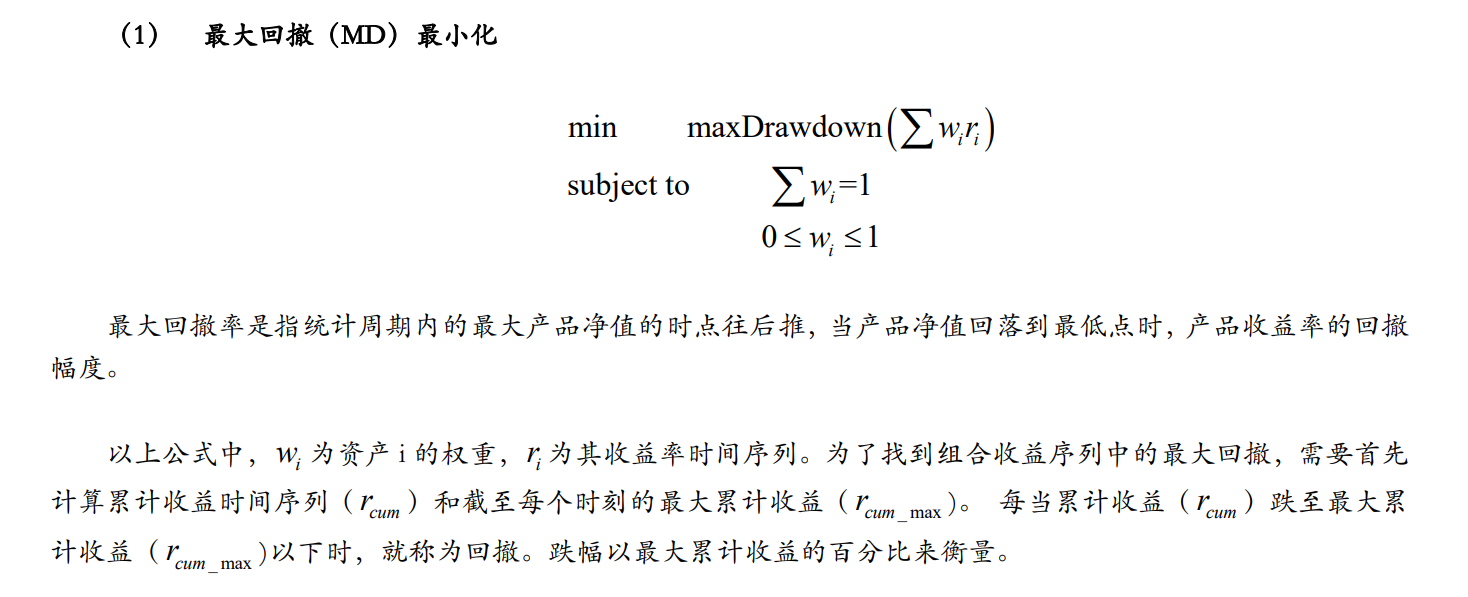
算法伪代码为：

1. 初代种群 该算法首先选择 P n 个随机选择的解。初始种群的 P n 个解由 P d n × 维矩阵 (0) P 表示， d 是函数域的维数。
2. （二）新解的构建 对于由矩阵 (0) P 的列表示的每个解 P ii n , 1, , = … ,该算法从三个随机选择的列（解） 123 rr r , , 构造一个新解。这一过程由四个步骤完成： 第一步：构建向量 ( ) 2 3 (0) (0) FP P × − • • , , r r ，其中 F 是给定的缩放因子第二步：构建向量 ( ) 1 23 ( ) (0) (0) (0) , , ,, v P P FP P • • •• i r rr = +× −第三步：按以下规则组合 (0) P•,i 和v 来构造 (P P u ⎧⎪ ≤ = ⎨ ⎪ > ⎩ 值得注意的是，向量 ( ) , u Pj i 中至少有一个元素 j 是来自于 ( ) •, v P i 。第四步：新种群中第i 个解由下决定：



1. DE\_strategy\_demo.py：

采取最大回撤（Maxdrawdown）最小化的目标



网格寻参，选出差分进化算法最佳参数，计算每季度的最佳持仓

对每季度的持仓情况回测该季度的结果

参考代码及原理出处：

<https://github.com/hugo2046/QuantsPlaybook/tree/master/D-%E7%BB%84%E5%90%88%E4%BC%98%E5%8C%96/DE%E7%AE%97%E6%B3%95%E4%B8%8B%E7%9A%84%E7%BB%84%E5%90%88%E4%BC%98%E5%8C%96>