

多策略组合遗传算法生成随机音乐序列

成员分工

姓名	分工
孟	撰写遗传算法的基础模板
孙	探究关于时值的适合度函数的实现思路
荣	探究关于音高的适合度函数的实现思路
黄	整合组员思路、进行多种策略的实现与分析、横向对比各种变异方式、适合度函数设计、生成的最终旋律效果

算法背景

遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

1967年，美国密歇根大学J. Holland教授的学生Bagley在他的博士论文中首次提出了遗传算法这一术语，并讨论了遗传算法在博弈中的应用。

1975年，J. Holland等提出了对遗传算法理论研究极为重要的模式理论，出版了专著《自然系统和人工系统的适配》，在书中系统阐述了遗传算法的基本理论和方法，推动了遗传算法的发展。

在二十世纪八十年代中期之前，对于遗传算法的研究还仅仅限于理论方面，直到在伊利诺伊大学召开了第一届世界遗传算法大会。

随着计算机计算能力的发展和实际应用需求的增多，遗传算法逐渐进入实际应用阶段。1989年，纽约时报作者约翰·马科夫写了一篇文章描述第一个商业用途的遗传算法——进化者。之后，越来越多种类的遗传算法出现并被用于许多领域中。

遗传算法主要由选择、交配和突变三部分完成，通过多次选择、交配和突变的迭代，算法有极大可能找到全局最优解或较优解。

与模拟退火[1]等一系列随机算法类似，遗传算法通常被用于在极大的空间范围内去寻找空间最优解。作为一个在全空间范围内的以自然进化与生物变异为模板的随机算法，遗传算法具有极好的适应性与针对性——既可以适应不同的环境来工作，也能够在针对某一评价函数找出较好的空间最优解。由于遗传算法的适应性与针对性，以及其在极大空间内较好的效率，只需要找到合适的评价函数，我们就可以选择使用遗传算法来实现机器作曲的功能。

算法设计

经典遗传算法的流程用自然语言描述大致可以分为四个步骤：

1. 产生种群
2. 计算适合度
3. 重组、突变
4. 自然选择
5. 重复1-4，直至满足终止条件

其中第四步中如何对种群中的个体进行筛选是算法设计的重点，直接决定了能产生的音乐序列的动听程度。因此我们在适合度函数中对音乐的和谐度、连续性、多样性、复现性、合法性等进行了评分，并将这些评分结果

加权求和，得到一个音乐序列的适合度。相关参数与说明见表 2与表 4。

在此基础上，我们还为遗传算法设计了一系列的优化方法以提升算法的收敛速度与最终结果的质量。算法整体实现逻辑见图 1。

自适应突变 (Adaptive Mutation)

在一般的遗传算法中，突变率是一个固定的值，即每代个体中所含有突变的个体数期望是一致的，这也导致了种群在收敛到局部最优解的同时失去了泛化能力，因此也就无法搜寻到参数空间中的更优解乃至全局最优解[2]。因此我们从自然界中存在的自适应突变[3]取得灵感，为整个流程设计了能够根据当前收敛状态自动调整突变率的算法。

$$AMR = BMR \times \min \left(1 + \frac{SGN}{AIN}, MMR \right)$$

AMP:=Adaptive Mutation Rate

自适应突变率

BMR:=Base Mutation Rate

基础突变率

where BMR:=Maximum Mutation Rate

最大突变率

SGN:=Same-best Generation Number

最佳纪录保持代数

AIN:=Adapt Interval

调整间隔

参考突变 (Reference Mutation)

遗传算法能够迭代产生更优结果的能力主要来自于种群中个体间的重组和代际间的突变。在遗传算法搜索参数空间的过程中，如果按照 Darwin 模型则每次突变产生的新基因组与原基因组应当是无关的，这样就导致每次突变时将原基因组上对应位点的信息全完抛弃。参考自然界中单个碱基突变时更容易发生转换而非颠换[4]，以及 Baldwinian 在早年提出的模型[5] [6]，我们假设每个个体的基因组都不仅对应了当前的状态，还应当包含了梯度方向的信息，因此我们设计了一种参考突变算法，限制了每次突变产生的新基因组与原基因组的距离，更好地借用原基因组的信息，防止种群过度发散。

轮盘赌选择 (Roulette Wheel Selection)

在由亲代种群生成子代种群时，如何选择亲代很大程度上决定了整体收敛速度，有时甚至会对算法整体能力造成巨大的影响。这里我们参考已有的一些方法[7]，主要选择了两种基于适合度的选择方法，即轮盘赌选择

(Roulette Wheel Selection) 与用于对比的顺序选择 (Sequential Selection, 后称为 Best Mate Worst) 进行测试。顺序选择较为直观，即由第一名与最后一名杂交、第二名与倒数第二杂交，以此类推。对于轮盘赌选择法我们在其原来基础上进行了一些改进，在其之前加上了一层 Softmax 函数用于归一化选择概率，使得概率对数值正负以及小范围波动不敏感。

$$SP_j = \frac{\exp \frac{f_j}{SF}}{\sum_i \exp \frac{f_i}{SF}}$$

SP_j :=Selection Probability of j-th Individual

第 j 位个体的选择概率

where SF:=Scaling Factor

缩放因子

f_j :=Fitness of j-th Individual

第 j 位个体的适应度

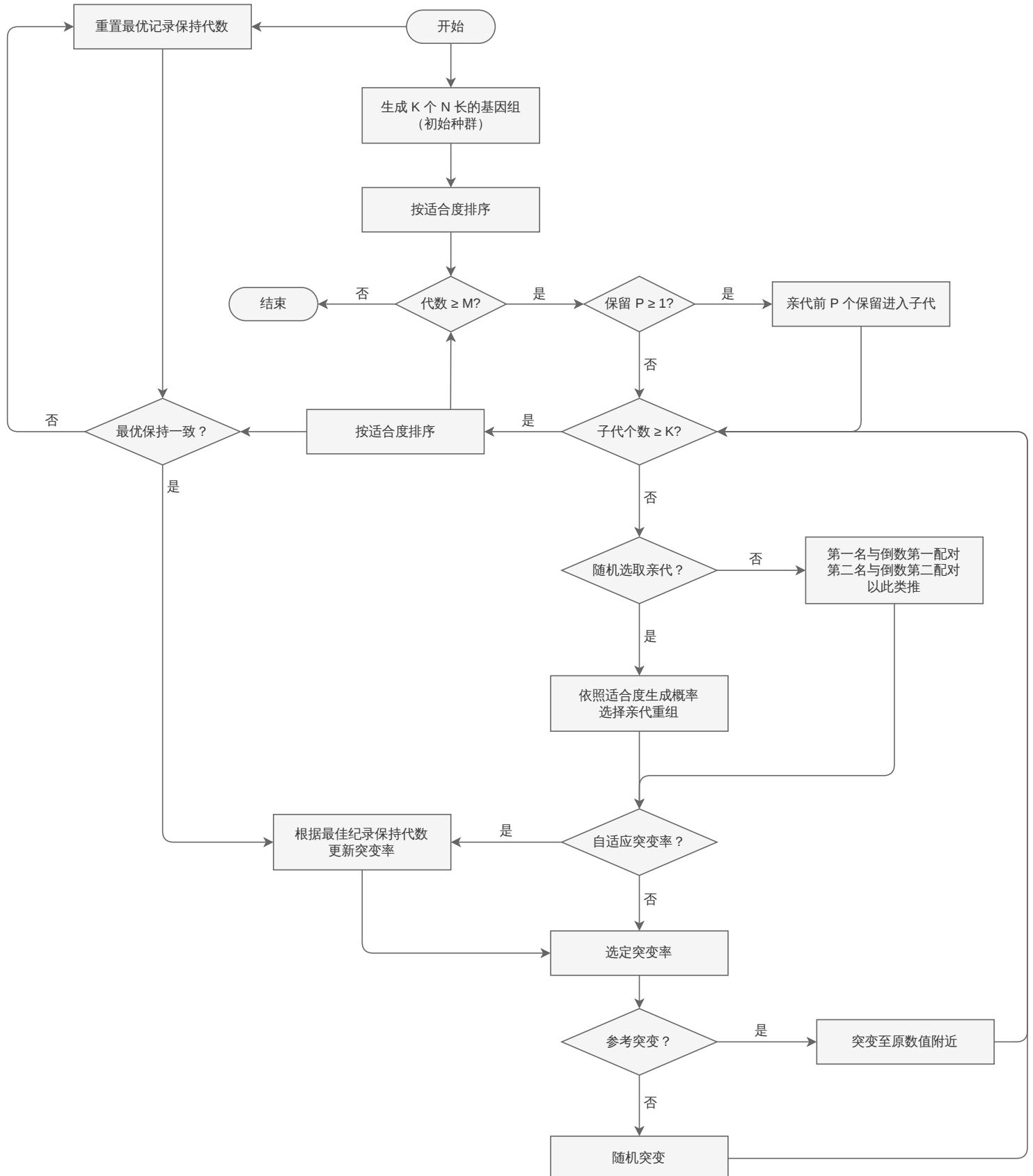


图 1: 算法决策流程图

参数列表

表 1: 音乐生成参数

参数名	数值	说明
PitchRangeLow	36	音高范围下界
PitchRangeHigh	73	音高范围上界
DurationRange	8	时值范围上界

BeatsPerBar	4	每小节拍数
DurationPerBeat	4	每拍时值

表 2: 目标期望参数

参数名	数值	说明
MediumFrequencyScale	4	平均音高所在音阶
RhythmVariation	2	节奏类型数
PitchVariation	6	音符种类数
SameDegreeProportion	0.4	一度音程比例
PitchJumpDistance	6	大于此半音数为音高跳跃
PitchSpan	12	音高最大距离

表 3: 算法超参数

参数名	数值	说明
N	60	生成音符数
M	100	迭代数
K	1000	每代个体数
P	1	保留上一代前 P 个最优
MutationRate	50	(千分数) 突变率
CrossOverRate	50	(千分数) 重组率

表 4: 权重列表

参数名	数值	说明
Weight_ChromaticNumber	50	相邻音符相差半音-罚分
Weight_ConsecutiveJumps	50	连续音高跳跃-罚分
Weight_HalfNote	100	小节最后一音符跨越节-罚分
Weight_MediumFrequencyScale	500	平均音高落于期望音阶-加分
Weight_OutOfScale	25	音符不在同一音阶-罚分
Weight_HighFrequencyComponent	5	音高高频变化的成分-罚分
Weight_PitchDevition	5	音高方差-罚分
Weight_PitchDiversity	100	音符种类多样性-加分
Weight_PitchJump	75	音高跳跃-罚分
Weight_PitchSpan	500	音高最大距离-加分
Weight_RhythmDiversity	200	节奏多样性-加分
Weight_RhythmSimilarity	10	节奏复现-加/罚分
Weight_RhythmVariationByBar	200	小节节奏型多样性-加分

Weight_ScaleConsistency	50	音高同属一类音阶-加分
Weight_StableConsonance	15	相邻音高和谐-加/罚分
Weight_StepWise	500	一度音程比例-加分

权重数值大小与优先级、函数计算方法均有关

表 5: 自适应突变率参数

参数名	数值	说明
beta_GenerationInterval	10	最优每保持此代数则增加突变率
beta_MaxMultiplier	10.0	突变率增加倍数上限

表 6: 参考突变参数

参数名	数值	说明
beta_PitchMutationRange	6	音高突变的邻域范围
beta_DurationMutationRange	2	时值突变的邻域范围

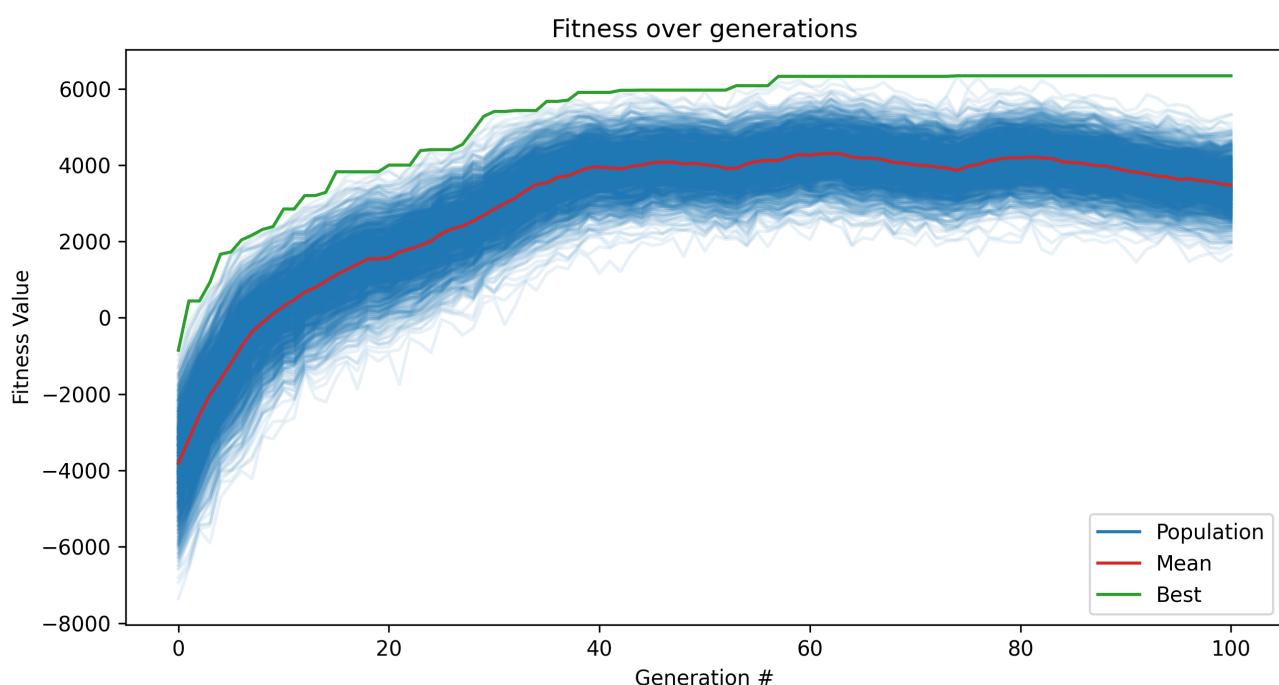
结果

音乐试听

[MIDI 链接](#)

标准策略稳定产出较好结果

运行结果可参见[附录](#)，标准策略组合下种群平均在 50 代左右进入相对稳定的状态。值得注意的是如下图的结果：



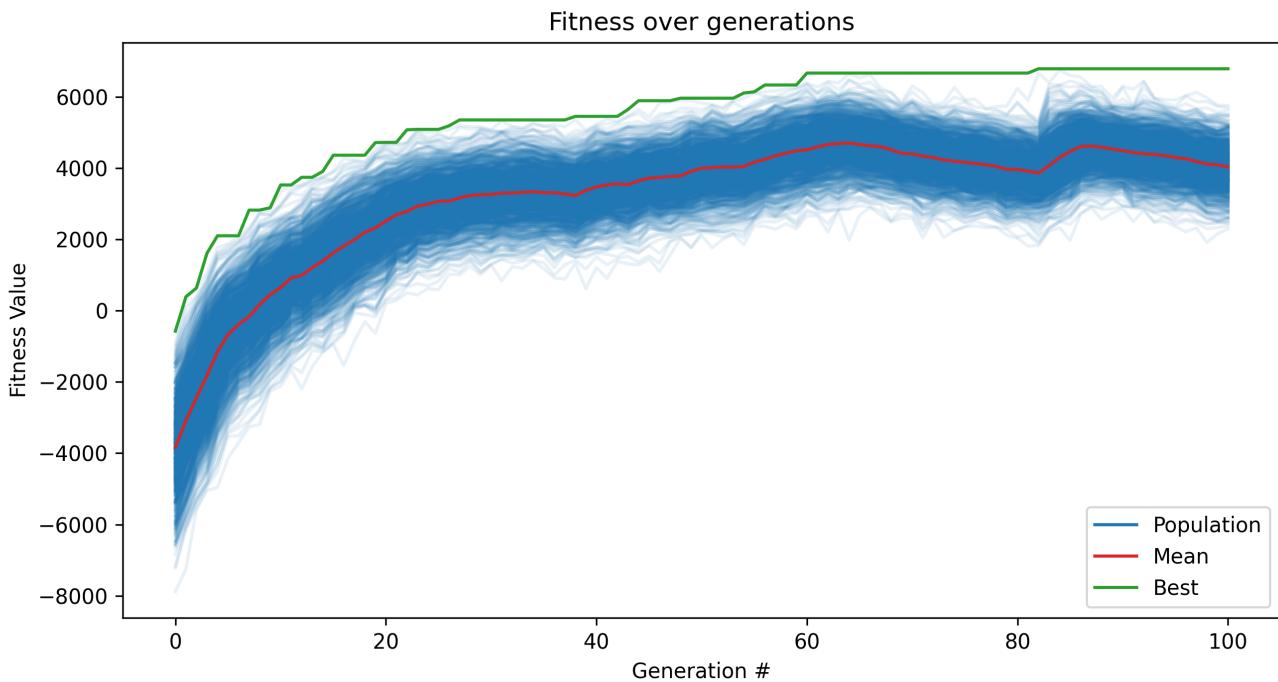
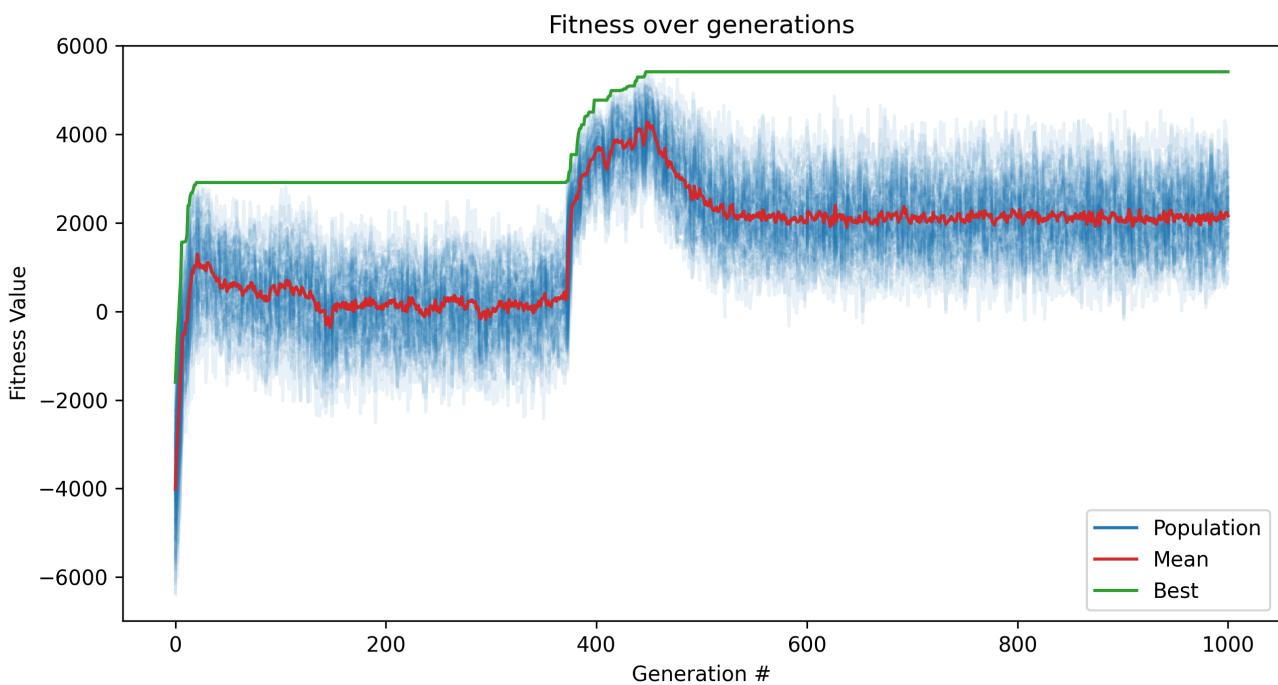


图 2: 标准策略组合下的非达尔文式进化

当迭代至约 60 代时，由于自适应突变率的上升和杂交策略的保留，种群进一步发散搜索到了更优化的音乐组合，类似于非达尔文的跳跃式进化[8]。即便我们在保持总个体数不变的情况下，减少每代个体数目，增加迭代代数，这样的现象依然普遍存在；只是此时由于每代个体数限制，更难收敛到比标准策略组更好的结果：



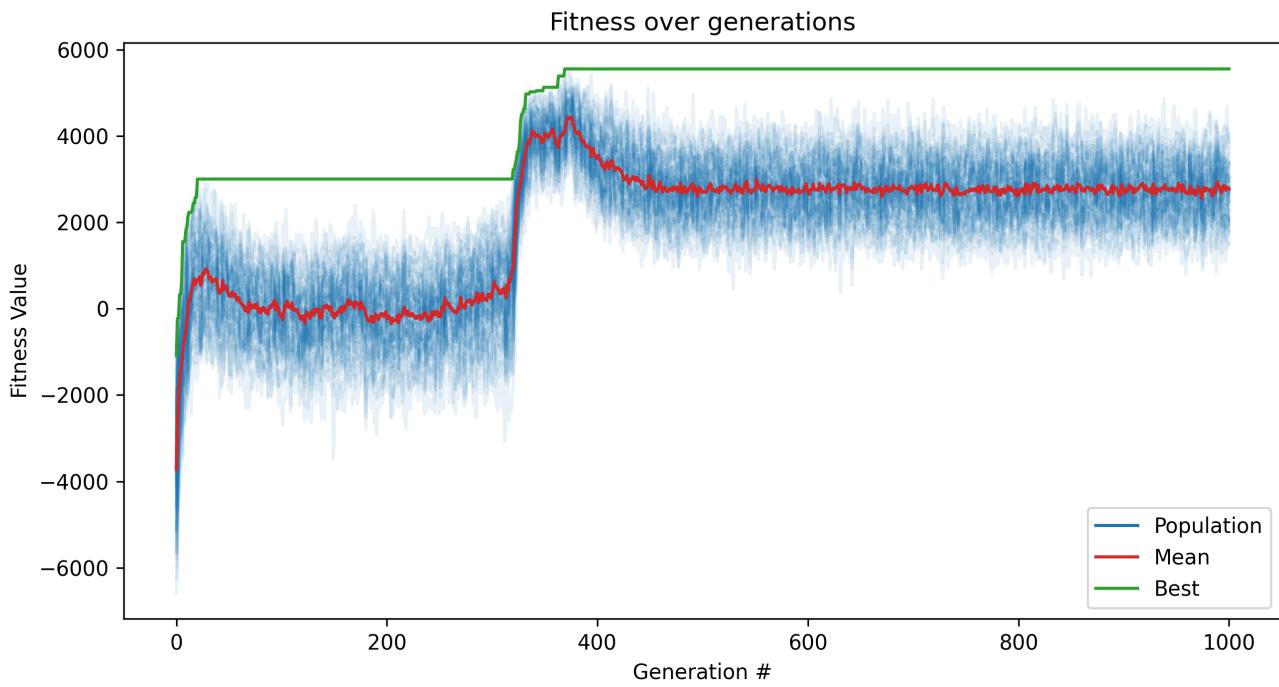
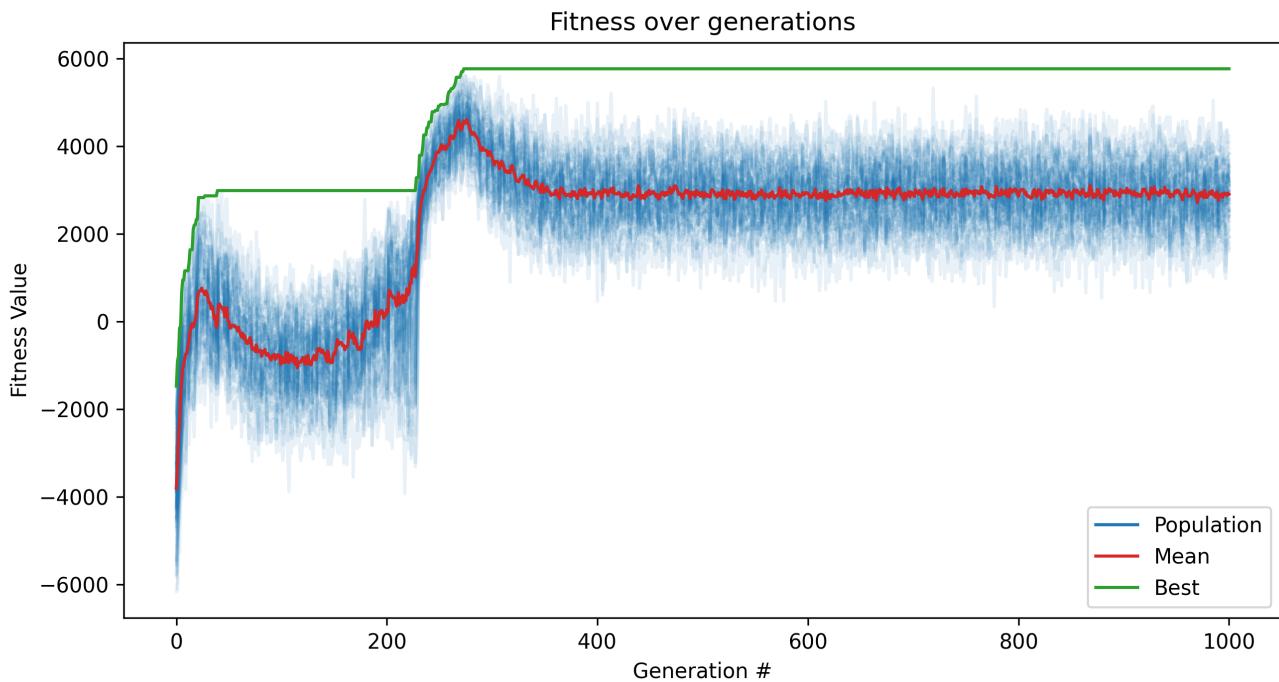


图 3: 更多代数策略下的非达尔文式进化

这样的跳跃式进化存在的恰恰说明了标准策略不仅能够快速、稳定地收敛至较好的结果，还具备一定的搜索能力，也正是因此使其具有较为独特的防止过拟合、寻找全局最优解的能力，这是在缺少或者改动任一策略时都无法达成的。

标准策略组合兼顾收敛速度与游走能力

为了进一步说明标准策略组合具备这样的独特能力，我们通过控制单个策略的变化，比较了不同遗传策略下适合度随代数的变化，结果见图 4 至图 9。

标准策略组合 (Standard)

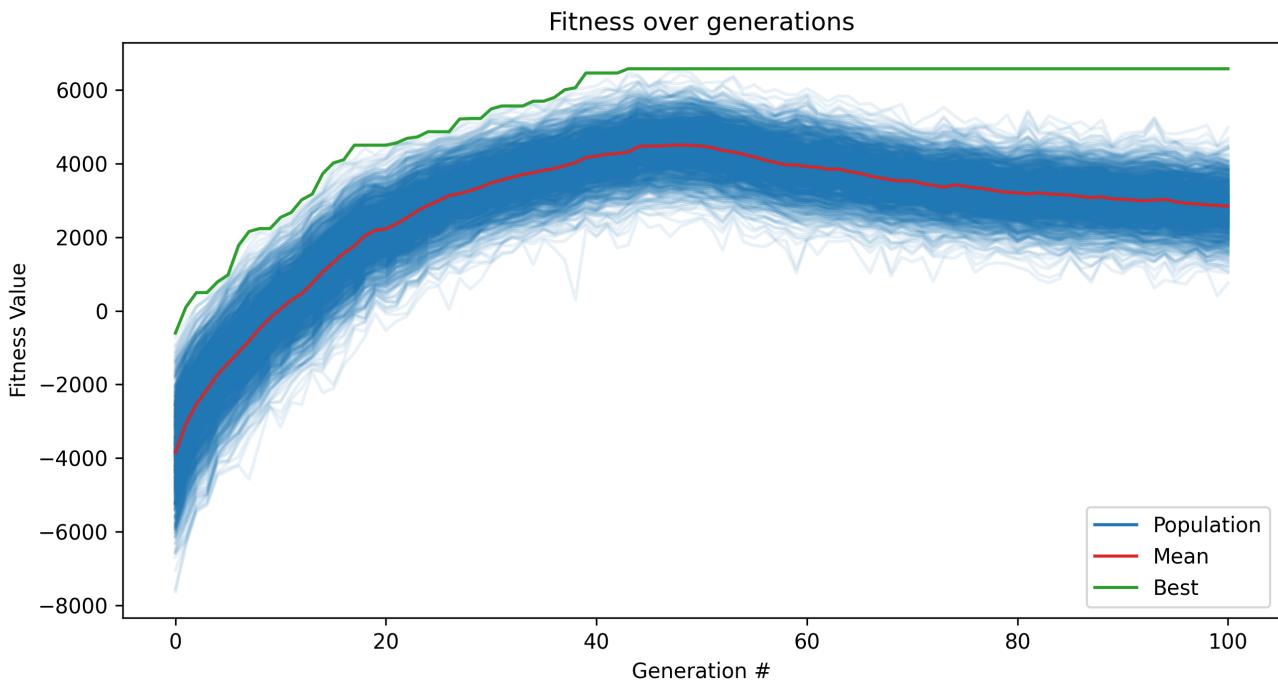


图 4: 标准策略下种群适合度变化曲线

由图 4 可见，结果相对较好，种群最优在 50 代左右稳定，此后其余种群逐渐发散寻找参数空间中更优解，防止过拟合。

固定突变率与重组率 (Fixed Mutation Rate)

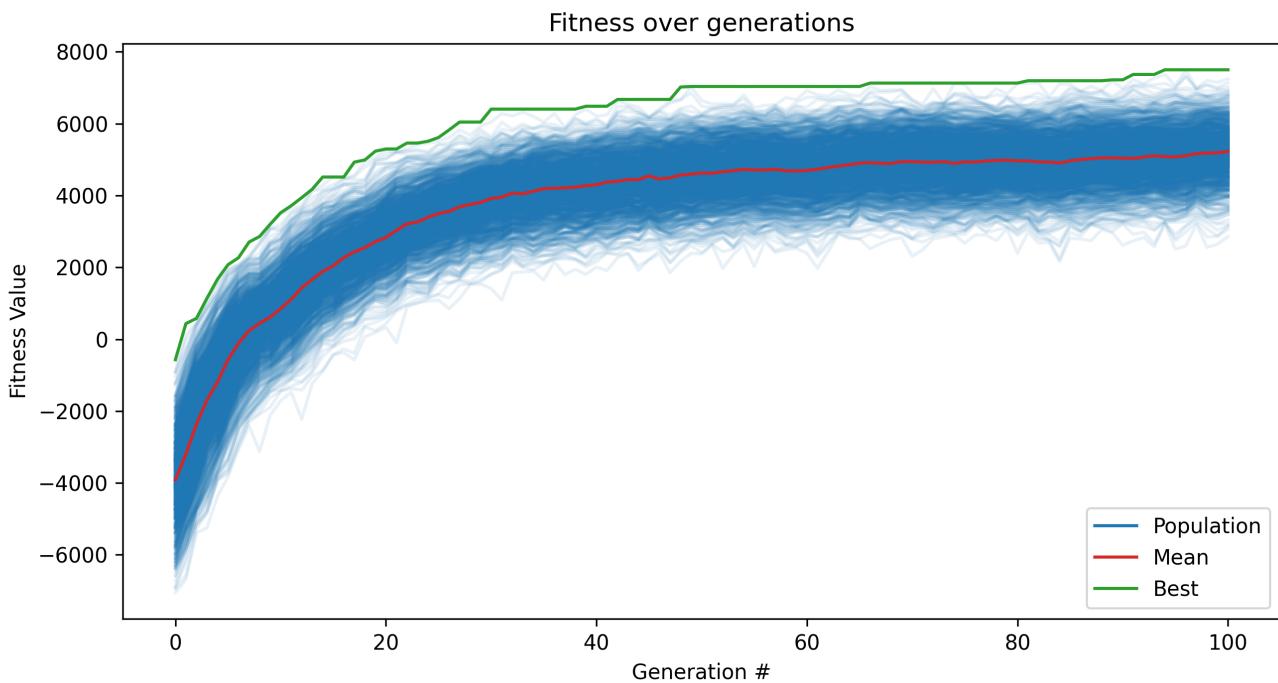


图 5: 固定突变率与重组率策略下种群适合度变化曲线

由图 5 可得知，该策略下种群同样在 50 代左右稳定，但是由于突变率限制，容易陷入局部最优解导致过拟合。

自由随机突变 (Random Mutation)

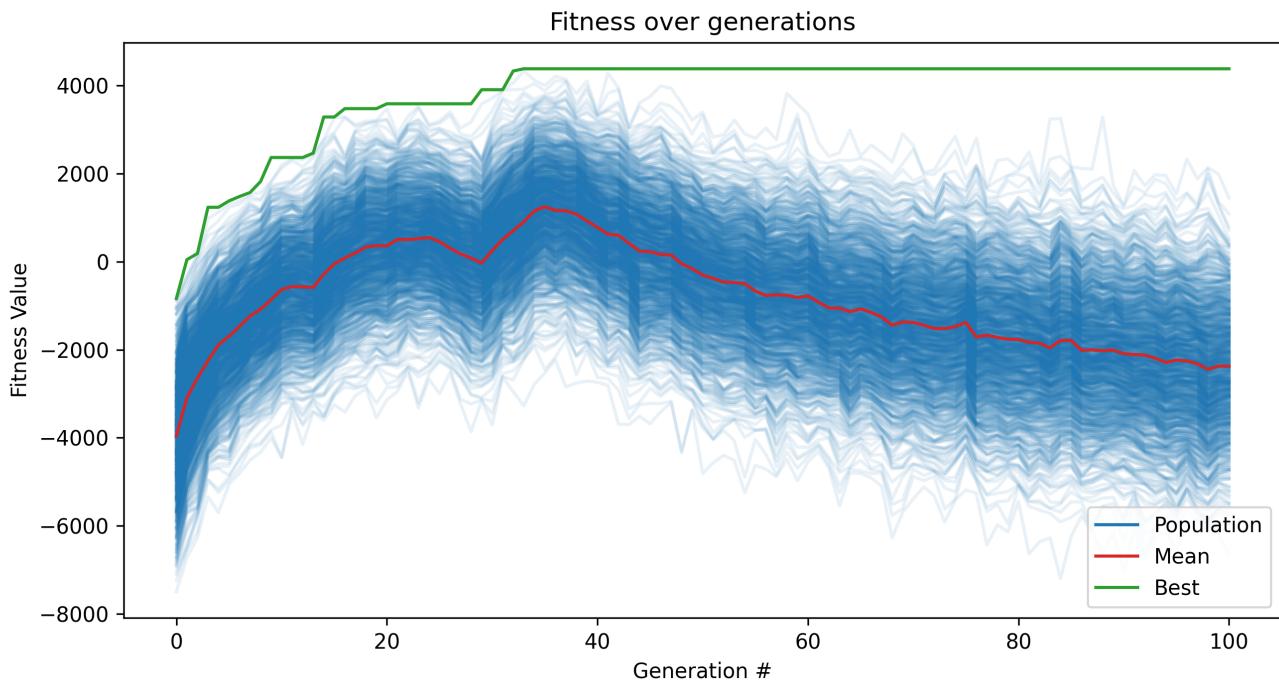


图 6: 自由随机突变策略下种群适合度变化曲线

由图 6 可见，种群最优在 40 代左右稳定，但是稳定时结果显然不如对照组。由于没有约束突变方向，故对于原种群中较好的特征无法保留，使得产生更好结果的可能大大下降。

顺序杂交 (Best Mate Worst)

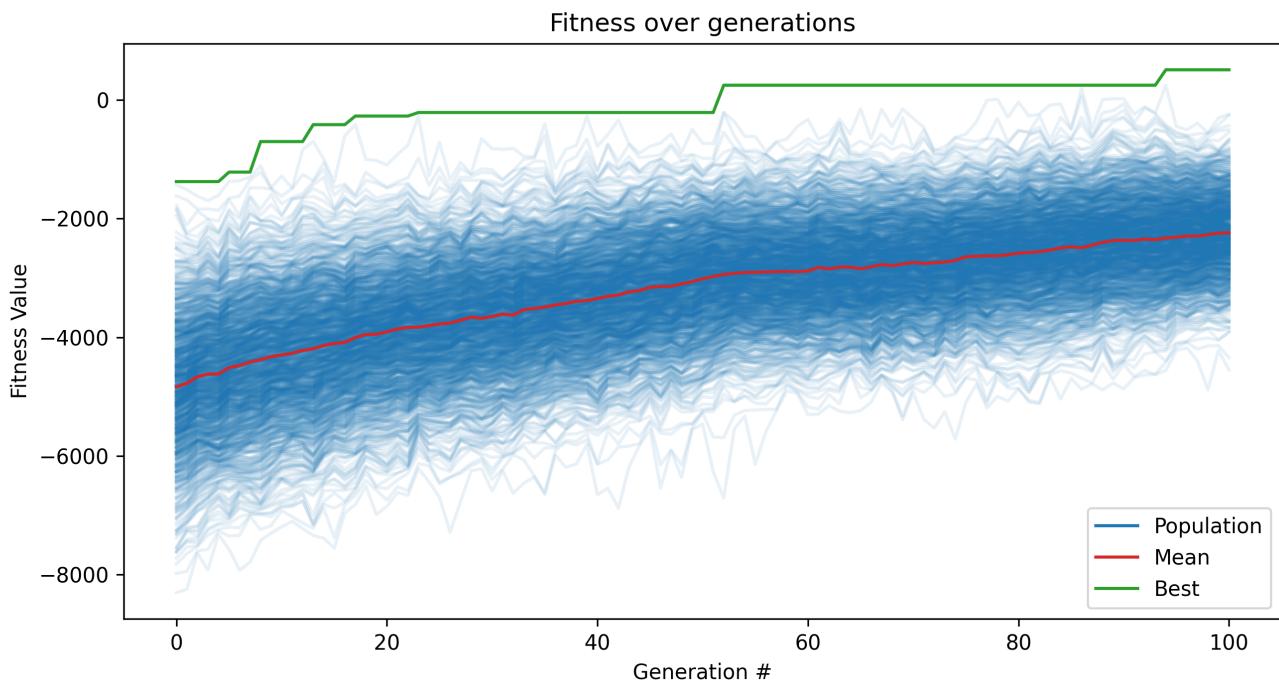


图 7: 顺序杂交策略下种群适合度变化曲线

为结果最差的一组，图 7 中种群没有在给定代数内没有收敛迹象。尽管这种方法在前人的工作[9]中取得了较好的效果，但是可能由于没有有效保留较好的特征，并且此处我们适合度函数涉及较多指标，因此无法较快收敛。

不保留最佳 (No Keep Best)

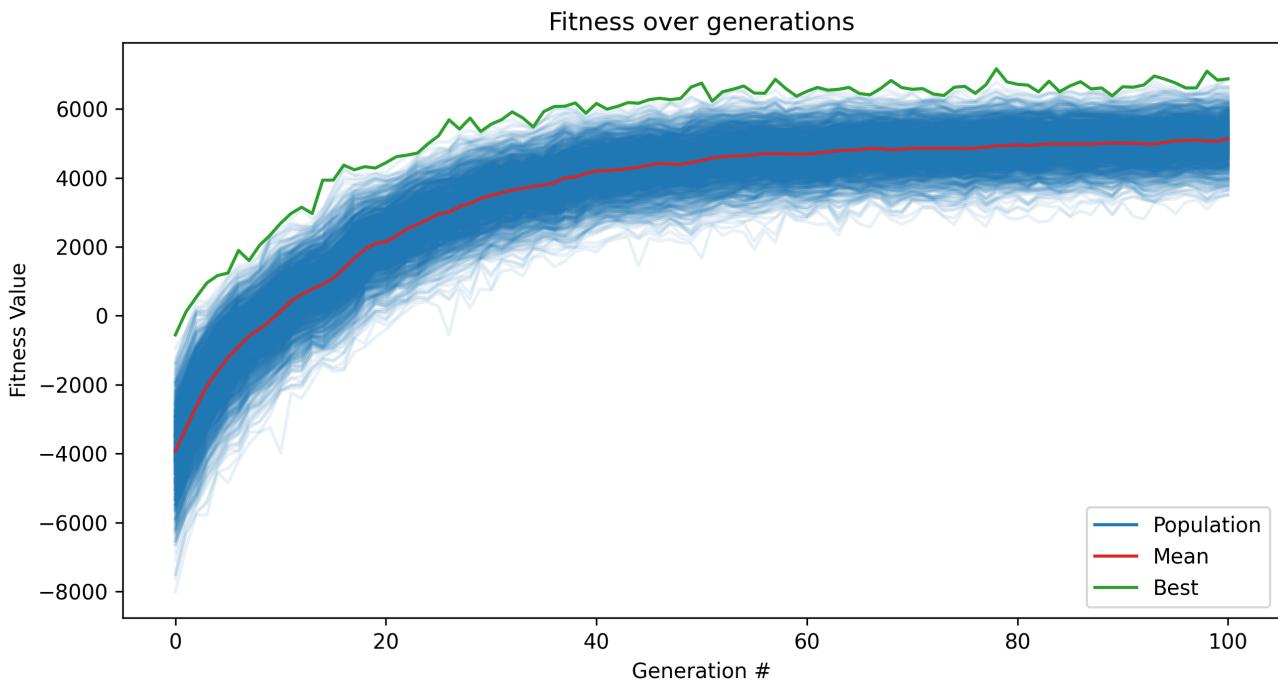


图 8: 不保留最佳策略下种群适合度变化曲线

由图 8 可知，在此策略下由于每一代的最优个体不稳定，因此实际上导致了自适应突变率策略的失效，因此不仅无法产出稳定的最优解，还容易由于过拟合导致实际结果不佳。

更少个体、更多代数 (1K Generations)

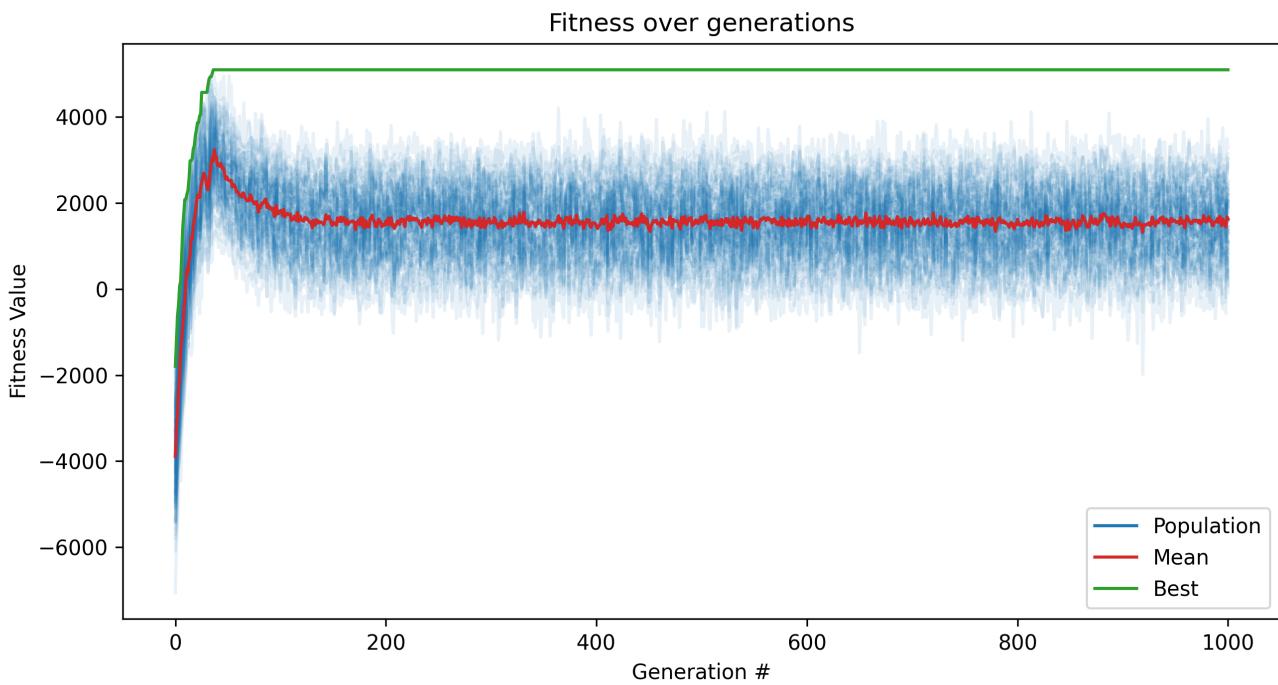


图 9: 更多代数策略下种群适合度变化曲线

本实验组实际上未对遗传策略进行任何调整，仅在保持总个体数不变的情况下调整了代数与每代个体数的比例。由图 9 可见，由于每代个体数较少，因此在稳定后进一步搜索参数空间时无法保留下较好的特征，因此效率不如对照组，整体评分较低。

横向比较印证标准策略的收敛速度与搜素能力

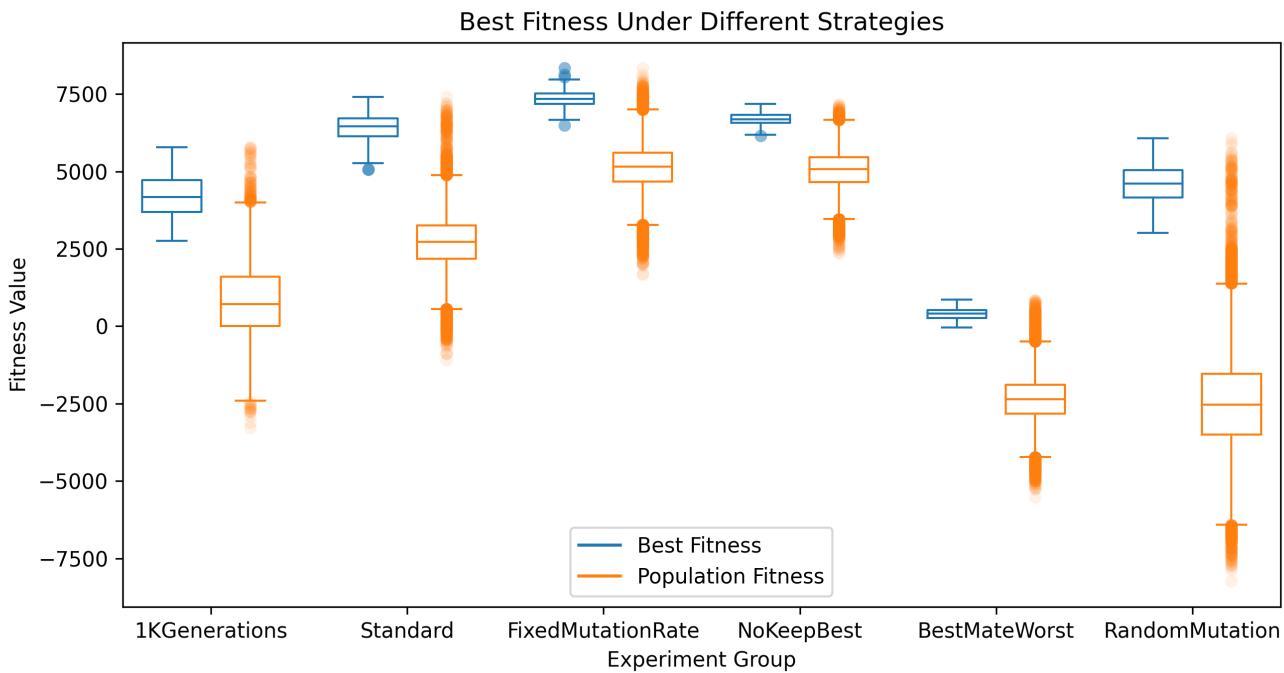


图 10: 不同遗传策略下稳定时种群的横向比较

为了进一步证明标准策略能够结合几种方法的长处，我们使每种策略组合分别运行 100 次，并对其结束时的种群适合度进行统计绘图，结果见图 10。由图中可以看出，标准策略组合（蓝色橙色，2）下，既可以收敛到和固定突变率（蓝色，3）近似的较好结果，同时保证群体足够发散，离群点较多，有类似于随机突变（橙色，6）的参数空间游走能力，反过来进一步帮助搜索更优解，有效防止了过拟合（蓝色，3-4）的发生。

附录

各类遗传策略下运行结果

结果参见[此目录](#)，各种策略组合的运行结果（音乐序列文本、midi 文件、适合度变化图）均置于对应名称的文件夹下。

原始代码

[C++ 代码](#)

参考文献

1. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1987). Optimization by simulated annealing. In Readings in computer vision (Elsevier), pp. 606–615. Available at: <https://doi.org/10.1016/b978-0-08-051581-6.50059-3>.
2. Paris, G., Robilliard, D., and Fonlupt, C. (2004). Exploring overfitting in genetic programming. In Lecture notes in computer science (Springer Berlin Heidelberg), pp. 267–277. Available at: https://doi.org/10.1007/978-3-540-24621-3_22.
3. Foster, P.L. (2000). Adaptive mutation: Implications for evolution. BioEssays 22, 1067–1074. Available at: [https://doi.org/10.1002/1521-1878\(200012\)22:12<1067::aid-bies4>3.0.co;2-q](https://doi.org/10.1002/1521-1878(200012)22:12<1067::aid-bies4>3.0.co;2-q).
4. PALERO, F., and CRANDALL, K. (2009). Phylogenetic inference using molecular data. In Decapod crustacean phylogenetics (CRC Press), pp. 67–88. Available at: <https://doi.org/10.1201/9781420092592-c5>.
5. Baldwin, J.M. (1896). A new factor in evolution. The American Naturalist 30, 441–451. Available at: <https://doi.org/10.1086/276408>.

6. Baldwin, J.M. (1897). Organic selection. *Science* 5, 634–636. Available at: <https://doi.org/10.1126/science.5.121.634>.
7. Bickle, T., and Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation* 4, 361–394. Available at: <https://doi.org/10.1162/evco.1996.4.4.361>.
8. Schlichting, C.D. (2007). Themes on variation. *BioScience* 57, 372–373. Available at: <https://doi.org/10.1641/b570411>.
9. Haghrah, A., Nazari-Heris, M., and Mohammadi-ivatloo, B. (2016). Solving combined heat and power economic dispatch problem using real coded genetic algorithm with improved mühlenbein mutation. *Applied Thermal Engineering* 99, 465–475. Available at: <https://doi.org/10.1016/j.applthermaleng.2015.12.136>.