

启发式搜索与演化算法：作业二

魏沐昊 502023370047

November 26, 2023

1 HW1

主要任务：实现演化算法求解子模优化问题-Maximum Cut

在本节中，将详细描述相关算子的设计以及实现，有关子模优化问题参看任务描述，此处不再赘述。并且，此处也不再提供代码实例，具体实现请参看相应代码文件。

- **generate_population**: 该函数生成一个种群，生成方式为从一个 $(-1, 1)$ 的均匀分布中随机采样。
- **tmp_population2population**: 该函数将实数值表示的个体转换为01串表示的个体。
- **score**: 该函数复用get_fitness函数为转换后得到的01串打分。
- **selection**: 该函数通过随机选取种群中的个体，并再次选择这些个体中score较大的作为选择后的个体。
- **generate_parents**: 该函数通过调用selection来生成父代个体。
- **mutation**: 此处即为 $(1+1)$ EA mutation。
- **crossover**: 该函数通过在父代个体中随机选择索引值将索引前后的序列交叉拼接。
- **children**: 该函数描述生成子代的过程，即先产生父代，接着父代通过先交叉再变异的过程产生初始子代，最后通过在种群和子代的集合中筛选其中score更大的个体生成后继者。
- **get_best_fitness**: 该函数返回种群中的最优解的fitness。
- **evolutionary_algorithm**: 该函数描述了一个相对完整的演化过程，返回子代以及迭代次数。
- **distance**: 该函数利用Hamming距离来描述不同个体之间的差异。

2 HW2

主要任务：演化算法改进：实现并比较特定的演化算法(standard bit-wise mutation vs heavy-tailed mutation)

- **heavytail_mutation**:继承上述**mutation**，其中根据 ζ 函数和normalization constant的不等式关系，设计一组 β 系数验证(从1.5起，步长为0.5)，取其中最小的 β 为分布的 β 值，可以发现 $\beta = 1.5$ 。变异策略请参看具体算法介绍以及(1+1)EA mutation。
- **heavytail_children**:继承上述**children**。
- **heavytail_evolutionary_algorithm**:继承上述**evolutionary_algorithm**。

算法结果如下图所示：

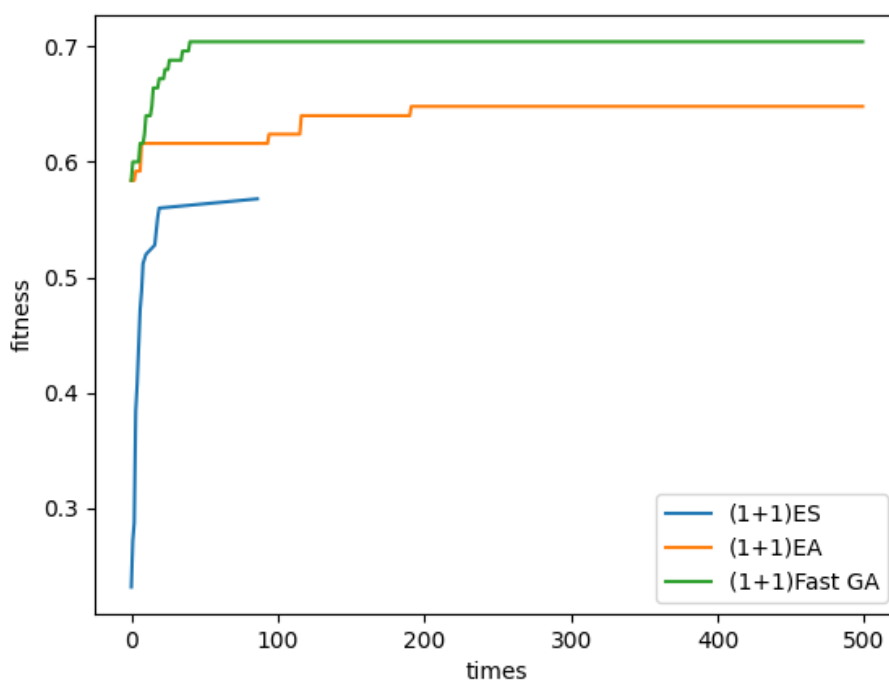


Figure 1: result

其效果显然得见：(1+1)fast GA显然优于(1+1) EA。

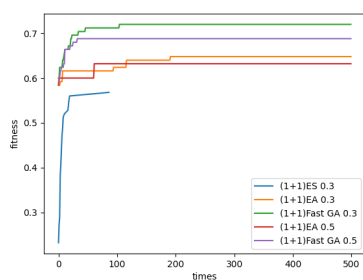
复现参数：保持原始参数不变，同时设置种群的个数为500。

3 HW3

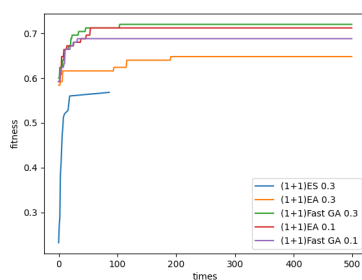
主要任务：演化算法改进：如何设计更适合Maximum Cut问题的演化算法？

此部分尝试了一系列参数来探究更好的演化算法。

3.1 调整mutation_rate



(a) mutation_ratio=0.5



(b) mutation_ratio=0.1

Figure 2: 调整mutation_rate

3.2 设置两个交叉点

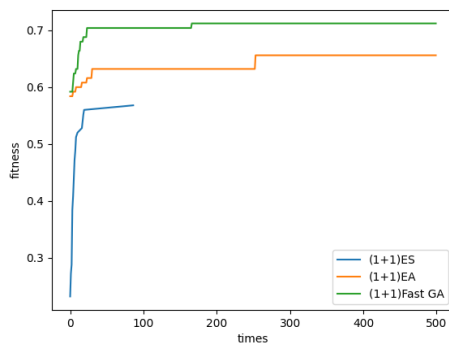
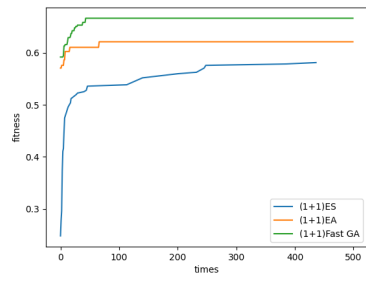
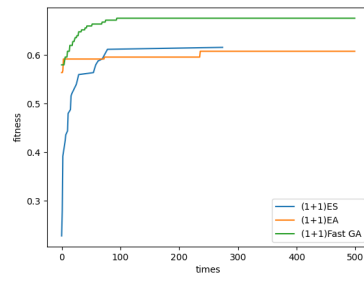


Figure 3: crossover_point=2

3.3 验证算法在大规模问题上的能力



(a) $\text{args.n_d}=15$



(b) $\text{n_node}=100$