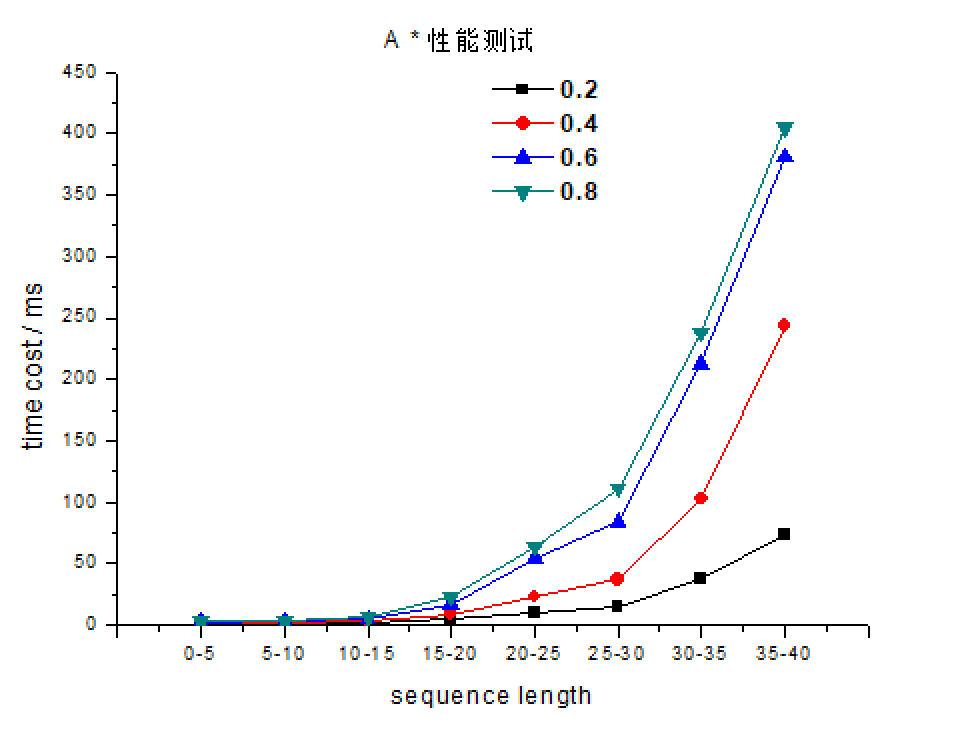
1. **实验目的**

使用真实模型数据测试A\*修复、基于结构修复和最小代价预估修复三种算法性能数据

1. **实验数据**

实验数据共99个模型，其中branching net 63个，causal net 26个，附带循环结构的模型10个，使用工具为其生成正确的模型序列，在使用时根据实验要求，对序列数据进行一定比例的打乱处理。

1. **A\* 修复算法性能测试**



**说明**：

四条曲线分别为序列乱序比例为：0.2、0.4、0.6和0.8比例的测试结果数据

X轴：序列长度范围

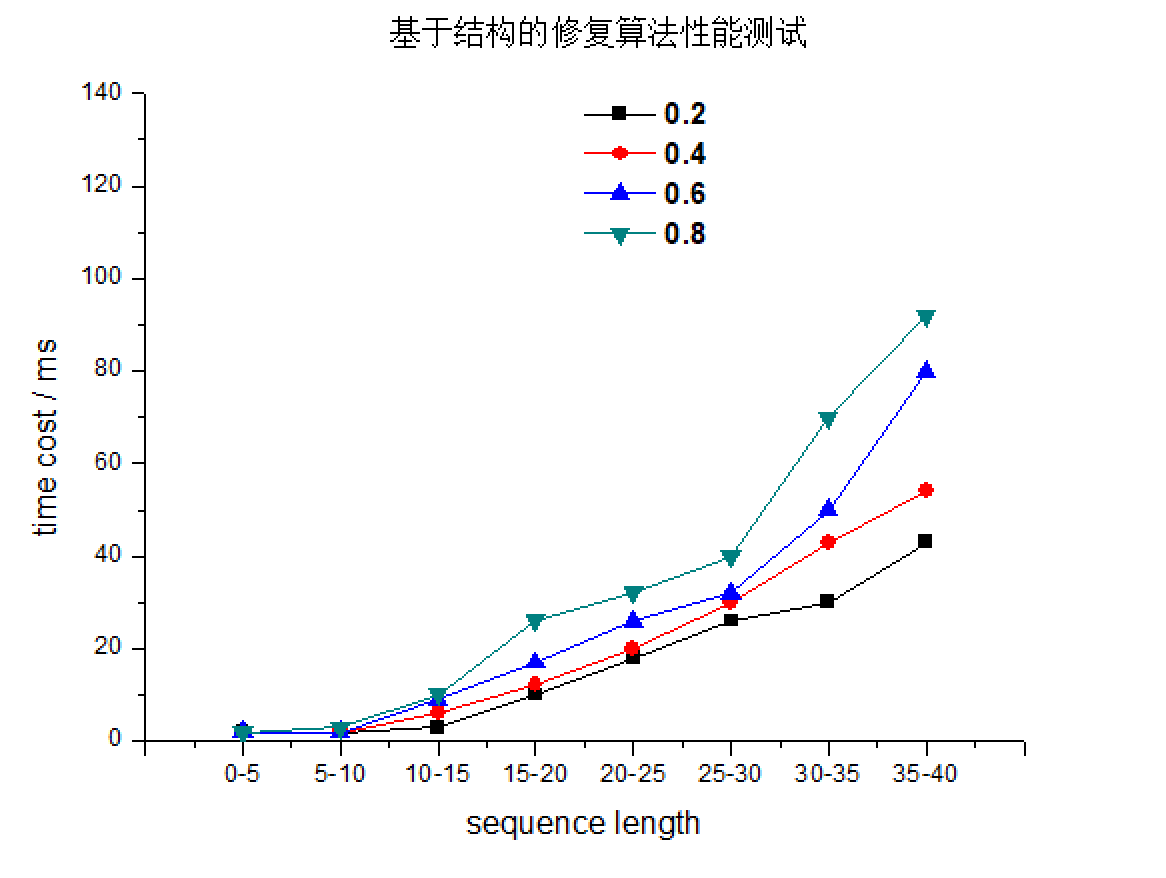
Y轴：修复时间

1. **基于模型结构的修复算法性能测试**

该方法在修复过程中，从前向后遍历整个序列，在遍历序列的过程中，同时解析模型，判断当前元素位于何种结构中，将按照如下方式进行处理：

|  |  |  |
| --- | --- | --- |
| 序号 | 结构 | 处理方式 |
| 1 | 顺序结构 | 解析模型，给出修复结构 |
| 2 | 分支结构 | 解析模型，判断所属分支，给出修复结构 |
| 3 | 并发结构 | 使用A\* 算法修复 |
| 4 | 循环结构 | 循环展开，递归操作。 |

给出的测试结果如下：



**说明**：

四条曲线分别为序列乱序比例为：0.2、0.4、0.6和0.8比例的测试结果数据

X轴：序列长度范围

Y轴：修复时间

**结论：**

从性能曲线可以看出，该方法的修复时间和序列长度之间更接近线性关系，可以进一步做拓展实验，人工增常测试序列，展示该算法在较长序列长的修复性能。

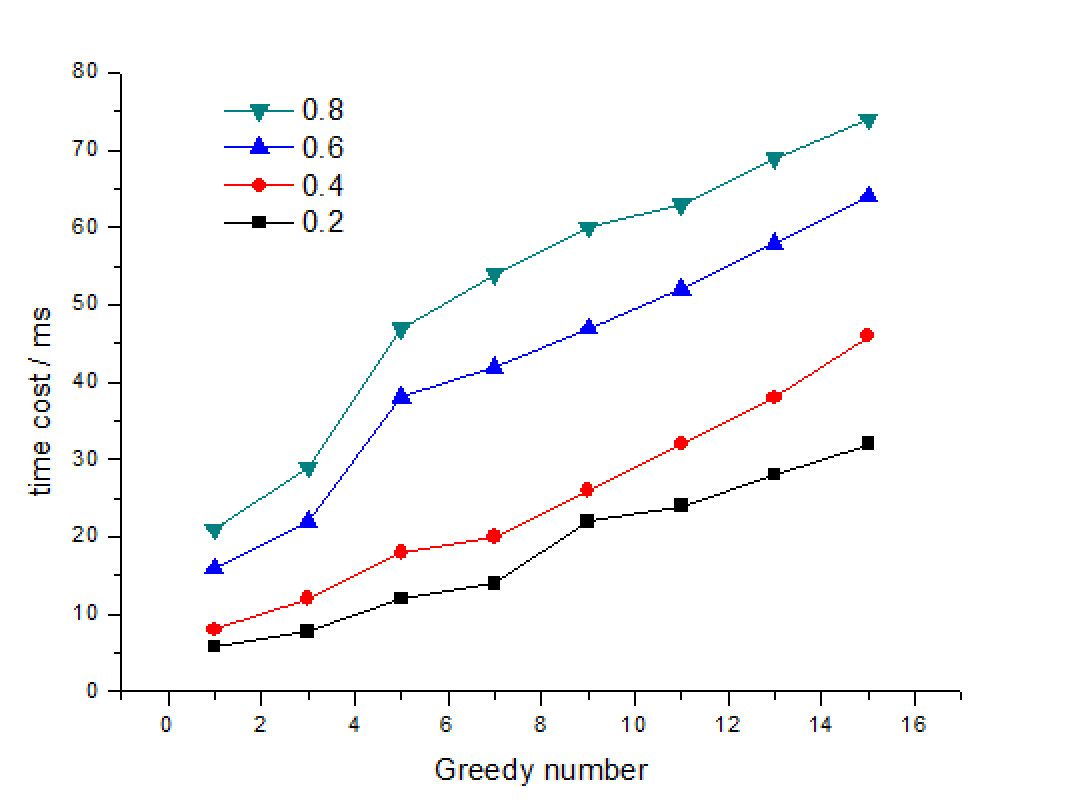
1. **最小预估代价修复算法性能测试**

最小预估代价修复算法本质属于A\* 修复算法，通过牺牲准确率的方式，提高算法性能，在长序列的修复中获得更好的性能。当算法贪心个数为1时，修复算法准确率最低，获得最好性能，当贪心元素个数等于序列长度时，最小预估代价修复算法退化为A\* 算法，当退化为A\*算法时，性能与A\* 算法相同。

**实验**：准确率和性能关系测试

**实验数据**：本次实验为每个模型生成10个序列数据，共99个模型，每个模型的序列按照一定比例进行打乱处理，算法修复时间取所有序列修复的平均值。

1. 性能: (贪心个数-性能)

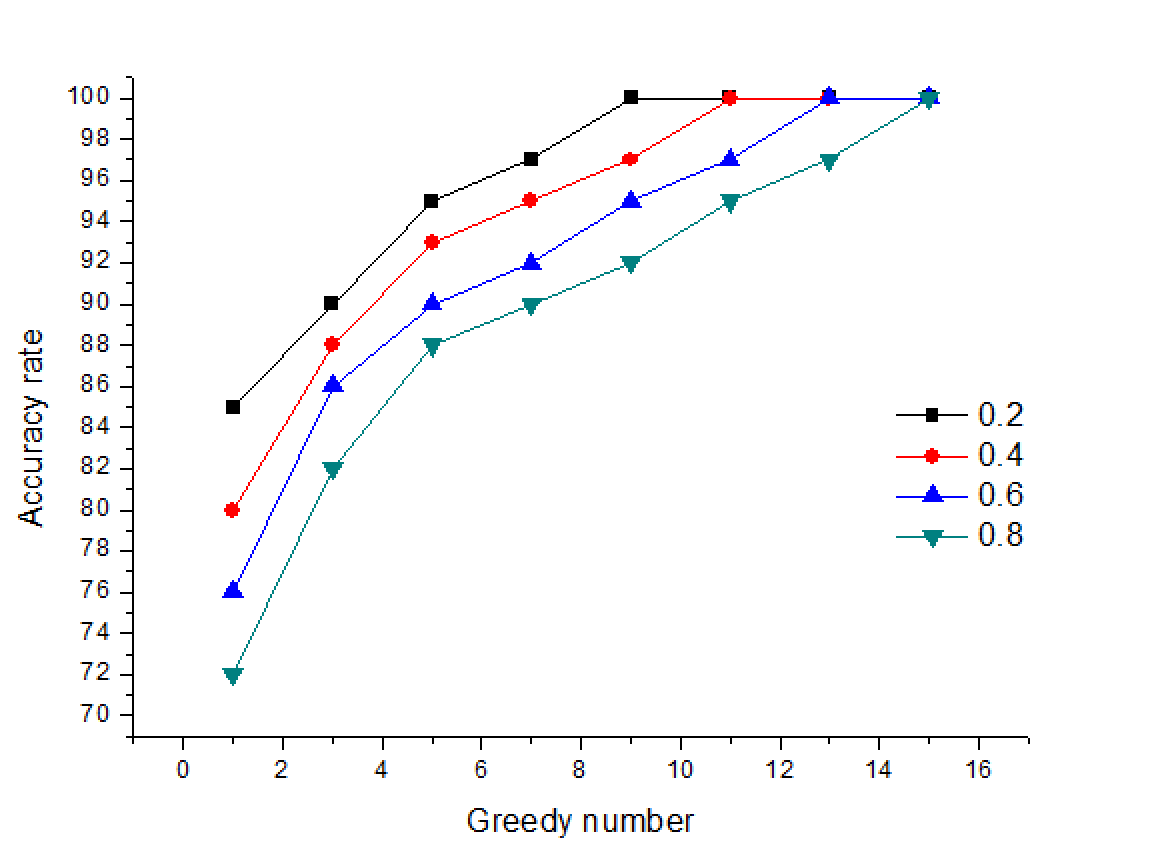


**说明：**

**X轴：算法贪心元素个数**

**Y轴：数据集样本上平均修复时间**

1. 准确率：（贪心个数-准确率）



**说明：**

**X轴：算法贪心元素个数**

**Y轴：数据集样本上平均修复准确率**