程序食用指南

学习方向就是我的代码迭代方向：

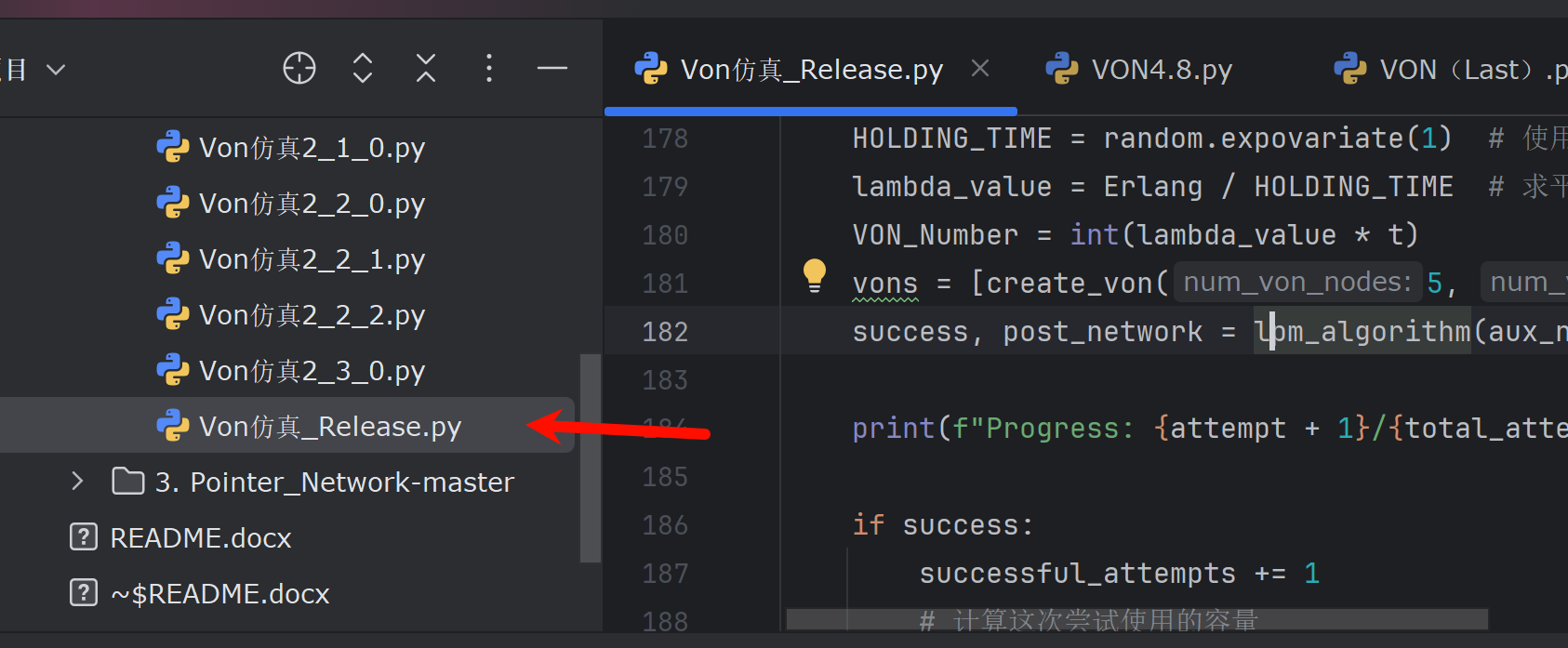
接下来的所有结果都是在此变量下进行



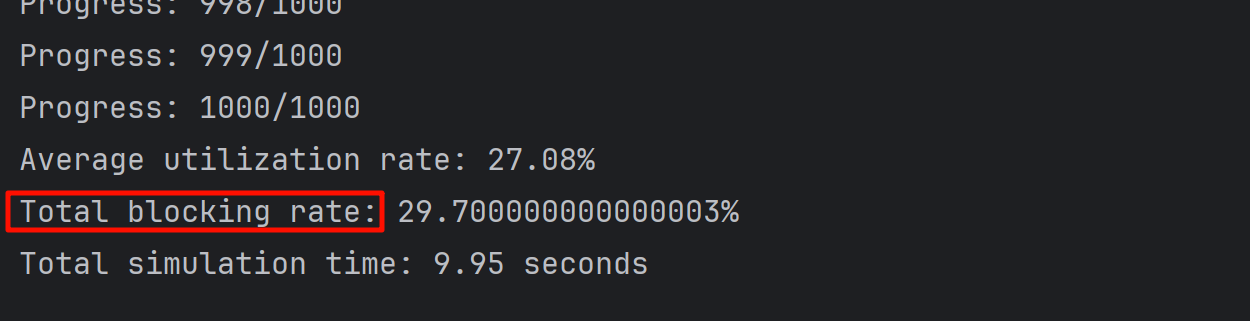
作图只需要在下面几个文件里更改Erlang值即可，记录出来的阻塞率与资源利用率

1. 最初的不带任何策略的LPM（NPM同理后续只拿LPM做介绍）

对应文件如下

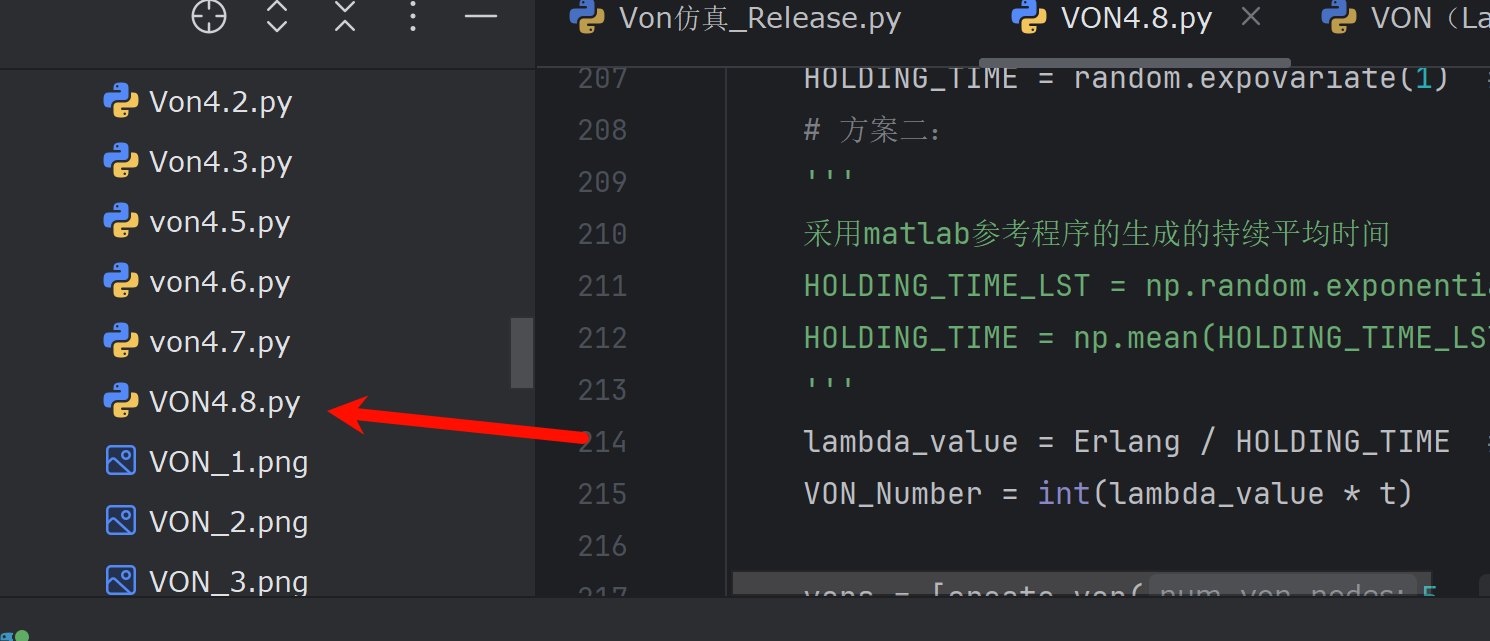


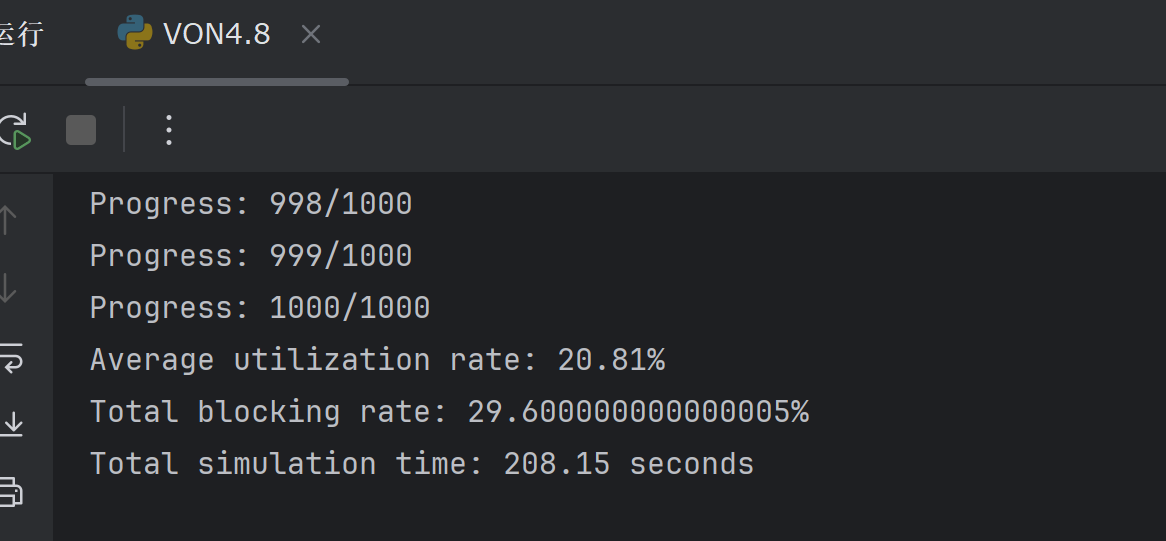
跑完后的阻塞率与资源利用率如图

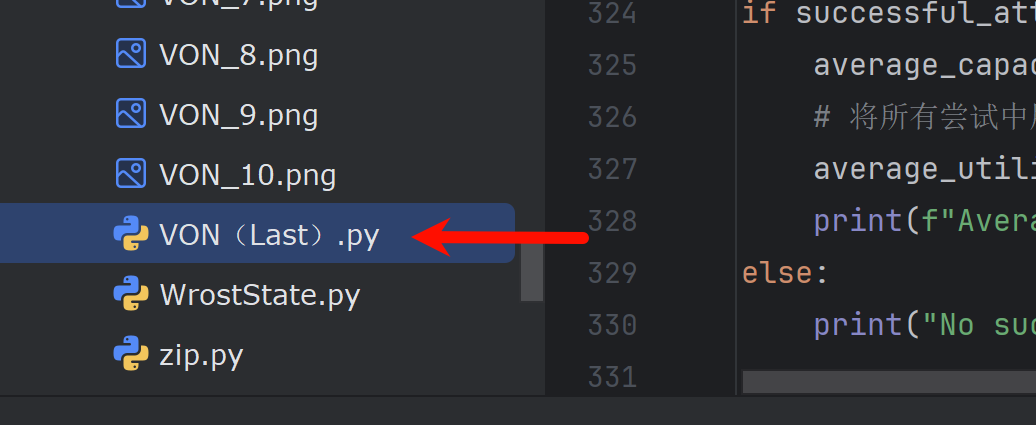


2．带动态释放策略的LPM

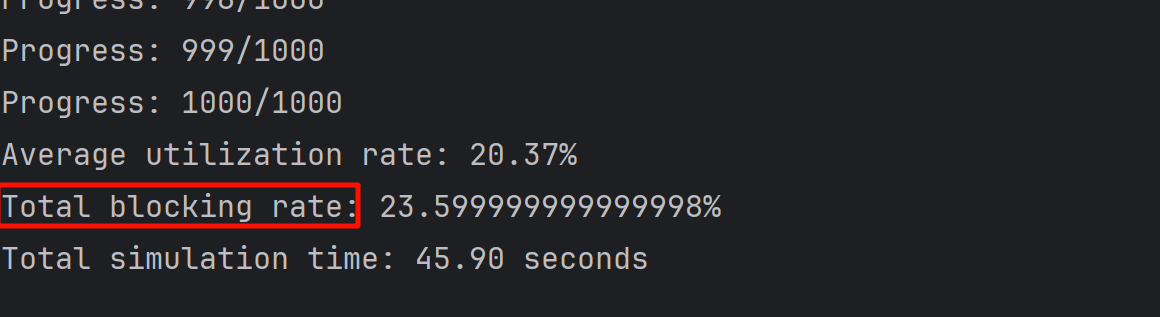
程序文件为4.8



结果（主要是让资源利用更优了）

3．最终版，加入调至格式，第二次命等各种策略

结果



另外还有两个Best和Wrost State 各代表该模型下的最好最坏情况（此时最好因为是基于最初版Release版本构成，没用硬来的节省资源来的猛，最后的版本阻塞率已低于Best，可在后续说明）

暂时没有方案继续提升，此时已经是节约带宽的瓶颈了，再于放置于链路上的策略不是很有效。假设节点的计算资源无限大，阻塞率才有很大的下降空间。说明是节点资源过少，当链路负载过多，节点映射失败增多。

4．固采用DQN的深度学习模型来让机器学习映射这一过程，让生成的模型里有过的网络，能够“记住”之前不堵的节点放置，从而降低阻塞率。（详细见另一文件）