

六子棋实验报告

**2023 年 秋 季学期**

姓名：周忠康 学号：21060021080

姓名：王颢凯 学号：21040021027

姓名：曹文汀 学号：21020041002

姓名：田晶怡 学号：21050005076

摘 要

针对要求的三种走法，我们首先对他们进行了预估，走法3胜率比走法2高，随后分别对其进行了编程实现，random02对应走法2，random03对应走法3。然后使用走法2和走法3分别作为黑棋先手白棋后手下500场，从实验结果可知，走法3不论是先手还是后手，都比走法2胜率高，由此验证了我们开始时预估的正确性。接着我们对走法3种两个变量中心区域长度和次数限制对其胜率的影响进行了探究，结果见图5-1和图5-2。

目 录

[**摘 要** 2](#_Toc152872159)

[**1. 走法要求** 1](#_Toc152872160)

[**1.1 三种走法** 1](#_Toc152872161)

[**1.2 要求** 1](#_Toc152872162)

[**2. 走法预估** 1](#_Toc152872163)

[**3. 代码实现** 1](#_Toc152872164)

[**3.1 准备工作** 1](#_Toc152872165)

[**3.2 Package random02** 2](#_Toc152872166)

[**3.3 Package random03** 3](#_Toc152872167)

[**4. 实验结果** 4](#_Toc152872168)

[**5. 思考与探究** 4](#_Toc152872169)

[**5.1 中心区域长度对走法3胜率的影响** 4](#_Toc152872170)

[**5.2 次数限制对走法3胜率的影响** 5](#_Toc152872171)

1. **走法要求**
   1. **三种走法**

走法1：两个子的位置均通过随机掷骰子的方式确定，在整个棋盘（19\*19）范围内掷骰子。

走法2：通过随机掷骰子确定第一个子的位置，第二个子下在与第一个子相邻的空位上，和第一个子相邻的空位最多有8个，要求随机选择一个空位落子，若相邻的8个位置都有子，则在整个棋盘上随机选择一个空位落子。

走法3：两个子的位置均通过随机掷骰子的方式确定， 在棋盘的中心区域（13\*13）随机掷骰子，若连续10次不中（没有找到空位），则在整个棋盘（19\*19）范围内掷骰子。

* 1. **要求**
* 预估走法2和走法3哪种走法的胜率更高。
* 编程实现走法2和走法3各自先手各比500场，并得出结论，以检验开始的预估是否正确。

1. **走法预估**

通过分析走法2和走法3，我们预估走法3比走法2胜率更高。具体考虑如下：

虽然走法2两次落子是相邻的，但是其随机掷骰子的范围是整个棋盘，即在整个棋盘区域内随机落子，导致整体上的落子比较分散。而走法3是在中心区域内随机掷骰子，虽然两次落子不一定相邻，但其能够更快地填满棋盘的中心区域，使落子更加密集，更容易连成六子，因此胜率比走法2高。

1. **代码实现**
   1. **准备工作**

在提供的代码模板中，Package g02是走法1的实现。我们模仿其编写了Package random02和Package random03，分别对走法2和走法3进行了实现。

考虑到走法2和走法3中都会通过随机掷骰子确定棋子的位置，因此我们在抽象类AI中添加了getRandomPosition方法，通过设置起始和终止的范围，在棋盘上得到该范围内一个随机点的坐标。

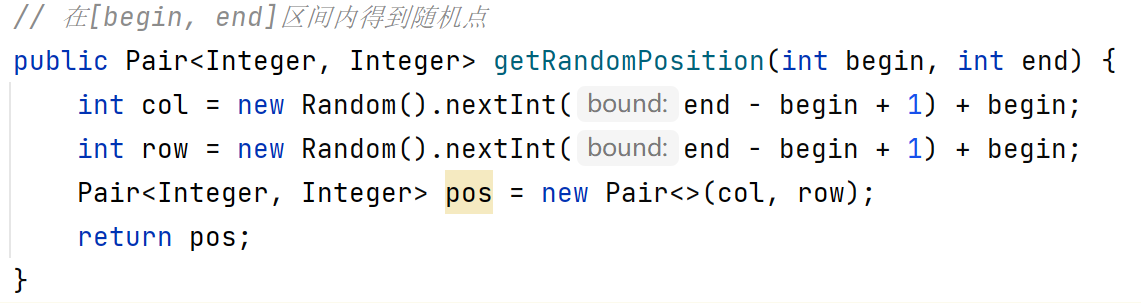


图3-1 getRandomPosition方法

* 1. **Package random02**

我们在Package random02中编写了Class AI对走法2进行了实现。走法2的第二个子下在与第一个子相邻的空位上，我们通过实现getAdjacentPosition方法得到棋盘上一个点的相邻位置，若相邻位置都有子，则返回空，反之，则随机选取一个空位，返回其坐标。

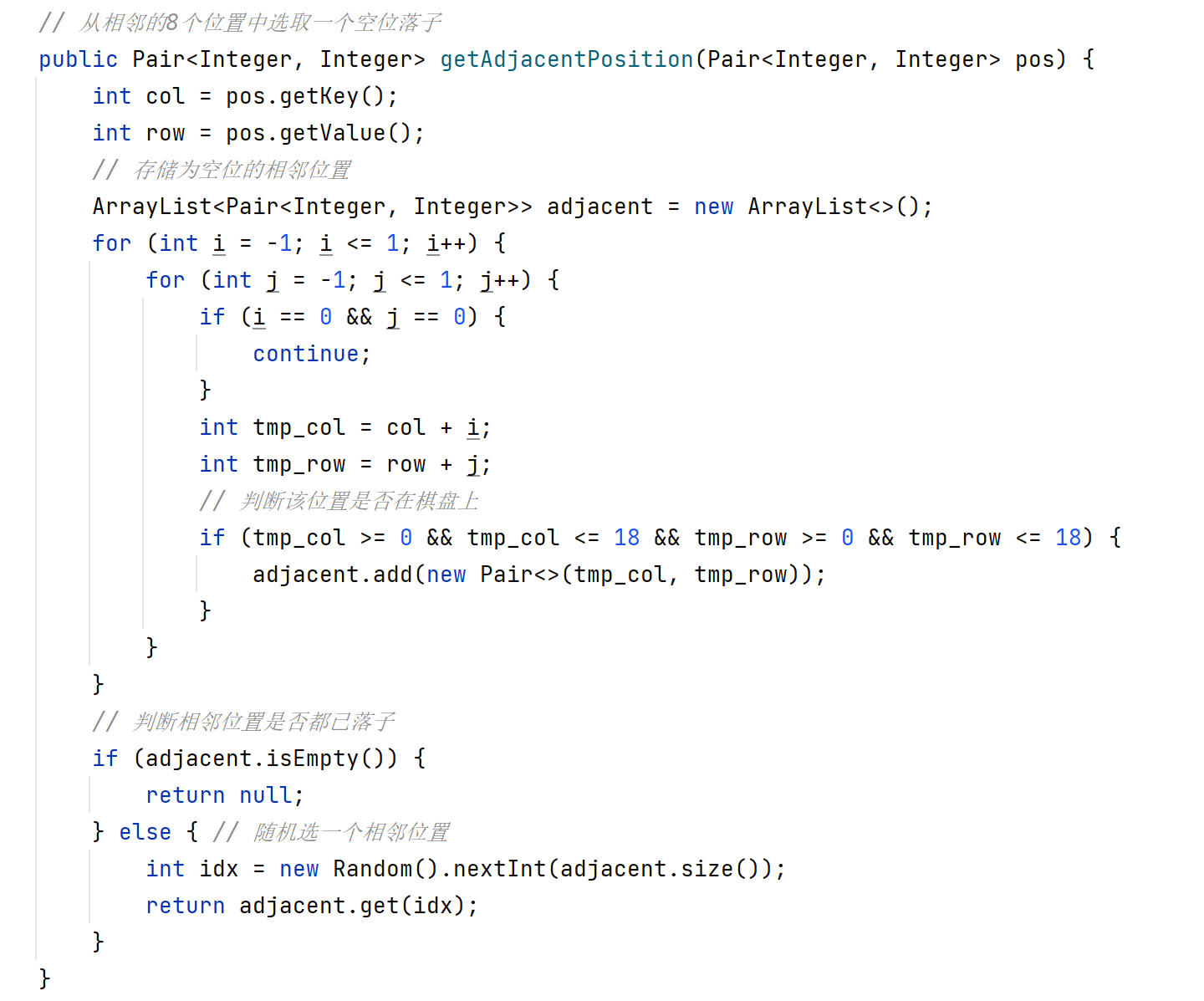


图3-2 getAdjacentPosition方法

通过重写findMove方法，我们对走法2进行了实现，通过在整个棋盘范围内随机掷骰子确定一个子的位置，然后判断第一个子相邻位置是否都有子，若有则重新再全局范围内随机掷骰子选择第二个子，否则随机选择一个相邻空位落子。

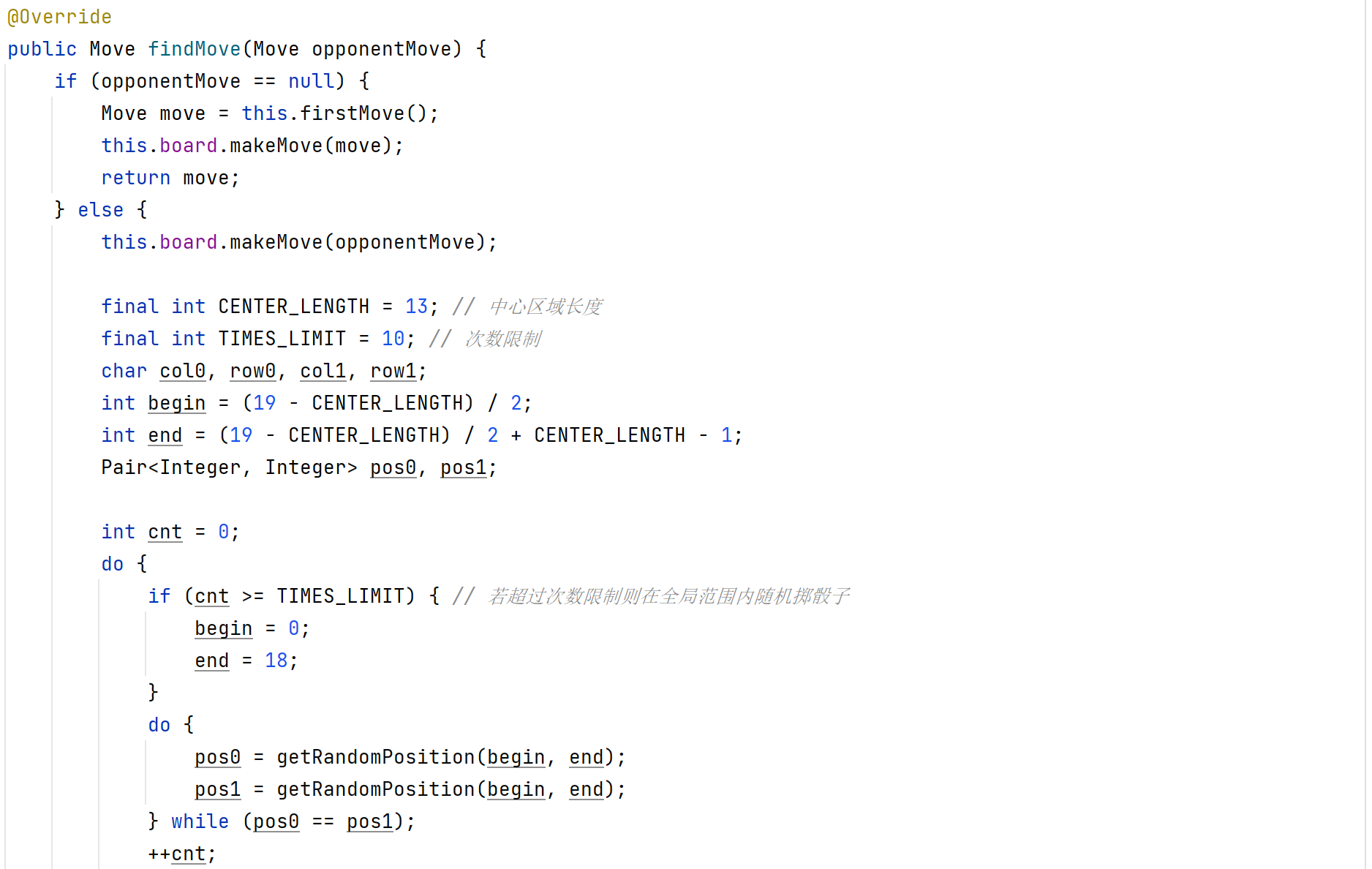




图3-3 random02中的findMove方法

* 1. **Package random03**

我们在Package random03中编写了Class AI对走法2进行了实现。我们通过重写findMove方法对走法3进行了实现，首先在棋盘的中心区域随机掷骰子，若连续10次都没有在中心区域找到空位，则在整个棋盘范围内掷骰子。



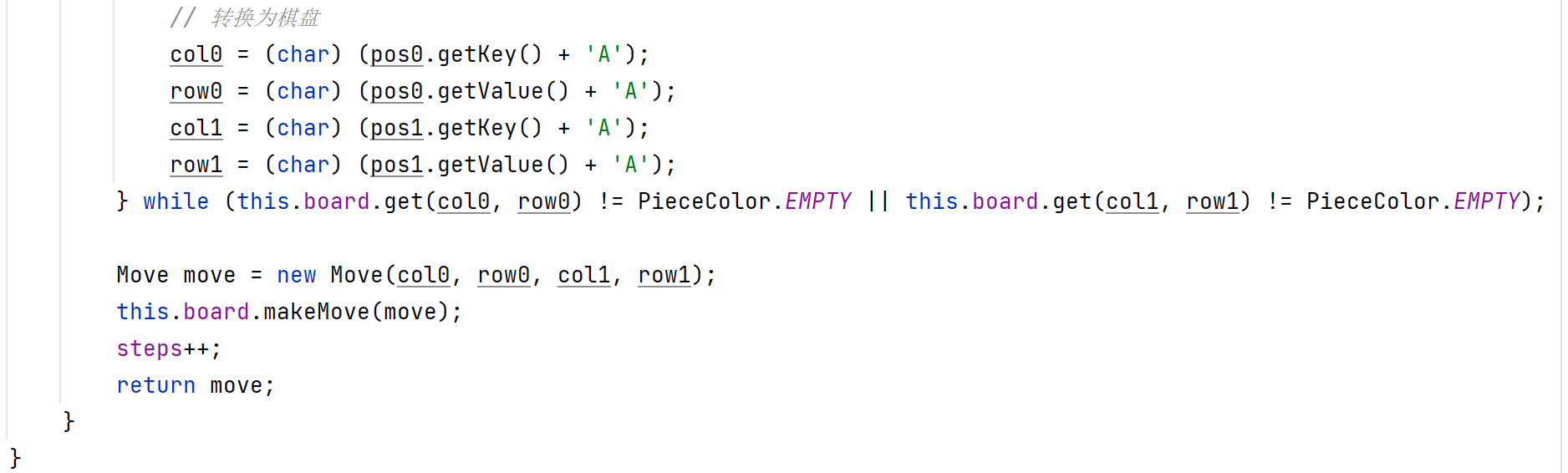
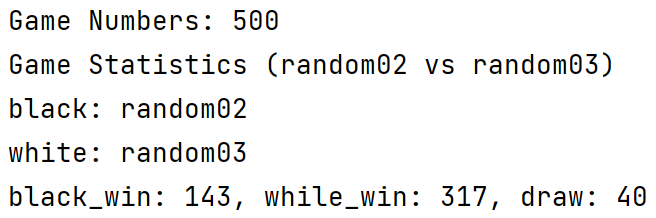


图3-4 random03中的findMove方法

1. **实验结果**

使用走法2和走法3分别作为黑棋先手白棋后手下500场，统计结果如下图所示。



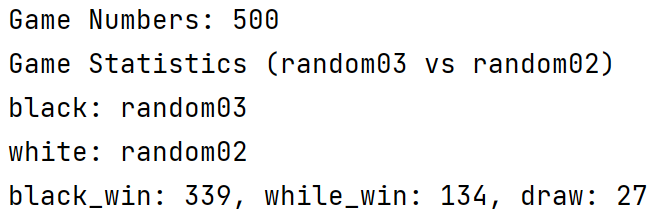


图4-1 实验结果

从实验结果可以看出，走法3不论是先手还是后手，都比走法2胜率高，由此可见，我们开始时的预估是正确的。

1. **思考与探究**

我们思考走法3中的两个参数中心区域长度center length和次数限制times limit对其胜率可能会造成一定的影响，现探究这两个参数对走法3胜率的影响。

* 1. **中心区域长度对走法3胜率的影响**

现固定次数限制times limit=10，以中心区域长度center length为自变量，从1变化至19，间隔为2，使用走法2和走法3各自先手各比500场，将两种走法各自先手的胜场数分别相加，得到图5-1，如下所示。

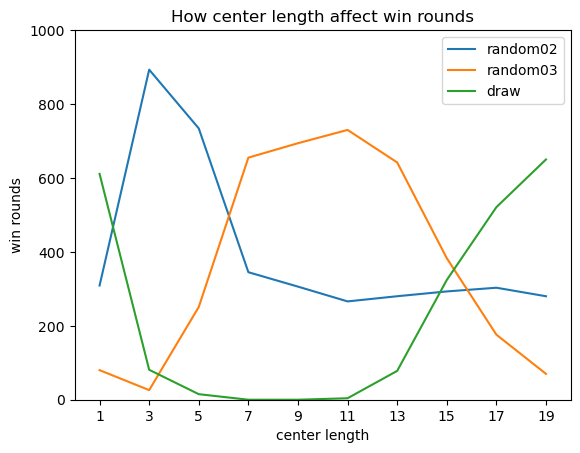


图5-1 探究中心区域长度对走法3胜率的影响

从图中可以看到，走法3的胜率曲线呈单峰型，在center length=13时达到最大，在此基础上增大或减小center length，都将减小走法3的胜率。当中心区域很小时，走法3退化成两步都是随机的走法；而当中心区域很大时，走法3同样退化成两步都是随机的走法，胜率均低于走法2。

* 1. **次数限制对走法3胜率的影响**

现固定中心区域长度center length=13，以次数限制times limit为自变量，从1变化至20，使用走法2和走法3各自先手各比500场，将两种走法各自先手的胜场数分别相加，得到图5-2，如下所示。

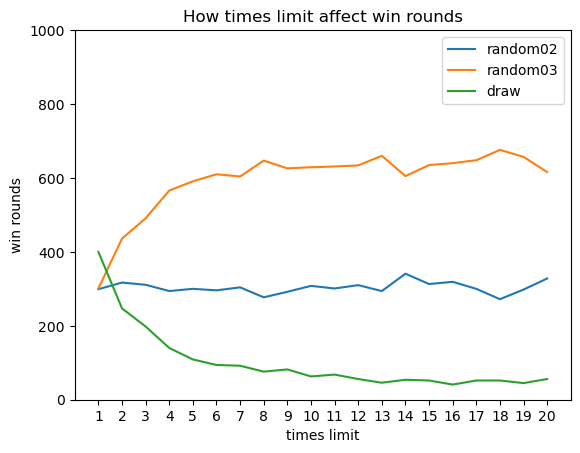


图5-2 探究次数限制对走法3胜率的影响

从图中可以看出，走法3的胜率先增长后趋于平稳。在times limit小于10时，走法3的胜场数会随times limit增大而增大，其原因是因为平局的次数随times limit的增大而减少。当times limit大于10之后，走法3的胜场数趋于平稳，平局的次数也趋于平稳。而不论times limit如何变化，走法2的胜场数都无明显变化，稳定在150场左右。

由此我们可以得出，影响走法3胜率的是开始的几步，如果开始几步不在中心区域，则走法3退化为走法1即在整个棋盘范围内随机取点，胜率较低。而到了后期，棋盘中心区域基本饱和，在中心区域内选取位置和在整体范围内选取位置将无明显区别。