

HW3 数学建模报告

PB19151769 马宇骁

摘要：以 2020 东京奥运会乒乓球项目底赛制作为分析背景，对乒乓球参赛选手的排兵布阵进行建模，为比赛的选手安排提供理论指导。

关键词：奥运会，乒乓球，对局分析，目标设定

1 背景

乒乓球项目的团体比赛中，需由教练员或队长根据双方运动员的具体情况，预先安排各自的出场阵容及顺序，然后运动员根据所安排的顺序依次出场，进行以场为单位的回合制比赛。因此，在这样的比赛中，赛前的排兵布阵尤为关键。

谈及乒乓球团体比赛中排阵的重要性时，有 91.3% 的教练员或运动员都认为在双方的实力相近时，排阵的方案十分关键，甚至是起到决定胜负的作用。^[1]

1.1 赛制背景

赛制以 2020 东京奥运会作为背景参考：

男单、女单比赛最多各 70 位选手参赛。排名前 16 的种子选手前两轮轮空，排名 16 至 32 位的选手首轮轮空。从第三轮开始，不再设轮空，比赛采用淘汰制。混双比赛和团体赛均为 16 支球队参赛，比赛不设轮空，从第一轮开始极为淘汰制赛制。^[2]

奥运会乒乓球有男子团体、女子团体、男子单打、女子单打、混合双打这五个比赛项目。每个国家最多可以报名三名男选手和三名女选手，其中各有两名男/女选手参加单打，各有至多一名男/女选手只参加团体不参加单打，至多一对混双选手，允许兼项。团体比赛采取五场三胜制，第 2,3,4,5 场为单打，第 1 场为双打，每名运动员出场两次。赛前两队抽签，确定主、客队。然后根据抽签的结果和对方可能出场的队员情况，开始分别确定出场阵容。

主队的 3 名运动员分别用 A、B、C 表示，客队的 3 名运动员分别用 X、Y、Z 表示。第 1 场为双打 (BC — YZ)，第 2 场为单打 (A — X)，第 3 场为单打 (C — Z)，第 4 场为单打 (A — Y)，第 5 场为单打 (B — X)。该赛制要求各队在赛前通过抽签的方式确定主客队，随后根据选中的主队或是客队的出场顺序，确定 A、B、C 或 X、Y、Z 分别是哪一名运动员，然后按照规定的场次顺序进行比赛。

1.2 中国背景

乒乓球项目是我国竞技体育中的优势项目，也是奥运会上的重点夺金项目。从 2008 年北京奥运会上乒乓球项目加入了团体比赛起，截至目前，中国队包揽了所有 3 届的冠军。

因此，此次建模是要最大可能让中国队获得金牌为目的进行分析。

2 分析

2.1 混合双打

2020 东京奥运会开始第一次将混合双打作为奥运会正式项目纳入奥运章程，而且混双在很长周期内需要大量练习配合参加各项赛事取得足够积分。但限于奥运会国家队总共只能带男女队员各 3 人（不考虑递补名额），因此中国体育总局的安排是由男女队内 3 号选手组队参加混双（例如 2020 东京奥运会的许昕和刘诗雯组合）。

由于混双的比赛时间相比单打更早于团体赛，且他们不用参加单打，故混双不考虑参与较多的建模分析，且他们的体力损耗在考虑计算时相应保留更多（见 3.2.3 的分析）。

2.2 单打

鉴于男子单打金牌是中国队必争的奖牌，除过 2004 年男单，在近几十年来该奖牌被中国队包揽。大满贯（世锦赛，世界杯和奥运会）是每一位高水平的乒乓球运动员所追求的最高荣誉（例如 2016 里约奥运会最快大满贯荣誉的张继科作为队内 2 号赛前带着腰伤打封闭的情况下国家队也并未使用递补名额，最终与马龙会师决赛惜败马龙未能取得男单历史上第一位双圈大满贯的荣誉，马龙则获得男单历史上第一个全满贯荣誉）。

因此，考虑到本身的实力和运动员的荣誉，中国体育总局派出参加男子单打比赛的一直是男子单打队内 1 号与 2 号队员。他们分列上下半区，从八分之一决赛开打。

由于单打比赛决赛距离团体赛八分之一决赛时间间隔很短（2020 东京奥运会的间隔为 2 天），因此此时需要考虑这 4 位选手的体力损耗。

2.3 团体赛

根据 2018 年国际乒联团体世界杯比赛期间对参赛的各国教练员或运动员进行了访谈，共访谈 46 人（其中教练员 22 人，运动员 24 人），访谈内容主要围绕乒乓球团体比赛中排阵的重要性、排阵的关键因素、排阵对比赛结果的影响等方面展开。^[1] 统计并整理访谈结果后，将被提及频率超过 50% 的词条陈列如下 (表1):

分类	词条	被提及次数	提及频率/%
排阵的重要性	双方实力相近时阵容非常关键	42	91.3
	排阵时会有犹豫不决的情况出现	25	54.3
排阵的关键因素	了解对手的情况很重要	39	84.8
	教练员排阵的经验很重要	33	71.7
	教练员排阵时候需要有胆魄	27	58.7
	主客队选择很重要	26	56.5
排阵对比赛结果的影响	运动员的实力是胜负的关键	38	82.6
	有时换个出场顺序比赛结果可能不一样	31	67.4

表 1: 访谈结果——教练员与运动员对乒乓球团体比赛赛制的认识

在关于乒乓球团体比赛中排阵的关键因素的访谈中，经归纳后被提及频率超过 50% 的词条有 4 个，分别是“了解对手的情况很重要”“教练员排阵的经验很重要”“教练员排阵时需

要有胆魄”“主客队的选择很重要”。

由于主客队的选择是赛前抛硬币等随机手段决定，因此选择作为主队难以把握，故此时的排兵才是关键。

正如访谈结果所示，运动员的竞技实力是影响比赛结果的重要因素，若某个队的实力远远高于另一个队，那么比赛的结果肯定是明确的，跟排阵不可能会有太大的关系。但也有 67.4% 的教练员或运动员也提到，有时候若是改变一下出场顺序，比赛的结果可能就会发生改变。乒乓球团体比赛中的排阵对比赛结果是有影响的，但要排除双方实力差距太大的情况。实力稍高的队伍可能会因为没有排好阵容而输了比赛，实力稍弱的队伍也可能会因为排的阵容较好而取得胜利。

3 建模

3.1 模型假设

1. 不存在伤病等突发因素使得比赛中断；
2. 不考虑递补队员；
3. 对弈双方均正常发挥；
4. 双方均理性，即考虑自身胜率最大的排兵布阵。

3.2 模型建立

建立排阵的模型需要一些数学和计算机的工具：

3.2.1 能力值计算公式

使用国际乒联（ITTF）官方数据库（2021.8）排位中的积分数据^[3]作为能力值的原始数据（见图4, 5）。

由于男子和女子团体赛制相同，因此以下以男子团体比赛建模分析：

鉴于积分差距过大，且积分远大于 0，因此在使用特征函数之前先将原始数据进行归一化再进行能力值转化。取前 16 名的数据进行预处理。

$$g_i = \frac{p_i - \mu}{\hat{\sigma}} \quad (1)$$

其中 p_i 是原始积分数据。

此时考虑使用 Sigmoid 函数进行处理初步计算能力值：

$$S(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

将数据映射到 (0,1) 的区间。

考虑到积分差距头部仍然过大，再对 Sigmoid 处理后的数据做修正 \arctan ：

$$S'(x) = \frac{\arctan(x)}{\pi} + 0.5 \quad (3)$$

其中， $x \geq 0$ ，此时，绘图如下：

中国队的三位选手樊振东、马龙和许昕能力值结果如下：

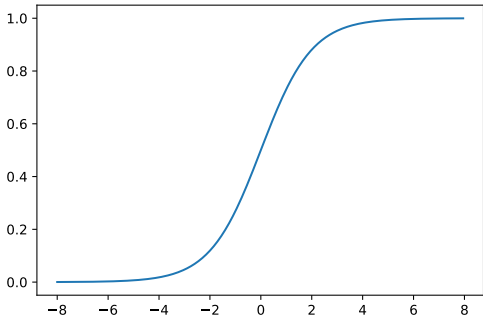


图 1: Sigmoid 函数

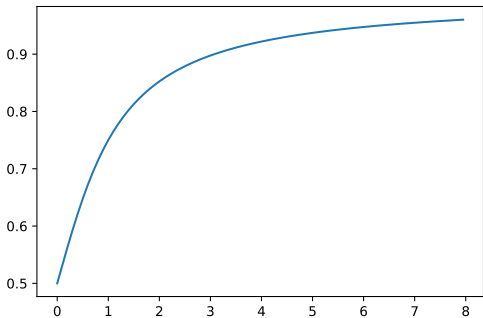


图 2: 修正 arctan 函数

- 0.7338804388095789
- 0.7315988916808487
- 0.7076299653921264

将积分前 16 名的能力值绘制如图3.

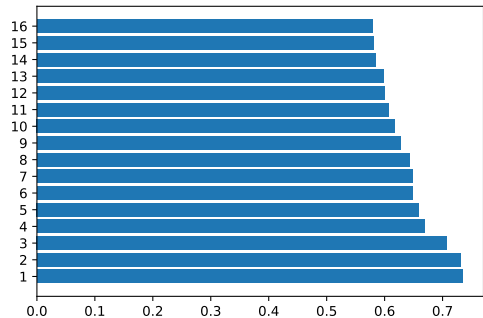


图 3: 能力值天梯图

看得出中国队的实力相比其他对手还是很高的。

TABLE TENNIS WORLD RANKING			
Senior Men's Singles			
2021 Week #35			
Rank	Name	Assoc	Points
1	FAN Zhendong	CHN	12494
2	MA Long	CHN	12212
3	XU Xin	CHN	10356
4	Tomokazu HARIMOTO	JPN	8709
5	LIN Yun-Ju	TPE	8375
6	LIN Gaoyuan	CHN	8034
7	Dimitrij OVTCHAROV	GER	7999
8	Hugo CALDERANO	BRA	7864
9	Mattias FALCK	SWE	7357
10	Timo BOLL	GER	7086
11	LIANG Jingkun	CHN	6723
12	JANG Woojin	KOR	6531
13	JEOUNG Youngsik	KOR	6485
14	Liam PITCHFORD	ENG	5995
15	WANG Chuqin	CHN	5871
16	Koki NIWA	JPN	5841
17	Patrick FRANZISKA	GER	5822
18	Simon GAUZY	FRA	5623
19	Quadri ARUNA	NGR	5538
20	WONG Chun Ting	HKG	5487
21	Jun MIZUTANI	JPN	5482
22	LEE Sangsu	KOR	5032
23	Marcos FREITAS	POR	4673
24	ZHAO Zihao	CHN	4650
25	Darko JORGIC	SLO	4557
26	Robert GARDOS	AUT	4469
27	CHUANG Chih-Yuan	TPE	4384
28	Kristian KARLSSON	SWE	4214
29	Vladimir SAMSONOV	BLR	4175
30	Kanak JHA	USA	4122
31	Jonathan GROTH	DEN	4113
32	Sharath Kamal ACHANTA	IND	4056
33	Tomislav PUCAR	CRO	3991
34	Yang WANG	SVK	3913
35	Ruwen FILUS	GER	3854
36	Omar ASSAR	EGY	3767
37	Gustavo TSUBOI	BRA	3765
38	Sathyan GNANASEKARAN	IND	3722
39	Yukiya UDA	JPN	3711
40	Emmanuel LEBESSON	FRA	3559
41	AN Jaehyun	KOR	3528
42	Benedikt DUDA	GER	3517
43	Jon PERSSON	SWE	3465
44	Daniel HABESONH	AUT	3450
45	Takuya JIN	JPN	3276
46	Kirill GERASSIMENKO	KAZ	3272
47	Andrej GACINA	CRO	3192
48	Masataka MORIZONO	JPN	3161
49	Panagiotis GIONIS	GRE	3126
50	Anton KALLBERG	SWE	3099
51	Dang QIU	GER	3081
52	Kirill SKACHKOV	RUS	3038
53	Ahmed SALEH	EGY	3035
54	Pavel SIRUCEK	CZE	2998

图 4: 男子单打世界排名

ITTF TABLE TENNIS WORLD RANKING			
Senior Women's Singles			
2021 Week #35			
Rank	Name	Assoc	Points
1	CHEN Meng	CHN	13850
2	SUN Yingsha	CHN	11240
3	Mima ITO	JPN	11198
4	WANG Manyu	CHN	8733
5	DING Ning	CHN	8070
6	ZHU Yuling	CHN	7335
7	LIU Shiwen	CHN	7245
8	CHENG I-Ching	TPE	7175
9	Kasumi ISHIKAWA	JPN	7100
10	FENG Tianwei	SGP	6947
11	WANG Yidi	CHN	6621
12	Miu HIRANO	JPN	6507
13	CHEN Xingtong	CHN	6222
14	JEON Jihee	KOR	6114
15	DOO Hoi Kem	HKG	5966
16	Sofia POLCANOVA	AUT	5641
17	Petrissa SOLJA	GER	5573
18	Adriana DIAZ	PUR	5494
19	Hitomi SATO	JPN	5352
20	Ying HAN	GER	5299
21	SUH Hyowon	KOR	4989
22	HE Zhuojia	CHN	4953
23	Miyu KATO	JPN	4665
24	QIAN Tianyi	CHN	4596
25	CHEN Szu-Yu	TPE	4596
26	Hina HAYATA	JPN	4522
27	Bernadette SZOCS	ROU	4486
28	Britt EERLAND	NED	4435
29	Elizabeta SAMARA	ROU	4256
30	SOO Wai Yam Minnie	HKG	4203
31	YU Mengyu	SGP	4177
32	Lily ZHANG	USA	4149
33	Xiaoxin YANG	MON	4136
34	Margaryta PESOTSKA	UKR	4058
35	Xiaona SHAN	GER	3977
36	Yue WU	USA	3888
37	Mo ZHANG	CAN	3862
38	Dina MESHREF	EGY	3854
39	Qian LI	POL	3837
40	Suthasini SAWETTABUT	THA	3826
41	Honoka HASHIMOTO	JPN	3708
42	Saki SHIBATA	JPN	3690
43	Nina MITTELHAM	GER	3687
44	Xia Lian NI	LUX	3657
45	Polina MIKHAILOVA	RUS	3598
46	GU Yuting	CHN	3591
47	Bruna TAKAHASHI	BRA	3512
48	LEE Ho Ching	HKG	3501
49	Miyuu KIHARA	JPN	3463
50	Matilda EKHOLM	SWE	3399
51	Georgina POTA	HUN	3361
52	Hana MATELOVA	CZE	3350
53	Barbora BALAZOVA	SVK	3316

图 5: 女子单打世界排名

3.2.2 python 模拟比赛

将对局分析分解为以下几个小步骤：

1. 获得程序运行参数：probA（A 的能力值）,probB（B 的能力值）,n（比赛场次）
2. 利用球员 AB 的能力值，模拟 n 场比赛
3. 输出球员 AB 获胜的场次及概率

利用对局规则：回合制，先由一方发球，如胜利，则得 1 分并继续发球，如失败，则双方分数不改变并交换球权。球员 AB 的能力为 0-1 之间的一个数值先得 11 分的一方为胜方；10 平后，先得 2 分的一方为胜方。每颗球的对局采用随机数生成与“能力值”作比较决定得分。由此编程（见代码附录）传入双方能力值（双打则为队友能力值取平均），下方例子中然模拟进行 100 局（5 局 3 胜制最多 500 轮），输出的各自胜率即为交手双方对应的期望。

以樊振东和马龙为例，模拟结果如下：

myx 进行比赛分析结果：
 程序运行需要 A 和 B 的能力值（以 0 到 1 之间的小数表示）
 请输入 A 的能力值 (0-1): 0.7338804388095789
 请输入 B 的能力值 (0-1): 0.7315988916808487
 模拟比赛的场次: 100
 竞技分析开始，共模拟 500 场比赛
 A 获胜 287 场比赛，占比 57.4%
 B 获胜 213 场比赛，占比 42.6%

根据实际情况水平接近，该参数化方法分析模型是可行且优良的。在建模中为了增加准确率使用 1000 局进行模拟。

3.2.3 损耗

由于赛制日程安排紧密，高水平高强度的比赛对于选手的体力的影响是不可忽视的。体力在一定程度上会影响选手的发挥。因此，对于既定收获了男单女单金银牌的四位选手在团体赛开打时的能力需要考虑做出削减。

不妨设历史损耗能力值：

冠亚军的能力只有理论值的 90%，每少打一轮损耗少 2%；混双决赛的 3 号选手的能力为理论值的 93%，每少打一轮损耗少 2.5%（考虑到男女混双对男选手的消耗比单打更大，女 3 为理论能力值的 96.5%，每少打一轮损耗少 0.5%）。

定义单场内损耗能力函数如下：

$$Cap_{t_r} = Cap_{t_{r-1}} * \frac{1}{e^{0.1}} \quad (4)$$

其中， r 是运动员在一场团体赛中出场地轮数， t 是第几场， $1 \leq t \leq 4$ 。

考虑到赛后一直到第二天的下一场体能少量恢复，因此两场比赛损耗能力函数定义为：

$$Cap_t = Cap_{t-1} * 0.95 \quad (5)$$

3.2.4 优化筛选

对理想情况进行枚举遍历过于复杂，因此，考虑使用 MiniMax 算法和 $\alpha - \beta$ 剪枝简化筛选流程。

- MiniMax 算法

极小极大算法从当前状态计算极小极大策略。它使用了简单的递归算法计算每个后继的极小极大值，自上而下一直前进到树的叶结点，然后随着递归回溯把极小极大值回转。^[4]

总结一下 Minimax 算法的步骤：

1. 首先确定最大搜索深度 D，D 可能达到终局，也可能是一个中间格局。
2. 在最大深度为 D 的格局树叶子节点上，使用预定义的价值评价函数对叶子节点价值进行评价。
3. 自底向上为非叶子节点赋值。其中 max 节点取子节点最大值，min 节点取子节点最小值。
4. 每次轮到我方时（此时必处在格局树的某个 max 节点），选择价值等于此 max 节点价值的那个子节点路径。

在模拟时，令中国队获胜的胜率拟合值为 +，德国获胜的拟合值为-来计算每个节点对应的 minimax 值。

- $\alpha - \beta$ 剪枝

将此技术应用到标准的极小极大搜索树上，会减掉那些不可能影响决策的分支，仍然返回和极小极大算法相同的结果。^[4] 核心思想就是，当你知道你有一个选择 A 时，此时你知道了 B 选择不如 A 选择好，那么你就不需要知道 B 选择有多坏。即对于一个结点 MIN（敌人结点），若能估计出其倒推值的上界（遍历的值），若这个上界不大于（小于）MIN 的父节点（最大收益），及时停止那些已无必要再扩展的结点。

至此，建模所需要的数学和计算机工具全部完成。

3.3 具体单场完整建模——以男团决赛为例

由于在决赛之前中国队的水平高于对手比较明显，因此，对于水平一般情况更为接近的决赛进行分析更能体现模型的价值。当然，对于之前的每一场淘汰赛均可以使用该模型进行分析排阵。

由已经过去的 2020 东京奥运会男团决赛为例，决赛是中国队对决德国队，德国队的名单为：奥恰洛夫，波尔，弗朗西斯卡。

观察他们的当时世界排位，分别是：7, 10, 17。

中国队此时樊振东，马龙，许昕经过历史损耗能力值，马龙樊振东分获男单金银牌，许昕获得混双银牌，再由比赛损耗能力函数计算他们的能力值为：

0.48930480078482746, 0.48724162847976304, 0.48758183234964675

德国队三人中，奥恰洛夫获得男单铜牌（与金银牌的比赛轮次数目相同损耗相同），波尔男单 16 强未进入八强少赛 3 轮，弗朗西斯卡混双八强未进入半决赛少赛 2 轮，经过历史损耗能力值和比赛损耗能力函数计算得到：

0.4366343392422034, 0.4344975963299249, 0.4203346107204399

可见，虽然中国队仍然在实力上领先对手，但是经过比赛的损耗，此时的差距已经不是十分显著。

由此时的数据，考虑单场内损耗能力函数，暂不考虑实际情况如许昕是国乒双打王必然参加双打比赛。因此，双方各有三种排兵方式，不妨令 {樊振东，马龙，许昕} 为 {1, 2, 3}; {奥恰洛夫，波尔，弗朗西斯卡} 为 {a, b, c}.

手绘 7 层 minimax 树见图6.

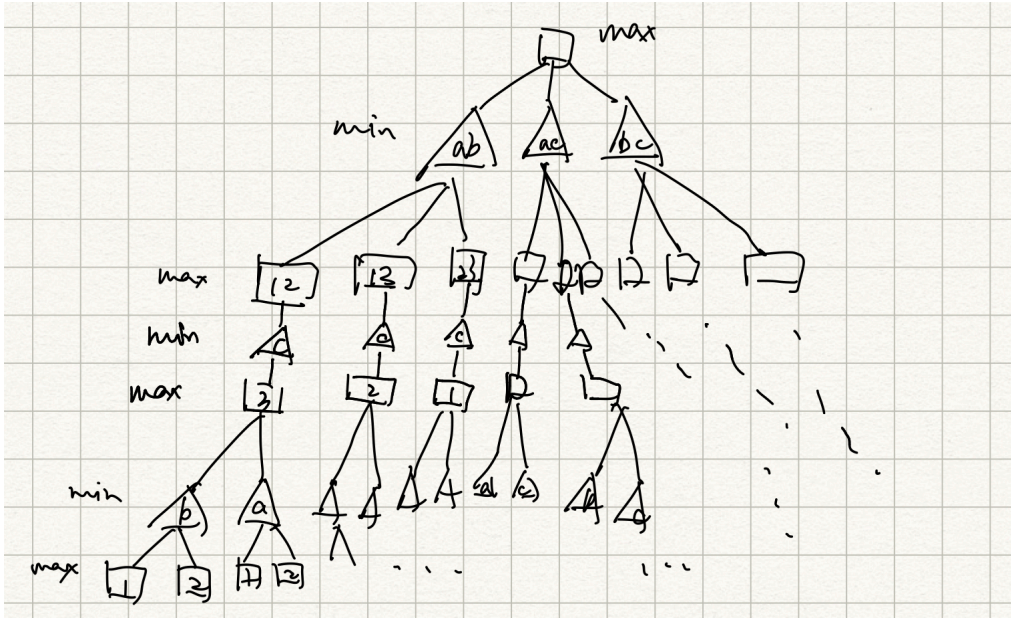


图 6: MiniMax 树

以树的左侧分支分析筛选过程如下：

目的：使得根节点即中国队的获胜期望最大。

• 第一场：双打

第一层：德国队有三种排法：ab,ac,bc. 对应的能力值为：

0.43556596778606416, 0.4284844749813217, 0.42741610352518244

第二层：左侧第一个分支对应中国队的三种排法：12,13,23. 对应的能力值为：

0.4882732146322952, 0.4884433165672371, 0.48741173041470487

暂时先不计算胜率期望，再向下遍历：

• 第二场：单打

由于双打为规则中的 B+C 对 Y+Z，因此，第二场的名单在第一场确定时已经确定。此时，继续向下。（因此，考虑将这两层隐藏。）

• 第三场：单打

第三层：因为双打只决定 BC 是谁，但具体谁是 B 谁是 C 仍然要讨论。德国队此时在 {12} 这颗子树上有两种选择：a,b.

第四层：同理，中国队在 {b} 这颗子树上有两种选择：1,2.

至此，ABC 与 XYZ 三人对应的人已经确定，第四场第五场都已经确定。

3.3.1 预计 3 场结束

带入单场内损耗能力函数利用 Minimax 算法向上更新极大极小值, 可以很快剪枝确定在第三层总共 9 个节点的选择如下:

12->c->3->b->2;
 13->c->2->b->3;
 23->c->1->b->2;
 12->b->3->c->2;
 13->b->2->c->3;
 23->b->1->c->2;
 12->a->3->c->2;
 13->a->2->c->3;
 23->a->1->c->2;

分别对应的中国队的期望值为: 0.7208000000000001, 0.6901999999999999,
 0.7475999999999999, 0.7263999999999999, 0.7398, 0.7388, 0.7432000000000001, 0.742, 0.7498
 此时, 选择最后一路树的枝作为 max 根节点, 即: 樊振东, 马龙, 许昕分别为 ACB (XZY)。

3.3.2 预计 4 场结束

4 场结束存在两种前提, 中国队是主场和中国队是客场, 区别在于主客场之间的第 4,5 场相互的对位是反过来的。

- 若中国队抽到客场:

一共 9 层 minimax 树 36 个叶子结点, 到第七层的 max 结点对应的值计算为:

0.4985999999999999, 0.513, 0.5034000000000001, 0.5073999999999999, 0.5126,
 0.4820000000000000, 0.491, 0.5166, 0.4754000000000000, 0.5054, 0.4889999999999999,
 0.5269999999999999, 0.4180000000000000, 0.4624, 0.5064, 0.4997999999999997,
 0.4168, 0.4289999999999999, 0.4998, 0.5195999999999999, 0.4128000000000000,
 0.4133999999999999, 0.5124000000000001, 0.4805999999999999, 0.425, 0.4226,
 0.4984, 0.5028000000000001, 0.441, 0.4238000000000000, 0.5146, 0.5114000000000001,
 0.4272000000000001, 0.4279999999999999, 0.4820000000000000, 0.5007999999999999

继续向上更新, 计算到第三层的 MAX 结点对应的值为: 0.9976, 1.0026000000000002,
 0.9954, 0.9643999999999999, 0.9465999999999999, 0.9261999999999998, 0.9851999999999999,
 1.0012, 0.9376

故, 此时, 选择樊振东, 马龙, 许昕作为客队的 YXZ 选手上场比赛。与与预计 3 场结束的安排不一样。

- 若中国队抽到主场:

第七层的 max 结点对应的值计算为:

0.3926, 0.4160000000000000, 0.4418, 0.4062000000000001, 0.4098, 0.4344000000000001,
 0.4444000000000001, 0.4108000000000005, 0.4074000000000000, 0.4208000000000006,
 0.427, 0.4184, 0.5056, 0.4924000000000006, 0.5118, 0.4864, 0.4790000000000004,

0.5104, 0.5137999999999999, 0.47580000000000006, 0.5056, 0.4872, 0.49539999999999995,
0.5118, 0.4818, 0.48179999999999995, 0.49240000000000006, 0.52540000000000001,
0.49520000000000003, 0.502, 0.5278, 0.48739999999999994, 0.508, 0.5191999999999999,
0.5304, 0.5065999999999999

继续向上更新, 计算到第三层的 MAX 结点对应的值为: 0.9032, 0.95100000000000001,
0.9242, 0.95380000000000001, 0.9402, 0.97660000000000001, 0.9383999999999999, 0.9708,
0.9813999999999999

故, 此时, 选择樊振东, 马龙, 许昕作为主队的 ACB 选手上场比赛。这个安排与预计 3 场结束的安排是一样的。

3.3.3 预计 5 场结束

5 场结束在计算的时候主客场对于结果是不存在影响到, 因为两场互换顺序的比赛都在最后 4 层单支的树结构上, 相加时不影响结果。

第七层的 max 结点对应的值计算为:

0.9092, 0.895, 0.9352, 0.9196, 0.9072, 0.9141999999999999, 0.93460000000000001,
0.91280000000000001, 0.9322, 0.93240000000000001, 0.9281999999999999,
0.9335999999999999, 0.9345999999999999, 0.9234, 0.9724, 0.97020000000000002,
0.9099999999999999, 0.9298, 0.9747999999999999, 1.01, 0.91280000000000001,
0.89040000000000001, 1.0168, 1.0116, 0.9401999999999999, 0.8997999999999999,
1.0221999999999998, 0.9924, 0.9432, 0.94260000000000001, 1.0286, 1.017, 0.922,
0.9417999999999999, 0.9978, 1.0156

继续向上更新, 计算到第三层的 MAX 结点对应的值为: 1.4308, 1.43820000000000001,
1.4412, 1.45780000000000002, 1.4252, 1.4702, 1.4349999999999998, 1.4777999999999998,
1.46620000000000002

由 minimax 计算结果可以看出, 此时安排樊振东, 马龙, 许昕作为 BAC (YXZ) 选手上场比赛。

3.4 建模结果分析

从对 2020 东京奥运会乒乓球男子团体决赛的建模过程中可以看出, 对于想要在几场拿下比赛, 派出的人员序列是不一样的。从当时的赛况结果看, 中国队抽到主队, 最终以 3: 0 战胜德国队卫冕成功。当时主教练排出的阵型就是:

- A. 樊振东
- B. 许昕
- C. 马龙

与建模过程中分析的预计 3 场结束的阵容完全相同, 也与预计 4 场结束中国队抽到主队的情况相同。证明模型效果十分好! 与建模对比, 主教练也应该是没有想要去进行 5 场比赛, 也想尽可能最快拿到金牌。

4 结论

通过积分量化能力值并使用 minimax 进行对局分析，主教练在选手均正常发挥的情况下只需要将对手和本方队员的数据进行拟合，根据自己预想几场完赛就可以得到非常合适的不同对策。这大大提高了主教练的变阵可靠性和分析的准确性。

根据具体人员，比如说选手在双打的契合度惯用手的匹配度，还可以在计算能力值的时候对这些特征再做加分加权使得该模型更准确好用。

回看文献中的统计记录^[1]:

主客队 ABC/XYZ	主队频次			客队频次		
	A	B	C	X	Y	Z
队内 1 号	30	4	4	19	11	8
队内 2 号	3	18	17	11	7	20
队内 2 号	5	16	17	8	20	10

表 2: 各队运动员出场次序安排统计

从主队的排阵统计看，特征表现较为明确，队内一号排在 A 的位置上的频次高达 30 次，显然，队内 1 号的主要任务是两场单打; 队内 2 号和 3 号主要安排在 B 或 C 的位置上，合计频次也超过 30，且队内 2 号和 3 号分别在 B 或 C 的位置上频次基本相同。队内 2 号与 3 号的主要任务是配合双打，第 3 场与第 5 场单打的安排根据对手情况安排较为灵活。

从客队的排阵统计看，特征表现并不十分明确，队内一号排在 X 的位置上的频次与排在 Y 或 Z 位置上的频次相等，即客队的排阵中，出现队内一号参与双打的概率较大; 当队内 1 号参与双打时，队内 2 号与 3 号出现在 X 位置上的频次也比较接近; 队内 2 号参与双打时，主要安排在 Z 的位置上，而队内 3 号参与双打时，主要安排在 Y 的位置上。

这与建模过程中主客队呈现不同的策略是一致的。再次证明模型的建立十分成功!

代码附录

部分 Python 代码展示如下:

```
1 from numpy import *
2 l = [12494,12212,10356,8709,8375,8034,7999,7864,7357,7086,6723,6531,6485,5995,5871,
      5841,5822,5623,5538,5487,5482,5023,
      4673,4650,4557,4469,4384,4214,4175,
      4122,4113,4056]
3 p = [12494,12212,10356,8709,8375,8034,7999,7864,7357,7086,6723,6531,6485,5995,5871,
      5841]
4 g = [(i-mean(p))/std(p) for i in l]
5
6 import matplotlib.pyplot as plt
7 import numpy as np
8
9 def sigmoid(x):
10     # 直接返回sigmoid函数
```

```

11     return 1. / (1. + np.exp(-x))
12
13 # param: 起点, 终点, 间距
14 x = np.arange(-8, 8, 0.05)
15 y = sigmoid(x)
16 plt.plot(x, y)
17 plt.savefig('sig.pdf')
18 plt.show()
19
20 s = [sigmoid(i) for i in g]
21 def at(x):
22     y = np.arctan(x)/np.pi + 0.5
23     return y
24 x = np.arange(0, 8, 0.05)
25 z = at(x)
26 plt.plot(x, z)
27 plt.savefig('s2.pdf')
28 plt.show()
29
30 ss = [np.arctan(i)/np.pi + 0.5 for i in s]
31
32 fig, ax = plt.subplots()
33 b = ax.barh(range(1, len(s)+1), s)
34 ax.set_yticks(range(1, len(s)+1))
35 # ax.set_yticklabels(s.index)
36 plt.savefig('cap.pdf')
37 plt.show()

```

```

1  ## 演示代码如下
2  from random import random
3  def printIntro():          #打印程序介绍信息
4      print("myx进行比赛分析结果:")
5      print("程序运行需要A和B的能力值(以0到1之间的小数表示)")
6  def getInputs():          #获得程序运行参数
7      a = eval(input("请输入A的能力值(0-1): "))
8      b = eval(input("请输入B的能力值(0-1): "))
9      n = eval(input("模拟比赛的场次: "))
10     return a, b, n
11  def simNGames(n, probA, probB):    # 进行N场比赛
12      winsA, winsB = 0, 0
13      for i in range(n):
14          for j in range(5):        #进行5局3胜的比赛
15              scoreA, scoreB = simOneGame(probA, probB)
16              if scoreA > scoreB:
17                  winsA += 1
18              else:
19                  winsB += 1
20      return winsA, winsB
21  def gameOver(a,b):            #正常比赛结束
22      return a==11 or b==11
23  def gameOver2(a,b):          #进行抢12比赛结束
24      return a==12 or b==12
25  def simOneGame(probA, probB):    #进行一场比赛
26      scoreA, scoreB = 0, 0        #初始化AB的得分
27      serving = "A"
28      while not gameOver(scoreA, scoreB):    #用while循环来执行比赛
29          if scoreA==10 and scoreB==10:
30              return(simtwoGame2(probA,probB))
31          if serving == "A":

```

```

32         if random() < probA:                ##用随机数生成胜负
33             scoreA += 1
34         else:
35             serving="B"
36     else:
37         if random() < probB:
38             scoreB += 1
39         else:
40             serving="A"
41     return scoreA, scoreB
42 def simtwoGame2(probA,probB):
43     scoreA,scoreB=10,10
44     serving = "A"                #假如先让队伍A发球
45     while not gameOver2(scoreA, scoreB):
46         if serving == "A":
47             if random() < probA:
48                 scoreA += 1
49             else:
50                 serving="B"
51         else:
52             if random() < probB:
53                 scoreB += 1
54             else:
55                 serving="A"
56     return scoreA, scoreB
57 def printSummary(winsA, winsB):
58     n = winsA + winsB
59     print("竞技分析开始，共模拟{}场比赛".format(n))
60     print("A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))
61     print("B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))
62
63 def main():
64     printIntro()
65     probA, probB, n = getInputs()
66     winsA, winsB = simNGames(n, probA, probB)
67     printSummary(winsA, winsB)
68 main()
69     ##多轮模拟时修改取消print的内容和传参方式

```

```

1  l1 = [i for i in ss[:3]]
2  l2 = [ss[6],ss[9],ss[16]]
3  l1[0] = l1[0]*.9
4  l1[1] = l1[1]*.9
5  l1[2] = l1[2]*.93
6  l2[0] = l2[0]*0.9
7  l2[1] = l2[1]*0.96
8  l2[2] = l2[2]*0.98
9  l1 = [i/(exp(0.1*3)) for i in l1]
10 l2 = [i/(exp(0.1*3)) for i in l2]
11
12 def lose(a,b,n=100):
13     # printIntro()
14     probA, probB, n = a,b,n
15     winsA, winsB = simNGames(n, probA, probB)
16     return -winsA/(winsA + winsB)
17 def win(a,b,n=100):
18     # printIntro()
19     probA, probB, n = a,b,n
20     winsA, winsB = simNGames(n, probA, probB)

```

```

21     return winsB/(winsA + winsB)
22 def expt(a,b,n=100):
23     e = win(a,b,n) + lose(a,b,n)
24     return e
25
26 cap1 = [(l1[0]+l1[1])/2,(l1[0]+l1[2])/2,(l1[1]+l1[2])/2]
27 cap2 = [(l2[0]+l2[1])/2,(l2[0]+l2[2])/2,(l2[1]+l2[2])/2]
28
29 def loss(x,r=1):
30     x = x/exp(0.1*r)
31     return x
32
33 r1 = expt(cap2[0],cap1[0])+expt(l2[2],l1[2])+expt(loss(l2[1]),loss(l1[1]))
34 r2 = expt(cap2[0],cap1[1])+expt(l2[2],l1[1])+expt(loss(l2[1]),loss(l1[2]))
35 r3 = expt(cap2[0],cap1[2])+expt(l2[2],l1[0])+expt(loss(l2[1]),loss(l1[1]))
36 r1 = expt(cap2[1],cap1[0])+expt(l2[1],l1[2])+expt(loss(l2[2]),loss(l1[1]))
37 r2 = expt(cap2[1],cap1[1])+expt(l2[1],l1[1])+expt(loss(l2[2]),loss(l1[2]))
38 r3 = expt(cap2[1],cap1[2])+expt(l2[1],l1[0])+expt(loss(l2[2]),loss(l1[1]))
39 r1 = expt(cap2[2],cap1[0])+expt(l2[0],l1[2])+expt(loss(l2[2]),loss(l1[1]))
40 r2 = expt(cap2[2],cap1[1])+expt(l2[0],l1[1])+expt(loss(l2[2]),loss(l1[2]))
41 r3 = expt(cap2[2],cap1[2])+expt(l2[0],l1[0])+expt(loss(l2[2]),loss(l1[1]))
42
43 ll1 = [loss(i) for i in l1]
44 ll2 = [loss(i) for i in l2]
45 r = []
46 r.append(expt(ll2[0],ll1[0])+expt(ll2[2],ll1[1]))
47 r.append(expt(ll2[0],ll1[1])+expt(ll2[2],ll1[0]))
48 r.append(expt(ll2[1],ll1[0])+expt(ll2[2],ll1[1]))
49 r.append(expt(ll2[1],ll1[1])+expt(ll2[2],ll1[0]))
50 r.append(expt(ll2[0],ll1[0])+expt(ll2[2],ll1[2]))
51 r.append(expt(ll2[0],ll1[2])+expt(ll2[2],ll1[0]))
52 r.append(expt(ll2[1],ll1[0])+expt(ll2[2],ll1[2]))
53 r.append(expt(ll2[1],ll1[2])+expt(ll2[2],ll1[0]))
54 r.append(expt(ll2[0],ll1[1])+expt(ll2[2],ll1[2]))
55 r.append(expt(ll2[0],ll1[2])+expt(ll2[2],ll1[1]))
56 r.append(expt(ll2[1],ll1[2])+expt(ll2[2],ll1[2]))
57 r.append(expt(ll2[1],ll1[2])+expt(ll2[2],ll1[1]))
58 r.append(expt(ll2[0],ll1[0])+expt(ll2[1],ll1[1]))
59 r.append(expt(ll2[0],ll1[1])+expt(ll2[1],ll1[0]))
60 r.append(expt(ll2[2],ll1[0])+expt(ll2[1],ll1[1]))
61 r.append(expt(ll2[2],ll1[1])+expt(ll2[1],ll1[0]))
62 r.append(expt(ll2[0],ll1[0])+expt(ll2[1],ll1[2]))
63 r.append(expt(ll2[0],ll1[2])+expt(ll2[1],ll1[0]))
64 r.append(expt(ll2[2],ll1[0])+expt(ll2[1],ll1[2]))
65 r.append(expt(ll2[2],ll1[2])+expt(ll2[1],ll1[0]))
66 r.append(expt(ll2[0],ll1[1])+expt(ll2[1],ll1[2]))
67 r.append(expt(ll2[0],ll1[2])+expt(ll2[1],ll1[1]))
68 r.append(expt(ll2[2],ll1[2])+expt(ll2[1],ll1[2]))
69 r.append(expt(ll2[2],ll1[2])+expt(ll2[1],ll1[1]))
70 r.append(expt(ll2[1],ll1[0])+expt(ll2[0],ll1[1]))
71 r.append(expt(ll2[1],ll1[1])+expt(ll2[0],ll1[0]))
72 r.append(expt(ll2[2],ll1[0])+expt(ll2[0],ll1[1]))
73 r.append(expt(ll2[2],ll1[1])+expt(ll2[0],ll1[0]))
74 r.append(expt(ll2[1],ll1[0])+expt(ll2[0],ll1[2]))
75 r.append(expt(ll2[1],ll1[2])+expt(ll2[0],ll1[0]))
76 r.append(expt(ll2[2],ll1[0])+expt(ll2[0],ll1[2]))
77 r.append(expt(ll2[2],ll1[2])+expt(ll2[0],ll1[0]))
78 r.append(expt(ll2[1],ll1[1])+expt(ll2[0],ll1[2]))
79 r.append(expt(ll2[1],ll1[2])+expt(ll2[0],ll1[1]))

```

```

80 r.append(expt(l12[2],l11[2])+expt(l12[0],l11[2]))
81 r.append(expt(l12[2],l11[2])+expt(l12[0],l11[1]))
82 rr = []
83 for i in range(len(r)//2):
84     rr.append(min(r[2*i],r[2*i+1]))
85 rrr = []
86 for i in range(len(rr)//2):
87     rrr.append(max(rr[2*i],rr[2*i+1]))
88
89 r4 = []
90 cnt = 0
91 for i in range(2,-1,-1):
92     for j in range(2,-1,-1):
93         r4.append(expt(l2[i],l1[j])+rrr[cnt])
94         cnt += 1
95 r5 = []
96 cnt = 0
97 for i in range(3):
98     for j in range(3):
99         r5.append(expt(cap2[i],cap1[j])+r4[cnt])
100        cnt += 1

```

参考文献

- [1] 张瑛秋, 周星栋, 金烨, 等. 东京奥运会乒乓球团体赛制改革背景下中国队夺金策略-基于团体新赛制及其排阵策略的视角[J]. 北京体育大学学报, 2019, 42(12):44-52.
- [2] GOH Z. 东京 2020 奥运观赛指南: 乒乓球[EB/OL]. 2021. <https://olympics.com/zh/featured-news/things-to-know-olympic-table-tennis-tokyo-2020>.
- [3] ITTF. Ittf table tennis world ranking[EB/OL]. 2021. <https://www.ittf.com/rankings/>.
- [4] J.RUSSELL S, NORVIG P. 人工智能: 一种现代的方法 (第 3 版)[M]. 北京清华大学学研大厦 A 座: 清华大学出版社, 2013.