

借助模糊数学为目标预算，偏好提供合理的计算机配置解决方案

姚迪熙

F1803303 518021910367

计算机科学技术

Jimmyyao18@sjtu.edu.cn

陈子昊

F1808203 518021910712

生物医学工程

harrychenzh@sjtu.edu.cn

金轩铖

F1702111 517021910725

工业工程

jinxuancheng@sjtu.edu.cn

赵浩辰

F1803405 518021910927

信息工程

zhaohaochen@sjtu.edu.cn

1 背景与相关工作

如今计算机已经成为各行各业，各类群体所需的工具，更宽泛的说笔记本电脑，手机，pad 等等已经融为生活的一体。特别是在今年由于新型冠状病毒的疫情，很多人选择在家办公使得对此的需求更加的重要。因而选购合适的电子产品，如何用合理的价钱买到自己理想的电子设备也越来越重要。有的人喜欢大牌子，比如苹果，华为、外星人、华硕，也有的人喜欢组装机。

在选购计算机的时候会去关注它的配置，计算机的配置有 CPU，显卡，内存，硬盘诸如此类的硬件，有的人则会更加去关注外观，屏幕，莱卡摄像头等等。当我们在考量配置的时候除了去关心多核，主频，内存大小，硬盘大小等比较显而易见的参数以外还有很多细节的内容，比如并行性，指令集，流水线方式等等，内容极其复杂很难用直接的数学公式推导而难以直接量化。使得如何评估一台计算机变得困难。

除了性能，价格也是大家考量的方面，比如 NVIDIA TESLA T4¹的算力很强（拥有 320 个图灵运算核心，2560 个 NVIDIA CUDA® 核心，图 1），但很少会有人花 2 万块钱买一张显卡进行一些日常的工作。同时我们也不会想去花几千块买到不符合自己预期的配置。

模糊数学

模糊数学最早由 L.A. Zadeh 教授提出，在各大领域都有广泛的引用，特别是在计算机领域也有相关的研究，例如利用模糊数学理论进行计算机内部的数学编程，包括线性，非线性和图形化编程 [2]，又比如利用模糊数学理论解决人工智能，神经网络中的问题 [1]。本文探讨的则是计算机配置指标和购买力之间的方案，输入计算机领域与商业领域结合的方向。在本文中使用了模糊数学理论中的包括但不限于模糊集合，模糊概率，模糊识别，模糊聚类以及常见的模糊数学方法 [3], [4]，具体的方法如何开展，将在后续章节展开。

¹<https://www.nvidia.cn/data-center/tesla-t4/>

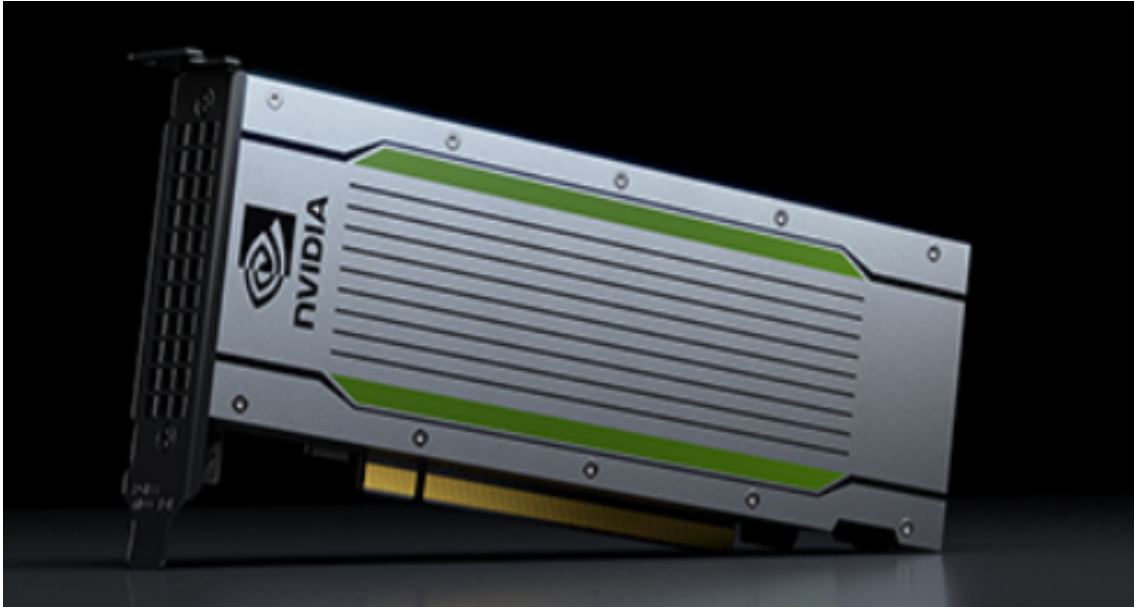


图 1: NVIDIA TESLA T4

计算机硬件指标评价

如何有效的评价计算机的硬件指标是一直以来非常受关注的问题。对于计算机的硬件评价，在 [5] 中提到，目前计算机有非常多的指标，在此不多赘述，具体有哪些细化的指标也不是本文关注的内容，对于计算机硬件评价有多种模型，常见的有测量法，模拟法和模型法。本文采用的方法为模型法。在 [6] 一书中将计算机模型非常详细的量化为了一个多级方程式所构建的数学模型。但是这样复杂的模型对于普通的计算机使用者来说难以理解，更不可能在购买计算机的时候带着这么多的评测模型和数学公式去商场或者网上购物，难以实现。并且也很难给出合理的价格估计，因而缺乏一定的实用价值。

本文基于现实考虑，将模糊数学理论和复杂的计算机硬件指标评价问题结合，对于计算机硬件评价给出合理的模糊数学模型，通过模糊理论来界定那些往往不能够二值化，确定化的数据。例如，一张显卡的好坏并不总是可以一概而论，并且在不同的论域上也有不同的价值。一台计算机可能可以玩游戏有很好的体验，却不能够获得较好的网络吞吐量，而另一台计算机可能有好的吞吐量，却没有合适的图形化界面。本文将这些难以界定的概念，借助大规模的市场统计数据，标准的数据评估指标建立了模糊数学模型，使得能够合理评估在一定价格定义上最合适的硬件配置。最后我们给出了解决方案，当一个用户给出预算和内心比较心仪的一些要求后，我们可以提供最好的购买建议，解决实际问题。本文的主要工作如下

- 量化了四个主要计算机硬件的各项指标
- 利用模糊集合隶属度量化了对于各个硬件的评价
- 借助模糊评价手段，给出用户购买计算机硬件的方案

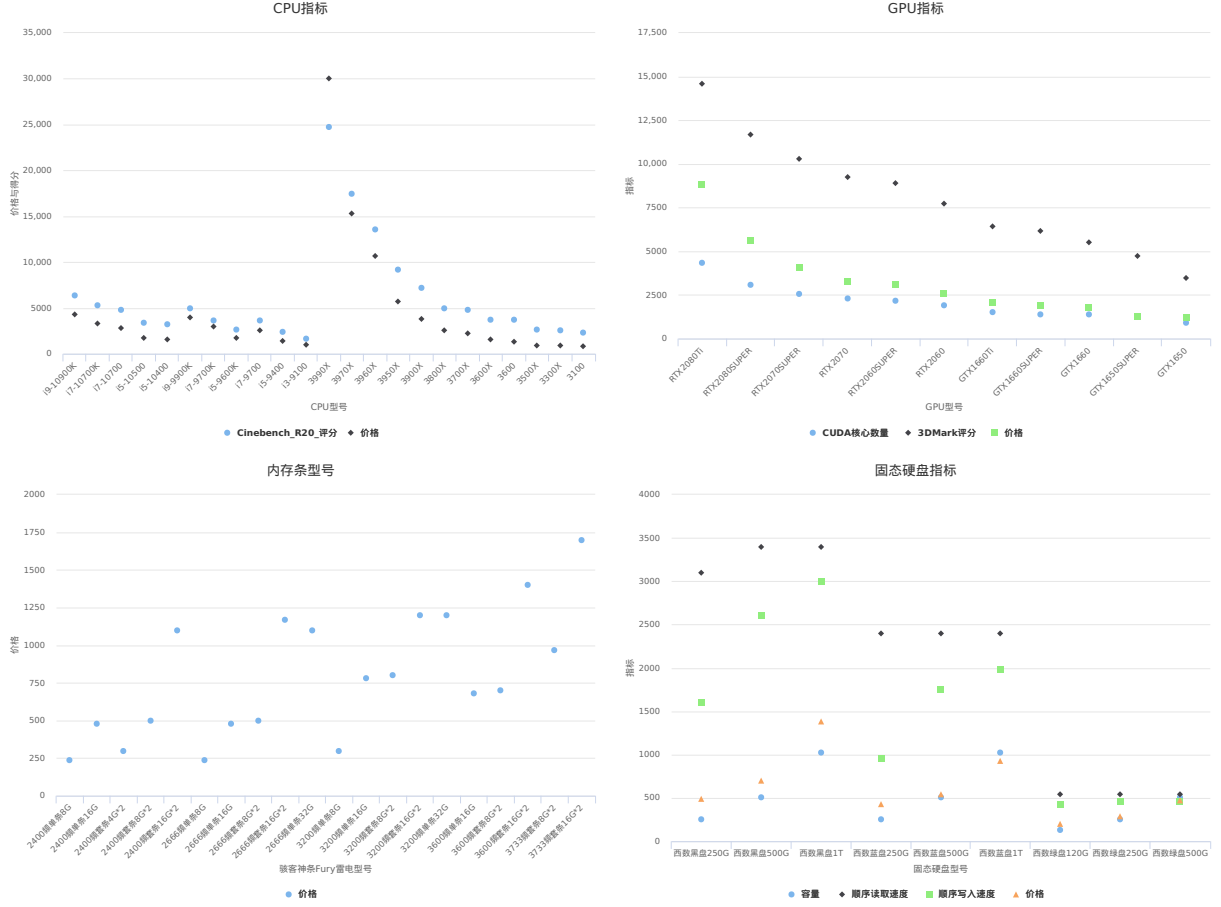


图 2: 四个主要硬件对象的部分指标

本文的组织结构为，在 2 章节会介绍数据的处理，在第 3, 4 章节介绍如何建立模糊分析，在第 5 章节给出最终的解决方案。

2 数据处理

我们综合了常见的计算机硬件的多项指标，从各大权威网站 (附录 A) 获得数据并做了一定的数据过滤和筛选选取所需要的数据，我们按照网站上的指标将一些零碎的重要性低的内容去除，在获得网站数据以后我们将一些错误的信息去除。经过筛选发现计算机在评价的时候主要是四种硬件：cpu、显卡、固态硬盘、内存，因此这是我们的对象集合

$$U = \{u_{cpu}, u_{gpu}, u_{ssd}, u_{sdram}\}$$

我们囊括了各个不同对象的可量化的参数指标作为我们的指标集合。而对于每一种硬件同样的我们也将那些时新度过低，很久以前已经被淘汰的商品去除，我们按照购买热度和流行度选取那些比较主流，也是市面上大家喜欢购买也买得到的硬件作为我们评价分析的对象。

另外主板、电源、机箱等其余外设硬件在各个计算机之间大同小异没有显著的区别，主要

影响的是继电量和外观，并不会对计算机产生过多或者极其显著的性能变化，而且价格之间也相差不多，作为方案提供商也没有必要过度的去思考客户要买那个机箱鼠标等等，这些因人而异，主要与审美爱好有关。因此我们不作为本文的着重考察内容。我们经过主流购物渠道的调研，这些硬件可以统一定价为 1300 元 (附录 A)。

接下来我们先做了一些预分析，如下是我们对于这四个主要对象部分的指标所做的可视化处理，价格单位是元 (RMB)。如图 2 所示。可以看见这些硬件的各个指标波动范围还是非常大的，有的指标相互之间有一定的关联性，而有的指标时间又没有什么关联性规律性，有的指标变化的规律非常明显，而有的却难以分析。因此我们需要建立合理的方法进行评测和分析。详细的我们所使用的数据见附录 B。

3 模糊集合建立

在获得我们相应的数据以后我们就可以对于各个硬件建立模糊集合。首先我们分别对于每一种硬件进行隶属度的计算。我们作如下的定义，对于每种硬件我们都对三个维度进行模糊函数的建立，分别是价格 (price)，总体表现 (overall) 和第二重要性表现 (secondary overall)。具体的每种硬件的三个维度的内容如表 1 所示。定义论域 U 为各个硬件的对应维度，包含三个类别的模糊统计分别是高、中、低，对于每个硬件的每个维度我们都计算相应的隶属度。

所以宏观上我们的论域是硬件总数 \times 维度数这么多元素，分别对于三个方面即高、中、低进行隶属度的计算。

硬件	Overall	Secondary Overall	Price
CPU	整体性能	单核性能	价格
GPU	运算性能	显存容量	价格
内存	频率	容量	价格
固态硬盘	读写速度	容量	价格

表 1: 各硬件的模糊评价维度

接下来我们计算隶属度，我们需要构建隶属度函数，因为各个性能指标比较离散，我们使用升半梯形分布，中间形梯形分布，降半梯形分布隶属函数来作为我们的高，中，低三种情况的隶属度。因为此隶属函数比较容易实现，并且我们在进行隶属函数的确定之前，先将我们用于建立隶属函数的数据进行了分析，我们的数据适合使用对应的隶属函数来进行表示，呈现出与该图形类似的趋势。三种隶属函数的解析式如下所示

- 升半梯形分布

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x \geq b \end{cases}$$

- 中间形梯形分布

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c < x < d \\ 0 & x \geq d \end{cases}$$

- 降半梯形分布

$$\mu(x) = \begin{cases} 1 & x \leq a \\ \frac{b-x}{b-a} & a < x < b \\ 0 & x \geq b \end{cases}$$

使用上述的函数我们对各个硬件作模糊统计，对于每一款硬件我们都确定相应的模糊隶属函数的参数。每一款硬件的每一个维度都有四个参数，分别对应上述公式中的 a, b, c, d 。当 b 和 c 相等的时候中间形梯形会退化为三角形。计算出的各参数如表 2 所示。

CPU	Overall	Secondary Overall	Price
a	2000	3.5	2000
b	6000	3.8	4000
c	8000	4.3	4000
d	10000	4.8	6000
GPU	Overall	Secondary Overall	Price
a	4	3000	1000
b	6	4000	4000
c	6	8000	4000
d	8	1200	6000
Mem	Overall	Secondary Overall	Price
a	2400	8	300
b	2666	16	700
c	2666	16	700
d	3200	32	1000
SSD	Overall	Secondary Overall	Price
a	500	200	300
b	1000	500	700
c	1000	500	700
d	2000	800	1000

表 2: 各硬件隶属函数参数

依照隶属函数我们就可以算出每一个硬件的每一个需要衡量的维度关于高，中，低三种情况的

类别	overall	secondary overall	price
RTX2080Ti	[1,0,0]	[1,0,0]	[1,0,0]
RTX2080SUPER	[0.9625,0.03750,0]	[1,0,0]	[0.7995,0.2005,0]
RTX2070SUPER	[0.786125,0.213875,0]	[1,0,0]	[0.0495,0.9505,0]
RTX2070	[0.65675,0.34325,0]	[1,0,0]	[0,0.7663,0.23367]
RTX2060SUPER	[0.6135,0.3865,0]	[1,0,0]	[0,0.69967,0.3003]
RTX2060	[0.4655,0.5345,0]	[0,1,0]	[0,0.533,0.467]
GTX1660Ti	[0.304875,0.695125,0]	[0,1,0]	[0,0.3663,0.63367]
GTX1660SUPER	[0.273125,0.726875,0]	[0,1,0]	[0,0.29967,0.7003]
GTX1660	[0.186875,0.813125,0]	[0,1,0]	[0,0.2663,0.73367]
GTX1650SUPER	[0.092,0.908,0]	[0,0,1]	[0,0.093,0.907]
GTX1650	[0,0.455,0.545]	[0,0,1]	[0,0.0663,0.93367]

表 3: GPU 评价各个维度对于高、中、低的隶属度

隶属情况，表 3展示了 GPU 的各个维度包括 GPU 硬件的总体的性能属性，GPU 硬件的显存和 CPU 的价格的高、中、低三个维度所呈现的结果。分别计算了相应的隶属度。其他硬件的隶属度请参考附录 C。用于计算隶属度的代码请参考附录 D。

4 模糊综合评价

依照我们通过模糊集合建立的结果，我们对于每一个硬件都有三个指标对于高中低的隶属度，依照这些结果我们需要糊集合建立的结果，我们对于每一个硬件都有三个指标对于高中低的隶属度，依照这些结果我们需要对每一款硬件给出一个最终的得分。

首先根据模糊隶属度函数给出 CPU、显卡、内存、固态硬盘等在各个指标中的评价程度（分为高中低三个评价）。由于每个关键组成部件都有三个指标，因此可以组成一个 3*3 的单因素评判矩阵。接下来我们对于给出每个部件不同指标的权重，运用模糊矩阵的乘法运算可以得到各个部件的多因素评价矩阵。

对于每一个硬件我们的权重指标选取如下

CPU: (单核性能, 整体性能, 价格) = (0.5, 0.3, 0.2)

显卡: (显存容量, 运算性能, 价格) = (0.35, 0.4, 0.25)

内存: (容量, 频率, 价格) = (0.45, 0.4, 0.15)

固态: (容量, 速度, 价格) = (0.5, 0.4, 0.1)

运用模糊一级评价我们可以获得计算的每一个硬件的品质结果。

1. CPU 计算结果

CPU 种类	高	中	低
i9-10900K	0.5	0.2	0.3
i7-10700K	0.5	0.2	0.3
i7-10700	0.5	0.2	0.3
i5-10500	0.333333	0.416667	0.25
i5-10400	0	0.625	0.375
i9-9900K	0.5	0.2	0.3
i7-9700K	0.5	0.2	0.3
i5-9600K	0.416667	0.333333	0.25
i7-9700	0.5	0.2	0.3
i5-9400	0	0.555556	0.444444
i3-9100	0	0.625	0.375
3990X	0.375	0.625	0
3970X	0.444444	0.555556	0
3960X	0.444444	0.555556	0
3950X	0.625	0.375	0
3900X	0.416667	0.333333	0.25
3800X	0.333333	0.416667	0.25
3700X	0.2	0.5	0.3
3600X	0.2	0.5	0.3
3600	0	0.625	0.375
3500X	0	0.555556	0.444444
3300X	0	0.625	0.375
3100	0	0.285714	0.714286

表 4: CPU 评价矩阵

2. GPU 计算结果

GPU 种类	高	中	低
RTX2080Ti	1	0	0
RTX2080SUPER	0.666112	0.333888	0
RTX2070SUPER	0.615385	0.384615	0
RTX2070	0.409452	0.351361	0.239188
RTX2060SUPER	0.385914	0.37289	0.241196
RTX2060	0.380952	0.380952	0.238095
GTX1660Ti	0.319283	0.418903	0.261814
GTX1660SUPER	0.29587	0.433311	0.270819
GTX1660	0.223301	0.477969	0.29873
GTX1650SUPER	0.109264	0.475059	0.415677
GTX1650	0	0.5	0.5

表 5: GPU 评价矩阵

3. 内存计算结果

内存种类	高	中	低
骇客神条 Fury 雷电 2400 频单条 8G	0	0	1
骇客神条 Fury 雷电 2400 频单条 16G	0	0.470588	0.529412
骇客神条 Fury 雷电 2400 频套条 4G*2	0	0	1
骇客神条 Fury 雷电 2400 频套条 8G*2	0	0.470588	0.529412
骇客神条 Fury 雷电 2400 频套条 16G*2	0.470588	0	0.529412
骇客神条 Fury 雷电 2666 频单条 8G	0	0.529412	0.470588
骇客神条 Fury 雷电 2666 频单条 16G	0	0.75	0.25
骇客神条 Fury 雷电 2666 频套条 8G*2	0	0.75	0.25
骇客神条 Fury 雷电 2666 频套条 16G*2	0.470588	0.529412	0
骇客神条 Fury 雷电 2666 频单条 32G	0.470588	0.529412	0
骇客神条 Fury 雷电 3200 频单条 8G	0.529412	0	0.470588
骇客神条 Fury 雷电 3200 频单条 16G	0.529412	0.470588	0
骇客神条 Fury 雷电 3200 频套条 8G*2	0.529412	0.470588	0
骇客神条 Fury 雷电 3200 频套条 16G*2	1	0	0
骇客神条 Fury 雷电 3200 频单条 32G	1	0	0
骇客神条 Fury 雷电 3600 频单条 16G	0.498615	0.443213	0.058172
骇客神条 Fury 雷电 3600 频套条 8G*2	0.527859	0.469208	0.002933
骇客神条 Fury 雷电 3600 频套条 16G*2	1	0	0
骇客神条 Fury 雷电 3733 频套条 8G*2	0.529412	0.470588	0
骇客神条 Fury 雷电 3733 频套条 16G*2	1	0	0

表 6: 内存评价矩阵

4. 固态硬盘计算结果

固态硬盘种类	高	中	低
西数黑盘 250G	0.46875	0.15625	0.375
西数黑盘 500G	0.555556	0.444444	0
西数黑盘 1T	1	0	0
西数蓝盘 250G	0.408163	0.265306	0.326531
西数蓝盘 500G	0.555556	0.444444	0
西数蓝盘 1T	1	0	0
西数绿盘 120G	0	0.166667	0.833333
西数绿盘 250G	0	0.25	0.75
西数绿盘 500G	0.042553	0.425532	0.531915

表 7: 固态硬盘评价矩阵

根据综合打分我们便可以获得各个硬件的评价结果，我们从上述表格可以获得一些可观的结论。

1. 由隶属度最大原则，综合评价为优秀，性价比比较高的 CPU 种类为 *i9-10900K* , *i7-10700K* , *i7-10700* , *i9-9900K* , *i7-9700K* , *i5-9600K* , *i7-9700* , *3950X* , *3900X*
2. 由隶属度最大原则,综合评价为优秀,性价比比较高的 GPU 种类为 *RTX2080Ti*, *RTX2080SUPER*, *RTX2070SUPER*, *RTX2070*, *RTX2060SUPER*, *RTX2060*
3. 由隶属度最大原则，综合评价为优秀，性价比比较高的内存条种类为骇客神条 *Fury* 雷电 3200 频单条 8G, 骇客神条 *Fury* 雷电 3200 频单条 16G, 骇客神条 *Fury* 雷电 3200 频套条 8G*2, 骇客神条 *Fury* 雷电 3200 频套条 16G*2, 骇客神条 *Fury* 雷电 3200 频单条 32G, 骇客神条 *Fury* 雷电 3600 频套条 8G*2, 骇客神条 *Fury* 雷电 3600 频套条 16G*2, 骇客神条 *Fury* 雷电 3733 频套条 8G*2, 骇客神条 *Fury* 雷电 3733 频套条 16G*2
4. 由隶属度最大原则，综合评价为优秀，性价比比较高的固态硬盘种类为西数黑盘 250G, 西数黑盘 500G, 西数黑盘 1T, 西数蓝盘 250G, 西数蓝盘 500G, 西数蓝盘 1T

求解上述内容的代码见附录 E

5 客户方案

根据我们现在获得的数据，用户便可以按照自己的情况进行硬件选择。我们可以绘制下面的策略图如图 3 所示，图中纵轴代表隶属度，横轴代表产品的型号，序号的排列规则分别是组内序号，1 就是代表第一号 CPU 或者第一号 GPU 或者第一号内存条或者第一号固态硬盘依照颜色和图例决定。对应的序号和实际的产品型号参考附录 F。

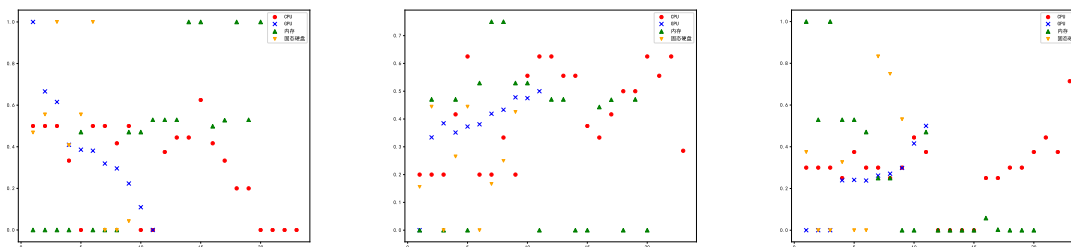


图 3: 硬件质量策略图, 从左到右依次为高、中、低

首先客户按照自己的期望和心理价位的预判有一个基本的考量, 大致的决定自己要买高中低哪一个水准的电脑。根据自己心里的水准, 对于不同的水准查找相应的模糊评价结果或者策略图。

然后按照策略图或者模糊评价表格选取自己心仪的硬件, 如果不知道要买什么好可以按照隶属度就高原则选取隶属度比较高的硬件作为自己的电脑配置。如果几款硬件的隶属度相同可以按照一些固定指标诸如选择价格低的或者选择外型酷炫的等等。

选取的原则分成两步

- 第一步如果有确认一定要买的硬件, 直接选择和这款或者这几款硬件隶属度最接近的硬件购买。
- 第二步, 对于没有特殊需求的硬件, 直接选取隶属度最高的硬件买。

综合上述我们就可以为客户指定最适合和最合适的计算机硬件购买策略。举个例子, 我们小组成员喜欢用高质量的电脑, 我们按照模糊综合评价的结果进行计算最终在购买电脑的时候, 购买一台拥有固态硬盘为西数蓝盘 500G, 内存条为骇客神条 Fury 雷电 3200 频单条 8G, GPU 为 RTX2070SUPER, CPU 为 i7-9700 的电脑, 在我们小组看来是比较合理的。具体如何选择还是需要看自己的需求以及允许的资金情况。这个配置也算是比较高端的配置, 相互之间的硬件也是匹配较为良好的, 是合理的相互之间的搭配。当然也要在资金允许的情况下。

6 总结

往往在购买计算机的时候会举棋不定, 不知道如何能够挑选最适合自己的计算机硬件配置。本小组利用了模糊数学, 模糊数学统计, 模糊集合理论, 模糊综合评价等手段有效的将计算机的硬件指标可观的进行了量化。给出了合理的评价指标。并且最终给出了用户在挑选计算机的时候如何选择最合理的计算机硬件配置。

同时, 在之后也可以增加更多的后续工作, 我们可以在现有基础上增加更多的硬件型号品类, 可以增加更多的指标, 可以有更多的类型不仅限于高、中、低三种情况。可以尝试更多种类的模糊隶属度函数。可以使用更丰富的手段诸如模糊聚类, 模糊规划等等手段使得硬件的挑选更加的细化, 定制化, 能够进一步分拓展。

总体上来说我们成功的使用合适的手段, 完成了预期的计划!

致谢

感谢 MA500 模糊数学及其应用课程授课教师马春翔教授，以及每一位参与课程的同学！

参考文献

- [1] M. Syed Ali, Govindasamy Narayanan, Vineet Shekher, Hamed Alsulami, Tareq Saeed, Dynamic stability analysis of stochastic fractional-order memristor fuzzy BAM neural networks with delay and leakage terms, Applied Mathematics and Computation, Volume 369, 2020, 124896, ISSN 0096-3003, <https://doi.org/10.1016/j.amc.2019.124896>.
- [2] Bing-Yuan Cao and Ji-hui Yang, Xue-Gang Zhou, Zeinab Kheiri, Faezeh Zahmatkesh, Xiao-Peng Yang, Fuzzy Relational Mathematical Programming - Linear, Nonlinear and Geometric Programming Models, Studies in Fuzziness and Soft Computing, 389, Springer, 2020, <https://doi.org/10.1007/978-3-030-33786-5>
- [3] Zadeh L A . Fuzzy Sets[J]. Information & Control, 1965, 8(3):338-353.
- [4] Mordeson J N . Fuzzy Mathematics[M]// Foundations of Image Understanding. Physica-Verlag, 2011.
- [5] 兰德尔 E. 布莱恩特. 深入理解计算机系统：机械工业出版社，2016
- [6] 翟永, 刘津, 陈杰, 刘磊, 邢绪超, 杜江. 计算机硬件设备的运维决策和评估数学模型分析 [J]. 计算机科学, 2018, 45(6A): 568-572, 579. ZHAI Yong, LIU Jin, CHEN Jie, LIU Lei, XING Xu-chao, DU Jiang. Analysis on Mathematical Models of Maintenance Decision and Efficiency Evaluation of Computer Hardware[J]. Computer Science, 2018, 45(6A): 568-572, 579.

Appendices

A 数据来源

cpu

英特尔 10 代处理器 <https://item.jd.com/100011978522.html>

英特尔 9 代处理器 <https://item.jd.com/100000634429.html>

AMD3 代处理器 <https://item.jd.com/100006445340.html>

Cinebench R20 评分 (多核) https://www.cpu-monkey.com/en/cpu_benchmark-cinebench_r20_multi_core-10

显卡

七彩虹高端显卡 <https://item.jd.com/8963629.html#crumb-wrap>

七彩虹中端显卡 <https://item.jd.com/100002004455.html>

七彩虹低端显卡 <https://item.jd.com/100005665255.html>

3DMark 评分 <https://benchmarks.ul.com/compare/best-gpus>

固态

西数固态硬盘 <https://item.jd.com/100003226990.html#crumb-wrap>

内存

金士顿内存骇客神条 Fury 雷电系列 <https://item.jd.com/2551103.html>

主板 + 电源 + 机箱 ≈ 1300

微星主板迫击炮 550 <https://item.jd.com/6833426.html#crumb-wrap>

振华电源 650W 半模组 550 <https://item.jd.com/600637.html#crumb-wrap>

先马坦克机箱 200 <https://item.jd.com/1579807.html#crumb-wrap>

B 采用数据

cpu 型号	核心数	线程数	主频	最高睿频	Cinebench_R20_ 评分	价格
i9-10900K	10	20	3.7	5.3	6399	4299
i7-10700K	8	16	3.8	5.1	5292	3299
i7-10700	8	16	2.9	4.8	4834	2799
i5-10500	6	12	3.1	4.5	3392	1699
i5-10400	6	12	2.9	4.3	3197	1599
i9-9900K	8	16	3.6	5	4995	3999
i7-9700K	8	8	3.6	4.9	3656	2999
i5-9600K	6	6	3.7	4.6	2624	1699
i7-9700	8	8	3	4.7	3644	2599
i5-9400	6	6	2.9	4.1	2407	1399
i3-9100	4	4	3.6	4.2	1630	999
3990X	64	128	2.9	4.3	24763	29999
3970X	32	64	3.7	4.5	17444	15299
3960X	24	48	3.8	4.5	13552	10699
3950X	16	32	3.5	4.7	9148	5749
3900X	12	24	3.8	4.6	7178	3799
3800X	8	16	3.9	4.5	4960	2549
3700X	8	16	3.6	4.4	4834	2249
3600X	6	12	3.8	4.4	3751	1549
3600	6	12	3.6	4.2	3689	1349
3500X	6	6	3.6	4.1	2652	949
3300X	4	8	3.8	4.3	2588	899
3100	4	8	3.6	3.9	2351	799

表 8: CPU 相关数据

显卡型号	显存容量	核心频率	CUDA 核心数量	3DMark 评分	价格
RTX2080Ti	11	1635	4352	14593	8799
RTX2080SUPER	8	1815	3072	11700	5599
RTX2070SUPER	8	1770	2560	10289	4099
RTX2070	8	1620	2304	9254	3299
RTX2060SUPER	8	1650	2176	8908	3099
RTX2060	6	1680	1920	7724	2599
GTX1660Ti	6	1770	1536	6439	2099
GTX1660SUPER	6	1785	1408	6185	1899
GTX1660	6	1860	1408	5495	1799
GTX1650SUPER	4	1755	1280	4736	1279
GTX1650	4	1860	896	3455	1199

表 9: GPU 相关数据

固态型号	容量	顺序读取速度	顺序写入速度	价格
西数黑盘 250G	250	3100	1600	489
西数黑盘 500G	512	3400	2600	699
西数黑盘 1T	1024	3400	3000	1379
西数蓝盘 250G	250	2400	950	429
西数蓝盘 500G	512	2400	1750	539
西数蓝盘 1T	1024	2400	1980	929
西数绿盘 120G	128	545	430	199
西数绿盘 250G	250	545	465	289
西数绿盘 500G	512	545	465	469

表 10: 固态硬盘相关数据

内存型号	容量	频率	价格
骇客神条 Fury 雷电 2400 频单条 8G	8	2400	239
骇客神条 Fury 雷电 2400 频单条 16G	16	2400	479
骇客神条 Fury 雷电 2400 频套条 4G*2	8	2400	299
骇客神条 Fury 雷电 2400 频套条 8G*2	16	2400	499
骇客神条 Fury 雷电 2400 频套条 16G*2	32	2400	1099
骇客神条 Fury 雷电 2666 频单条 8G	8	2666	239
骇客神条 Fury 雷电 2666 频单条 16G	16	2666	479
骇客神条 Fury 雷电 2666 频套条 8G*2	16	2666	499
骇客神条 Fury 雷电 2666 频套条 16G*2	32	2666	1169
骇客神条 Fury 雷电 2666 频单条 32G	32	2666	1099
骇客神条 Fury 雷电 3200 频单条 8G	8	3200	299
骇客神条 Fury 雷电 3200 频单条 16G	16	3200	778.7
骇客神条 Fury 雷电 3200 频套条 8G*2	16	3200	799
骇客神条 Fury 雷电 3200 频套条 16G*2	32	3200	1199
骇客神条 Fury 雷电 3200 频单条 32G	32	3200	1199
骇客神条 Fury 雷电 3600 频单条 16G	16	3600	679
骇客神条 Fury 雷电 3600 频套条 8G*2	16	3600	699
骇客神条 Fury 雷电 3600 频套条 16G*2	32	3600	1399
骇客神条 Fury 雷电 3733 频套条 8G*2	16	3733	969
骇客神条 Fury 雷电 3733 频套条 16G*2	32	3733	1699

表 11: 内存相关数据

C 各硬件隶属度

类别	overall	secondary overall	price
i9-10900K	[0,0.1995,0.8005]	[1,0,0]	[0.1495,0.8505,0]
i7-10700K	[0,0,1]	[1,0,0]	[0,0.6495,0.3505]
i7-10700	[0,0,1]	[1,0,0]	[0,0.3995,0.6005]
i5-10500	[0,0,1]	[0.4,0.6,0]	[0,0,1]
i5-10400	[0,0,1]	[0,1,0]	[0,0,1]
i9-9900K	[0,0,1]	[1,0,0]	[0,0.9995,0.0005]
i7-9700K	[0,0,1]	[1,0,0]	[0,0.4995,0.5005]
i5-9600K	[0,0,1]	[0.6,0.4,0]	[0,0,1]
i7-9700	[0,0,1]	[0.8,0.20,0]	[0,0.2995,0.7005]
i5-9400	[0,0,1]	[0,0.6,0.4]	[0,0,1]
i3-9100	[0,0,1]	[0,0.8,0.20]	[0,0,1]
3990X	[1,0,0]	[0,1,0]	[1,0,0]
3970X	[1,0,0]	[0.4,0.6,0]	[1,0,0]
3960X	[1,0,0]	[0.4,0.6,0]	[1,0,0]
3950X	[0.574,0.426,0]	[0.8,0.2,0]	[0.8745,0.1255,0]
3900X	[0,0.589,0.411]	[0.6,0.4,0]	[0,0.8995,0.1005]
3800X	[0,0,1]	[0.4,0.6,0]	[0,0.2745,0.7255]
3700X	[0,0,1]	[0.2,0.80,0]	[0,0.1245,0.8755]
3600X	[0,0,1]	[0.2,0.80,0]	[0,0,1]
3600	[0,0,1]	[0,0.8,0.20]	[0,0,1]
3500X	[0,0,1]	[0,0.6,0.4]	[0,0,1]
3300X	[0,0,1]	[0,1,0]	[0,0,1]
3100	[0,0,1]	[0,0.2,0.8]	[0,0,1]

表 12: CPU 评价各个维度对于高、中、低的隶属度

类别	overall	secondary overall	price
骇客神条 Fury 雷电 2400 频单条 8G	[0,0,1]	[0,0,1]	[0,0,1]
骇客神条 Fury 雷电 2400 频单条 16G	[0,0,1]	[0,1,0]	[0,0.4475,0.5525]
骇客神条 Fury 雷电 2400 频套条 4G*2	[0,0,1]	[0,0,1]	[0,0,1]
骇客神条 Fury 雷电 2400 频套条 8G*2	[0,0,1]	[0,1,0]	[0,0.4975,0.5025]
骇客神条 Fury 雷电 2400 频套条 16G*2	[0,0,1]	[1,0,0]	[1,0,0]
骇客神条 Fury 雷电 2666 频单条 8G	[0,1,0]	[0,0,1]	[0,0,1]
骇客神条 Fury 雷电 2666 频单条 16G	[0,1,0]	[0,1,0]	[0,0.4475,0.5525]
骇客神条 Fury 雷电 2666 频套条 8G*2	[0,1,0]	[0,1,0]	[0,0.4975,0.5025]
骇客神条 Fury 雷电 2666 频套条 16G*2	[0,1,0]	[1,0,0]	[1,0,0]
骇客神条 Fury 雷电 2666 频单条 32G	[0,1,0]	[1,0,0]	[1,0,0]
骇客神条 Fury 雷电 3200 频单条 8G	[1,0,0]	[0,0,1]	[0,0,1]
骇客神条 Fury 雷电 3200 频单条 16G	[1,0,0]	[0,1,0]	[0.2623,0.73767,0]
骇客神条 Fury 雷电 3200 频套条 8G*2	[1,0,0]	[0,1,0]	[0.300,0.67,0]
骇客神条 Fury 雷电 3200 频套条 16G*2	[1,0,0]	[1,0,0]	[1,0,0]
骇客神条 Fury 雷电 3200 频单条 32G	[1,0,0]	[1,0,0]	[1,0,0]
骇客神条 Fury 雷电 3600 频单条 16G	[1,0,0]	[0,1,0]	[0,0.9475,0.0525]
骇客神条 Fury 雷电 3600 频套条 8G*2	[1,0,0]	[0,1,0]	[0,0.9975,0.0025]
骇客神条 Fury 雷电 3600 频套条 16G*2	[1,0,0]	[1,0,0]	[1,0,0]
骇客神条 Fury 雷电 3733 频套条 8G*2	[1,0,0]	[0,1,0]	[0.8967,0.103,0]
骇客神条 Fury 雷电 3733 频套条 16G*2	[1,0,0]	[1,0,0]	[1,0,0]

表 13: 内存评价各个维度对于高、中、低的隶属度

类别	overall	secondary overall	price
西数黑盘 250G	[1,0,0]	[0,0.167,0.833]	[1,0,0]
西数黑盘 500G	[1,0,0]	[0.04,0.960,0]	[1,0,0]
西数黑盘 1T	[1,0,0]	[1,0,0]	[1,0,0]
西数蓝盘 250G	[0.675,0.325,0]	[0,0.167,0.833]	[0.833,0.167,0]
西数蓝盘 500G	[1,0,0]	[0.04,0.96,0]	[1,0,0]
西数蓝盘 1T	[1,0,0]	[1,0,0]	[1,0,0]
西数绿盘 120G	[0,0,1]	[0,0,1]	[0,0.325,0.675]
西数绿盘 250G	[0,0.01,0.99]	[0,0.167,0.833]	[0,0.4125,0.5875]
西数绿盘 500G	[0,0.01,0.99]	[0.04,0.96,0]	[0,0.4125,0.5875]

表 14: 固态硬盘评价各个维度对于高、中、低的隶属度

D 隶属度求解代码

CPU 隶属度函数求解代码

```
function [HF,MF,LF,Hmark,Mmark,Lmark,Hprice,Mprice,Lprice]=fuzzycpu(F,mark,price)
HF=(F>4.8)+2*(F-4.3).*(F>4.3&F<=4.8);
MF=2*(4.8-F).*(F>4.3&F<=4.8)+2*(F-3.8).*(F>3.8&F<=4.3);
LF=(F<=3.5)+2*(4.3-F).*(F>3.8&F<=4.3);

Hmark=(mark>10000)+(mark>8000&mark<=10000).*(mark-8000)./2000;
Mmark=(mark>8000&mark<=10000).*(10000-mark)./2000+(mark>6000&mark<=8000).*(mark-6000)./2000;
Lmark=(mark<=6000)+1/2000.*(mark>6000&mark<=8000).*(8000-mark);

Hprice=(price>6000)+(price>4000&price<=6000).*(price-4000)./2000;
Mprice=(price>4000&price<=6000).*(6000-price)./2000+(price>2000&price<=4000).*(price-2000)./2000;
Lprice=(price<=2000)+(price>2000&price<=4000).*(4000-price)./2000;
return
end
```

GPU 隶属度函数求解代码

```
function [HVRAM,MVRAM,LVRAM,Hmark,Mmark,Lmark,Hprice,Mprice,Lprice]=fuzzygpu(VRAM,mark,price)
HVRAM=(VRAM>8)+1/2*(VRAM-6).*(VRAM>6&VRAM<=8);
MVRAM=1/2*(8-VRAM).*(VRAM>6&VRAM<=8)+1/2*(VRAM-4).*(VRAM>4&VRAM<=6);
LVRAM=(VRAM<=4)+1/2*(6-VRAM).*(VRAM>4&VRAM<=6);

Hmark=(mark>12000)+(mark>4000&mark<=12000).*(mark-4000)./8000;
Mmark=(mark>4000&mark<=12000).*(12000-mark)./8000+(mark>3000&mark<=4000).*(mark-3000)./1000;
Lmark=(mark<=3000)+1/1000.*(mark>3000&mark<=4000).*(4000-mark);

Hprice=(price>6000)+(price>4000&price<=6000).*(price-4000)./2000;
Mprice=(price>4000&price<=6000).*(6000-price)./2000+(price>1000&price<=4000).*(price-1000)./3000;
Lprice=(price<=1000)+(price>1000&price<=4000).*(4000-price)./3000;
return
```

end

内存隶属度函数求解代码

```
function [HF,MF,LF,HC,MC,LC,Hprice,Mprice,Lprice]=fuzzymem(F,capacity,price)
HF=(F>3200)+(F-2666).*(F>2666&F<=3200)/(3200-2666);
MF=(3200-F).*(F>2666&F<=3200)/(3200-2666)+(F-2400).*(F>2400&F<=2666)/266;
LF=(F<=2400)+(2666-F).*(F>2400&F<=2666)/266;

HC=(capacity>32)+(capacity>16&capacity<=32).*(capacity-16)./16;
MC=(capacity>16&capacity<=32).*(32-capacity)./16+(capacity>8&capacity<=16).*(capacity-8)./8;
LC=(capacity<=8)+1/8.*(capacity>8&capacity<=16).*(16-capacity);

Hprice=(price>1000)+(price>700&price<=1000).*(price-700)./300;
Mprice=(price>700&price<=1000).*(1000-price)./300+(price>300&price<=700).*(price-300)./400;
Lprice=(price<=300)+(price>300&price<=700).*(700-price)./400;
return
end
```

固态硬盘隶属度函数求解代码

```
function [HS,MS,LS,HC,MC,LC,Hprice,Mprice,Lprice]=fuzzySSD(seqr,seqw,capacity,price)
F=(seqr+seqw)/2;
HS=(F>2000)+(F-1000).*(F>1000&F<=2000)/1000;
MS=(2000-F).*(F>1000&F<=2000)/1000+(F-500).*(F>500&F<=1000)/500;
LS=(F<=500)+(1000-F).*(F>500&F<=1000)/500;

HC=(capacity>800)+(capacity>500&capacity<=800).*(capacity-500)./300;
MC=(capacity>500&capacity<=800).*(800-capacity)./300+(capacity>200&capacity<=500).*(capacity-200)./300;
LC=(capacity<=200)+(capacity>200&capacity<=500).*(500-capacity)/300;

Hprice=(price>1000)+(price>700&price<=1000).*(price-700)./300;
Mprice=(price>700&price<=1000).*(1000-price)./300+(price>300&price
```

```

    <=700).*( price -300)./400;
Lprice=(price <=300)+(price >300&price <=700).*(700 - price )./400;
return
end

```

隶属度求解主函数代码

```

for i=1:size(cpu,1)
    Ucpu(i).name=cpu{i,1};
    [Ucpu(i).single(1),Ucpu(i).single(2),Ucpu(i).single(3),Ucpu(i).
        overall(1),Ucpu(i).overall(2),Ucpu(i).overall(3),Ucpu(i).price
        (1),Ucpu(i).price(2),Ucpu(i).price(3)]=fuzzycpu(cpu{i,2},cpu{i
        ,3},cpu{i,4});
end

for i=1:size(gpu,1)
    Ugpu(i).name=gpu{i,1};
    [Ugpu(i).VRAM(1),Ugpu(i).VRAM(2),Ugpu(i).VRAM(3),Ugpu(i).overall(1)
        ,Ugpu(i).overall(2),Ugpu(i).overall(3),Ugpu(i).price(1),Ugpu(i).
        price(2),Ugpu(i).price(3)]=fuzzygpu(gpu{i,2},gpu{i,3},gpu{i,4});
end

for i=1:size(SSD,1)
    USSD(i).name=SSD{i,1};
    [USSD(i).speed(1),USSD(i).speed(2),USSD(i).speed(3),USSD(i).
        capacity(1),USSD(i).capacity(2),USSD(i).capacity(3),USSD(i).
        price(1),USSD(i).price(2),USSD(i).price(3)]=fuzzySSD(SSD{i,3},
        SSD{i,4},SSD{i,2},SSD{i,4});
end

for i=1:size(memory,1)
    Umem(i).name=memory{i,1};
    [Umem(i).frequency(1),Umem(i).frequency(2),Umem(i).frequency(3),
        Umem(i).capacity(1),Umem(i).capacity(2),Umem(i).capacity(3),Umem
        (i).price(1),Umem(i).price(2),Umem(i).price(3)]=fuzzymem(memory{
        i,3},memory{i,2},memory{i,4});
end

```

E 多因素评价矩阵求解代码

CPU 多因素评价矩阵求解代码

```
CPU=zeros(23,3);
w = [0.5,0.3,0.2];
for i = 1:23
    a = [Ucpu(i).single;Ucpu(i).overall;Ucpu(i).price];
    u = zeros(1,3);
    v = zeros(1,3);
    for j = 1:3
        for k = 1:3
            if w(k)<=a(k,j)
                u(k) = w(k);
            else
                u(k) = a(k,j);
            end
        end
        v(j) = max(u);
    end
    for j = 1:3
        CPU(i,j) = v(j)/sum(v);
    end
end
```

GPU 多因素评价矩阵求解代码

```
GPU=zeros(11,3);
w = [0.35,0.4,0.25];
for i = 1:11
    a = [Ugpu(i).VRAM;Ugpu(i).overall;Ugpu(i).price];
    u = zeros(1,3);
    v = zeros(1,3);
    for j = 1:3
        for k = 1:3
            if w(k)<=a(k,j)
                u(k) = w(k);
            else
                u(k) = a(k,j);
            end
        end
    end
end
```

```

        end
        v(j) = max(u);
    end
    for j = 1:3
        GPU(i,j) = v(j)/sum(v);
    end
end

```

内存多因素评价矩阵求解代码

```

MEM=zeros(20,3);
w = [0.45,0.4,0.15];
for i = 1:20
    a = [Umem(i).frequency;Umem(i).capacity;Umem(i).price];
    u = zeros(1,3);
    v = zeros(1,3);
    for j = 1:3
        for k = 1:3
            if w(k)<=a(k,j)
                u(k) = w(k);
            else
                u(k) = a(k,j);
            end
        end
        v(j) = max(u);
    end
    for j = 1:3
        MEM(i,j) = v(j)/sum(v);
    end
end

```

固态硬盘多因素评价矩阵求解代码

```

SSD=zeros(9,3);
w = [0.5,0.4,0.1];
for i = 1:9
    a = [USSD(i).speed;USSD(i).capacity;USSD(i).price];
    u = zeros(1,3);
    v = zeros(1,3);

```

```
for j = 1:3
    for k = 1:3
        if w(k) <= a(k, j)
            u(k) = w(k);
        else
            u(k) = a(k, j);
        end
    end
    v(j) = max(u);
end
for j = 1:3
    SSD(i, j) = v(j)/sum(v);
end
end
```

F 硬件序号与硬件型号对照表

序号	CPU	GPU	内存	固态硬盘
1	i9-10900K	RTX2080Ti	骇客神条 Fury 雷电 2400 频单条 8G	西数黑盘 250G
2	i7-10700K	RTX2080SUPER	骇客神条 Fury 雷电 2400 频单条 16G	西数黑盘 500G
3	i7-10700	RTX2070SUPER	骇客神条 Fury 雷电 2400 频套条 4G*2	西数黑盘 1T
4	i5-10500	RTX2070	骇客神条 Fury 雷电 2400 频套条 8G*2	西数蓝盘 250G
5	i5-10400	RTX2060SUPER	骇客神条 Fury 雷电 2400 频套条 16G*2	西数蓝盘 500G
6	i9-9900K	RTX2060	骇客神条 Fury 雷电 2666 频单条 8G	西数蓝盘 1T
7	i7-9700K	GTX1660Ti	骇客神条 Fury 雷电 2666 频单条 16G	西数绿盘 120G
8	i5-9600K	GTX1660SUPER	骇客神条 Fury 雷电 2666 频套条 8G*2	西数绿盘 250G
9	i7-9700	GTX1660	骇客神条 Fury 雷电 2666 频套条 16G*2	西数绿盘 500G
10	i5-9400	GTX1650SUPER	骇客神条 Fury 雷电 2666 频单条 32G	
11	i3-9100	GTX1650	骇客神条 Fury 雷电 3200 频单条 8G	
12	3990X		骇客神条 Fury 雷电 3200 频单条 16G	
13	3970X		骇客神条 Fury 雷电 3200 频套条 8G*2	
14	3960X		骇客神条 Fury 雷电 3200 频套条 16G*2	
15	3950X		骇客神条 Fury 雷电 3200 频单条 32G	
16	3900X		骇客神条 Fury 雷电 3600 频单条 16G	
17	3800X		骇客神条 Fury 雷电 3600 频套条 8G*2	
18	3700X		骇客神条 Fury 雷电 3600 频套条 16G*2	
19	3600X		骇客神条 Fury 雷电 3733 频套条 8G*2	
20	3600		骇客神条 Fury 雷电 3733 频套条 16G*2	
21	3500X			
22	3300X			
23	3100			