

AI算法学家：AI时代的算法设计

卜东波

中科院计算所

2024年3月8日

缘起：人求解问题的灵感哪里来？

2001年的困惑：这个算法是如何想出来的？

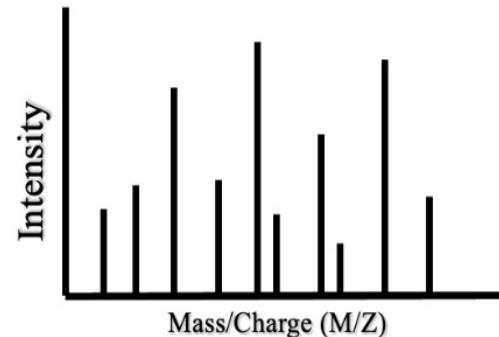
质谱鉴定问题

蛋白质序列

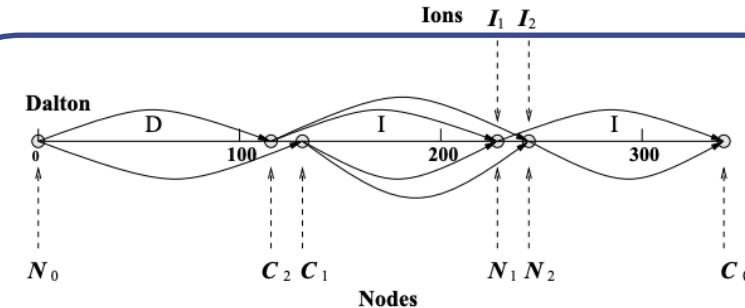
AHCGILSTVW

碎裂
鉴定

质谱



双端动态规划算法

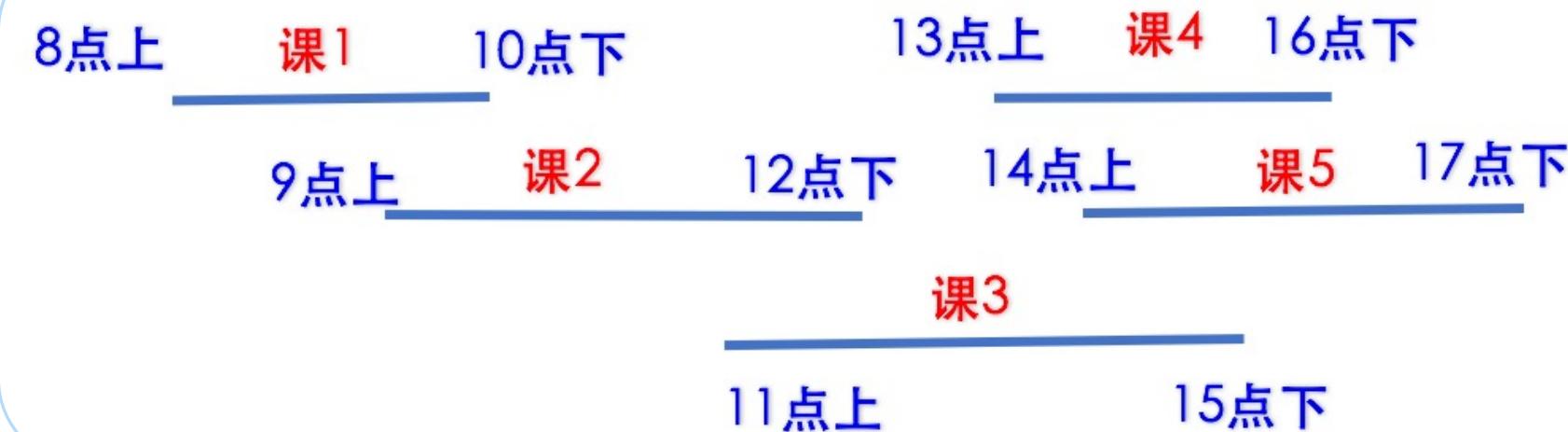


Algorithm Compute-N(G)

1. Initialize $N(i, j) = 0$ for all i and j ;
2. Compute $N(k, k - 1)$ and $N(k - 1, k)$;
3. For $j = k - 2$ to 0
4. For $i = k$ to $j + 2$
 - (a) if $N(i, j + 1) = 1$ and $E(x_j, x_i) = 1$, then $N(j, j + 1) = 1$;
 - (b) if $N(i, j + 1) = 1$ and $E(y_{j+1}, y_j) = 1$, then $N(i, j) = 1$;
 - (c) if $N(j + 1, i) = 1$ and $E(x_j, x_{j+1}) = 1$, then $N(j, i) = 1$;
 - (d) if $N(j + 1, i) = 1$ and $E(y_i, y_{j+1}) = 1$, then $N(j + 1, j) = 1$

沮丧：别人说出来我懂，不说我就想不出来

一个简洁的例子：排课问题



- 有5门课申请使用报告厅，已知上下课时间
- 问：最多能安排几门课？

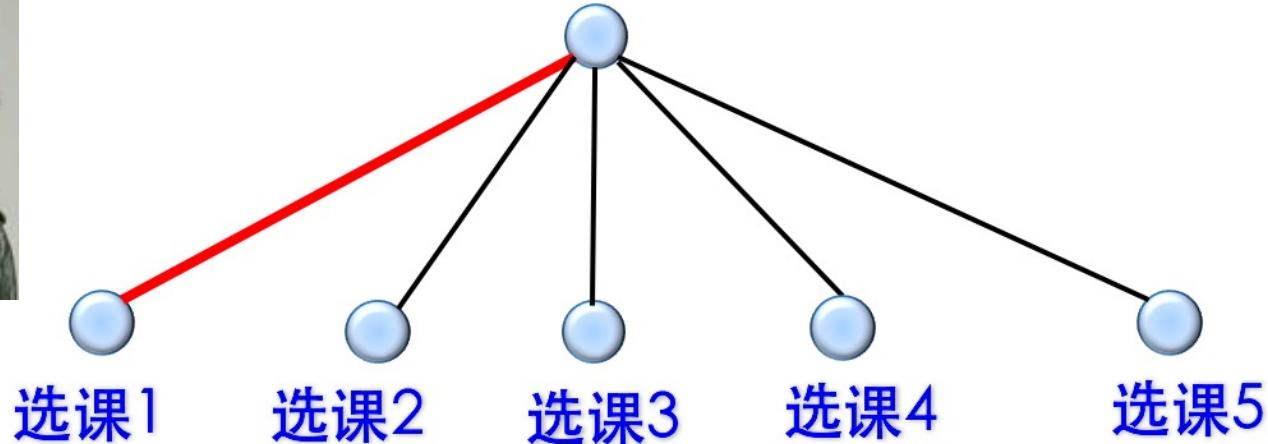
经典的贪心算法：选最早下课的！



GREEDY INTERVAL SCHEDULING(*CourseSet*)

- 1: **while** *CourseSet* $\neq \emptyset$ **do**
- 2: Select the course *C* with **earliest finishing time**;
- 3: Remove *C* and related courses from *CourseSet*;
- 4: **end while**

规则简单，想出来却不容易！



- 规则1：优先安排最早上课的； X
- 规则2：优先安排时长最短的； X
- 规则3：优先安排冲突最少的； X
- 规则4：优先安排最早下课的； ✓
- 规则5：优先安排最晚上课的； ✓

那个聪明人到底是怎样想出来的呢？

波利亚中学时有同样的困惑

“是的，这个解答看来是可行的，也是正确的，
但别人是怎样发现这个事实的呢？”

我怎样干才能想到这样一个解答呢？”

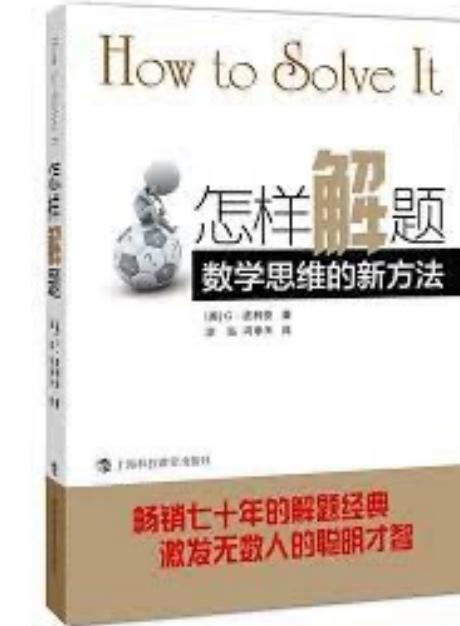
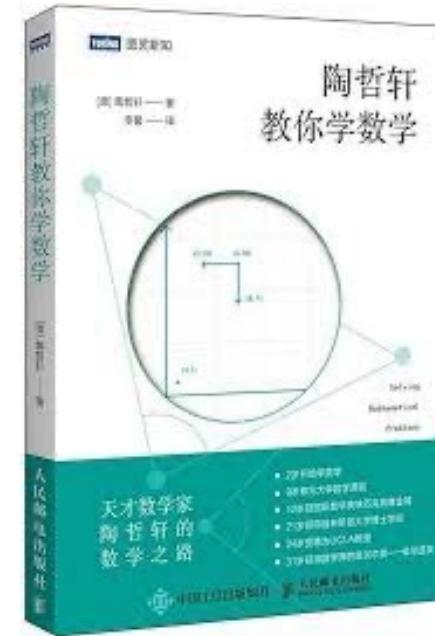
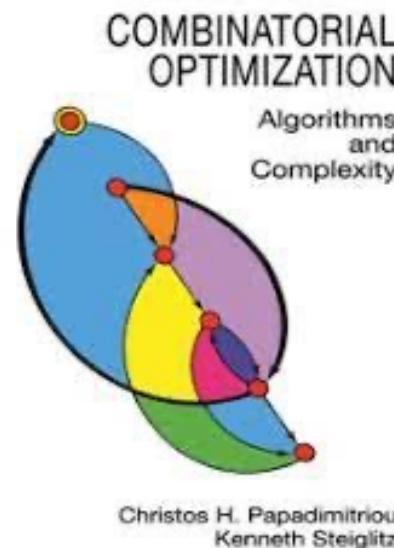
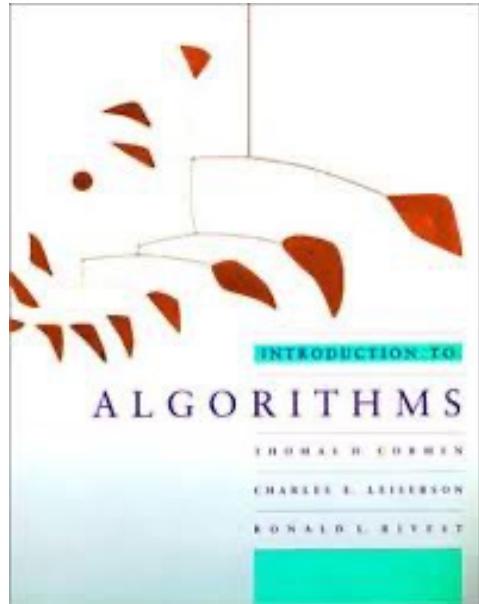


G. Polya (1887-1985)

阶段一：一棵问题求解思路树

(2001-2008)

摸索之一：读书、读传记、读回忆录



算法导论

组合最优化

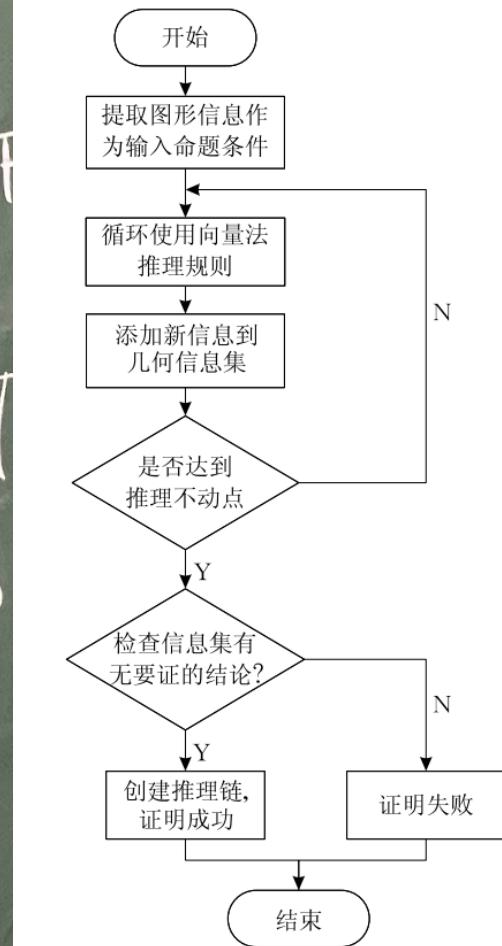
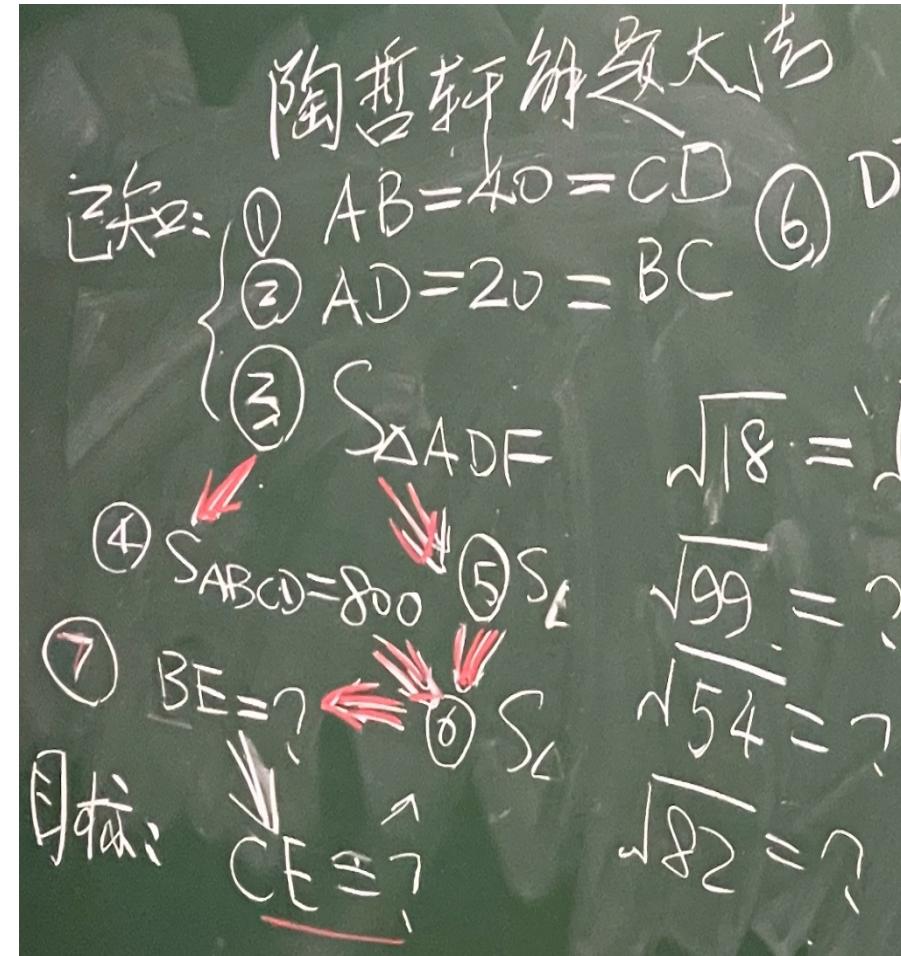
陶哲轩

怎样解题?

几何定理机器证明

怎样解题？从已知到目标的启发式搜索！

弄清问题	已知是什么？未知是什么？
	条件是什么？结论是什么？
	画个草图，引入适当的符号
	见过这道题或与之类似的题吗？
	能联想起有关的定理或公式吗？
	再看看未知条件！
	换一个方式来叙述这道题。
拟定计划	回到定义看看！！
	先解决一个特例试试。
	这个问题的一般形式是什么？
	你能解决问题的一部分吗？
	你用了全部条件吗？
实行计划	实现你的解题计划并检验每一步。
	证明你的每一步都是正确的。
	检查结果并检验其正确性。
回顾	换一个方法做做这道题。
	尝试把你的结果和方法用到其他问题上去。



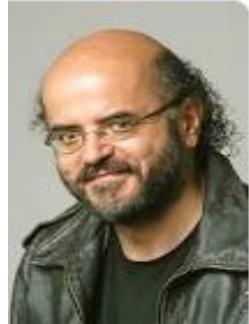
波利亚解题法

陶哲轩&张国强解题法

张景中证明法

摸索之二：访谈大家，“碰到问题，你如何下手？”

R. Karp, C. Papadimitriou, 李明, 姜涛, 王鲁生.....

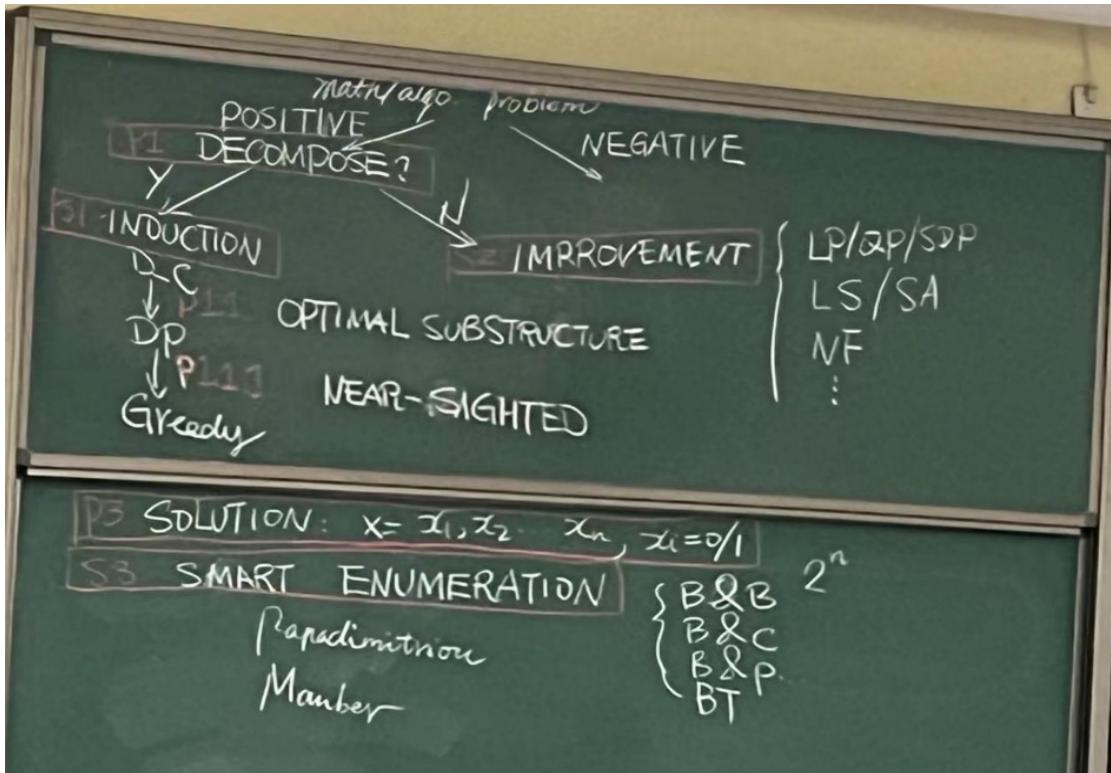


- 为证明一个问题是否是NP完全的，我们首先观察这个问题的**简单实例**，直到我们找到一些实例，能够展现出有趣的行为（问题结构）

----- C. Papadimitriou

观察问题的组合结构；有什么样的结构，设计什么样的算法

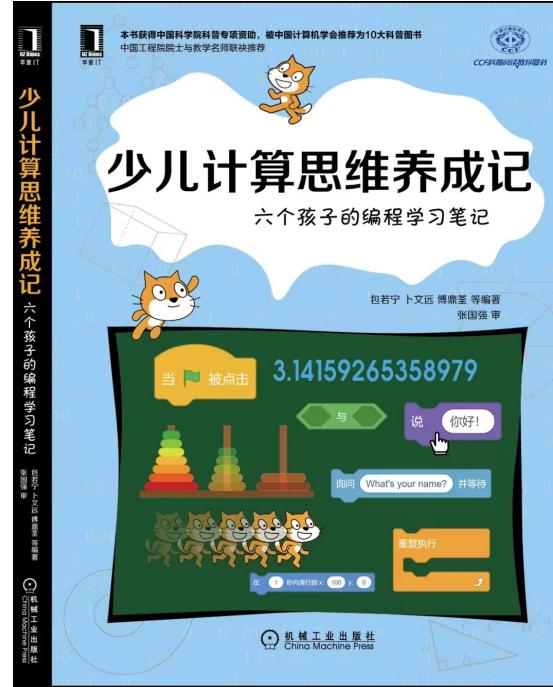
2008年的领悟：一棵问题解决思路树



6. 这个问题是最优化问题吗？如果是，咱们先理解一下问题：写出一个最简单实例，枚举出所有的可行解，对每个可行解算出目标函数值，进而找出最优解。（如果是，则转至10）
 - a. 这个问题能分解成子问题吗？咱们尝试把问题的求解过程描述成“多步决策过程”，即：在每个决策步骤，从多个选择项中选择一个，从而得到解的一个组成部分。（如果能，则转至11）
 - b. 这个问题有“最优子结构”性质吗？即：能否把子问题的最优解组合起来，得到原问题的最优解？你可以用递归表达式来表示“最优子结构”性质（如果能，则转至12）
 - c. 这个问题的子问题之间是独立的，还是有重叠的？如果是重叠的，则需要避免多次重复求解一个子问题，具体手段是“以存代算”，即：设计一个数据结构（通常是表格），把已经求解的子问题的解记录下来；求解一个子问题之前，先查询表格，看是否已经求解；如果已经求解过，则不需要重复求解。

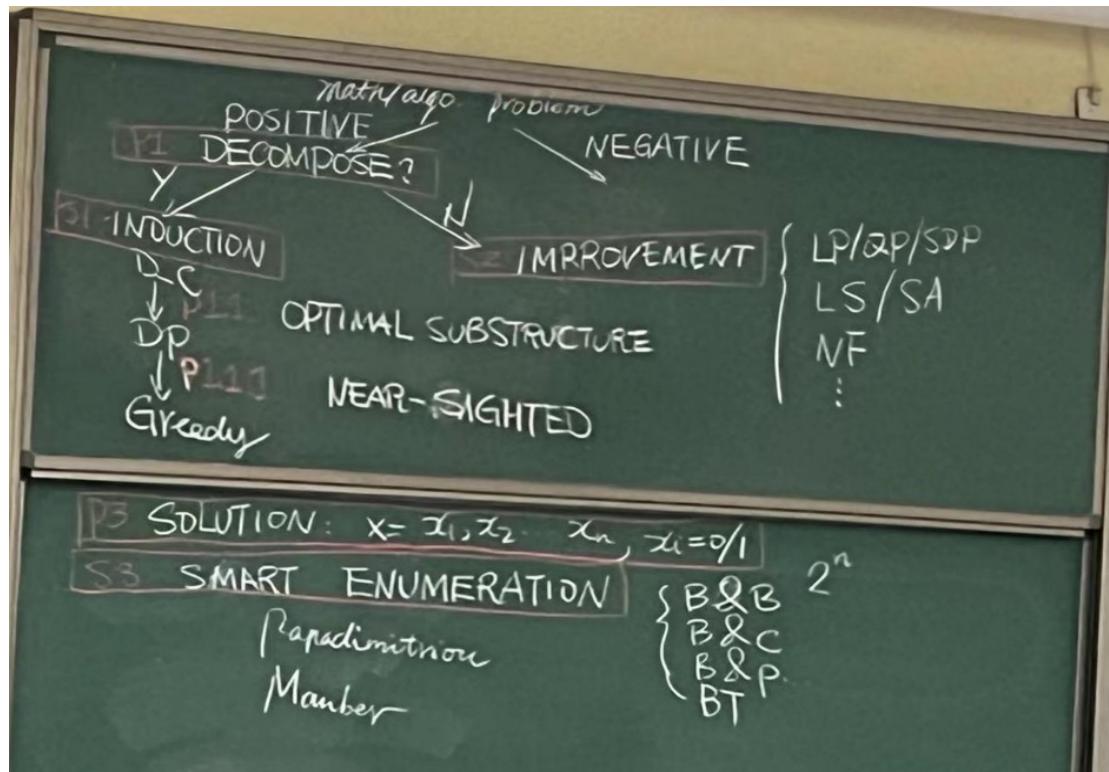
- **19个prompts：**自问自答，理性探索
- **设计过程：**“观察-尝试-修正”循环；观察到什么样的结构，设计什么样的算法
- **组合结构：**问题的可分解性、解之间的变换关系、解的形式与产生过程

另一个验证：少儿计算思维养成的引导、观察



结论：对研究生有效的计算思维，裁剪后，小学生也适用！

优点与不足



R. Karp随身带一个小本子，碰到问题，手工画几个最简单的例子，观察规律，寻找灵感.....

优点：理性的探索

不足：“人”执行探索

阶段二：AIA计划---用NN学出“人的灵感”

(2015-2023)

AIA: AI algorithmist

教育的桎梏：对理性建模的追求、对神经网络的挑剔



“....神经网路不过是美国人西部牛仔精神
在CS领域的大爆发...”

Esdger Dijkstra



“...NN超出我们大脑的capacity... ”

Jun Smith Liu

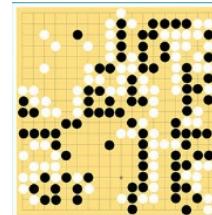
AlphaGo纠正我的观念：dirty的事情就让dirty tools干！

- 啥是dirty的事情？说不清楚的事情就是dirty！

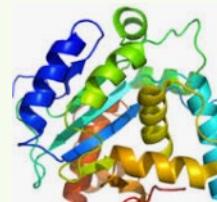
- 视觉机理



- 下棋如何落子



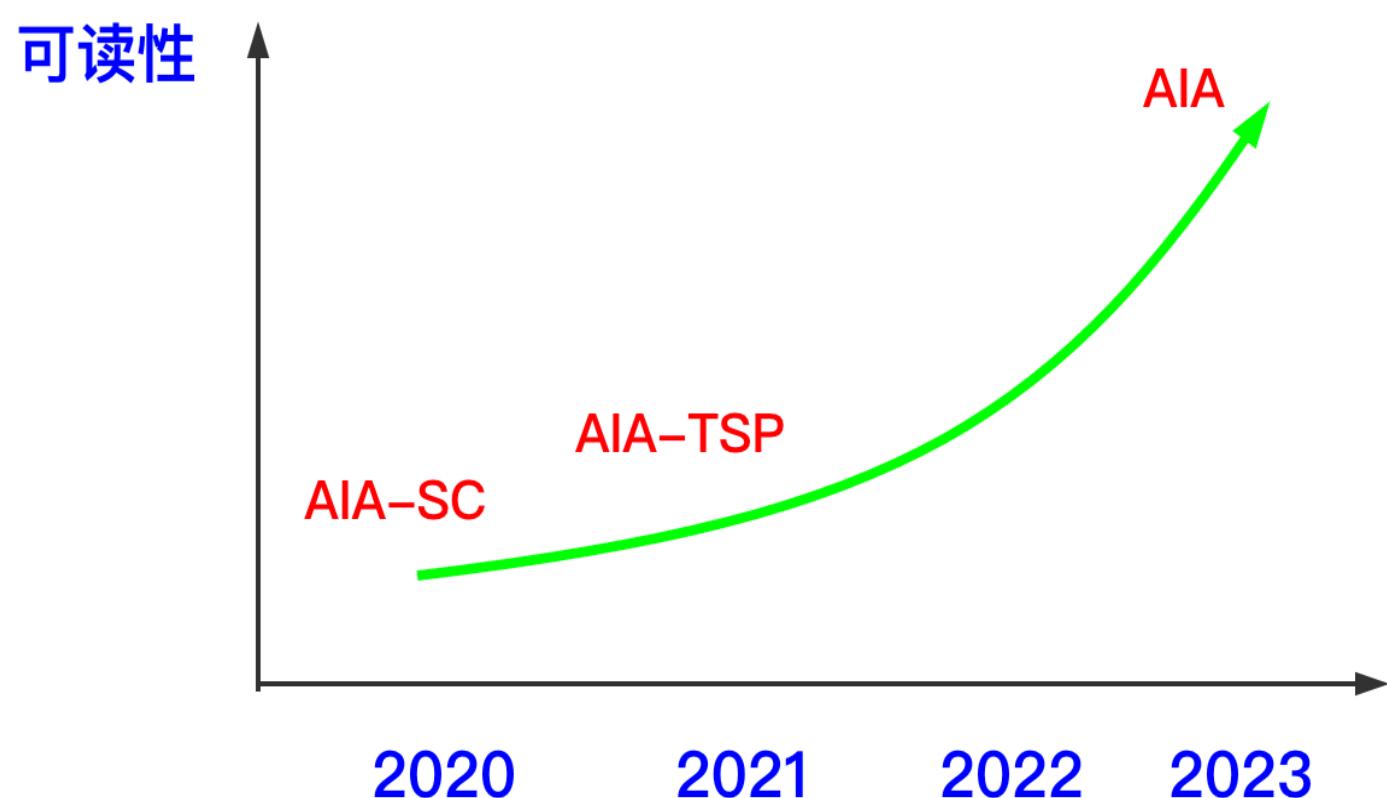
- 蛋白质折叠机理



- 算法设计的灵感

本页写于2015年底，两项预判皆已应验：
--- AlphaFold, Senior2018
--- 数学猜想, Davies2021

AIA计划：凡是人的灵感，都用NN+大数据替换

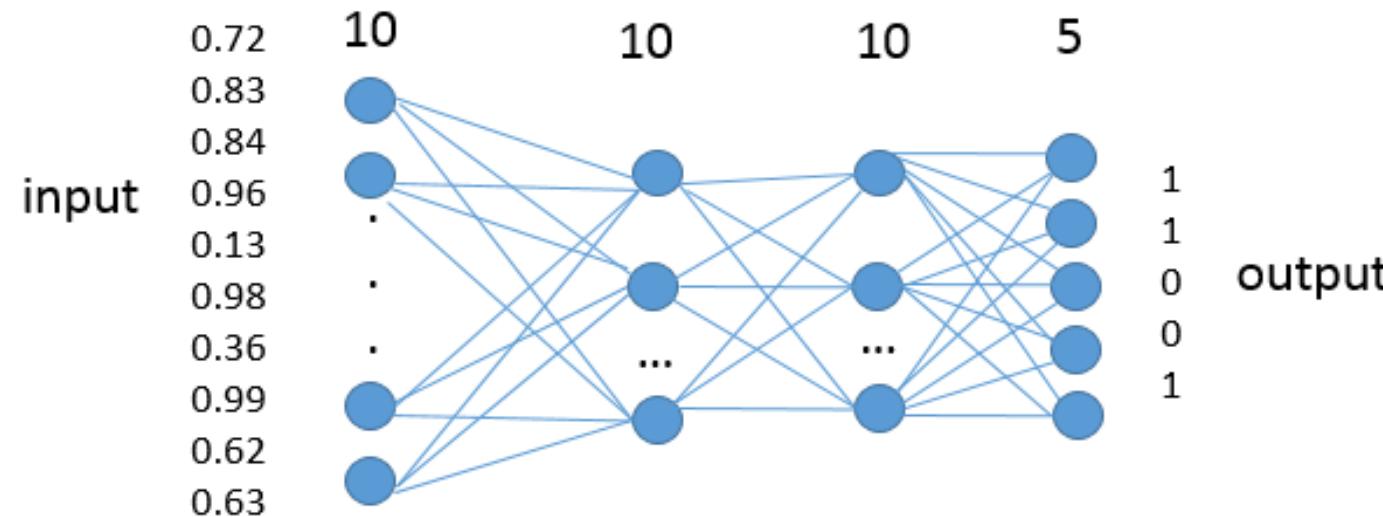


人的灵感

- 排课问题：**选哪门课？**
- 集合覆盖：**选哪个集合？**
- 插值问题：**基函数选啥？**
- 旅行商问题：**选哪些边交换？**
- 整数规划：**扩展哪个子问题？**

进展1：让NN自己学出排课算法中“聪明人的灵感”

- **实验设计：**
 - 随机产生10万个5门课的例子，
 - 先用笨拙的动态规划算法跑出标准答案
 - 然后用NN来学

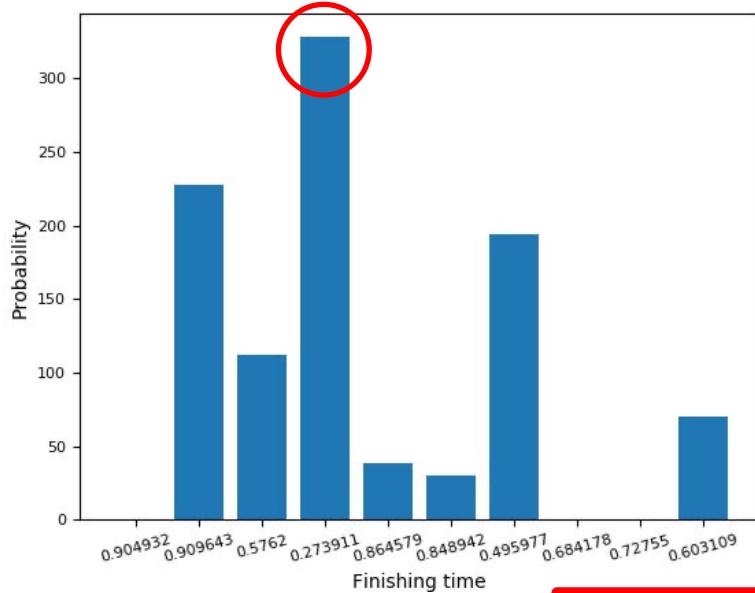


- **正确率：98%！**

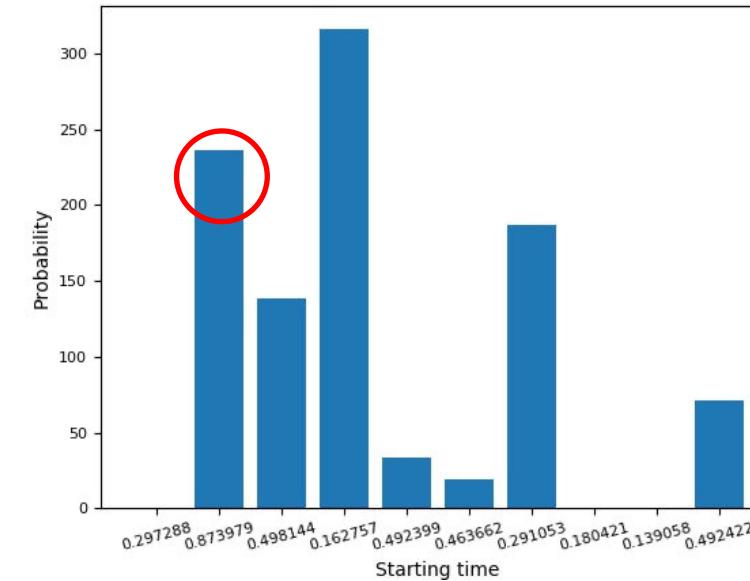
扩展一步：用小网络指导大实例

- 如何解10门课的例子？
- Bagging! 随机选5门课，跑小NN，投票！
- 正确率：95%！

最高票是下课最早的，即规则4



第二名是上课最晚的，即规则5

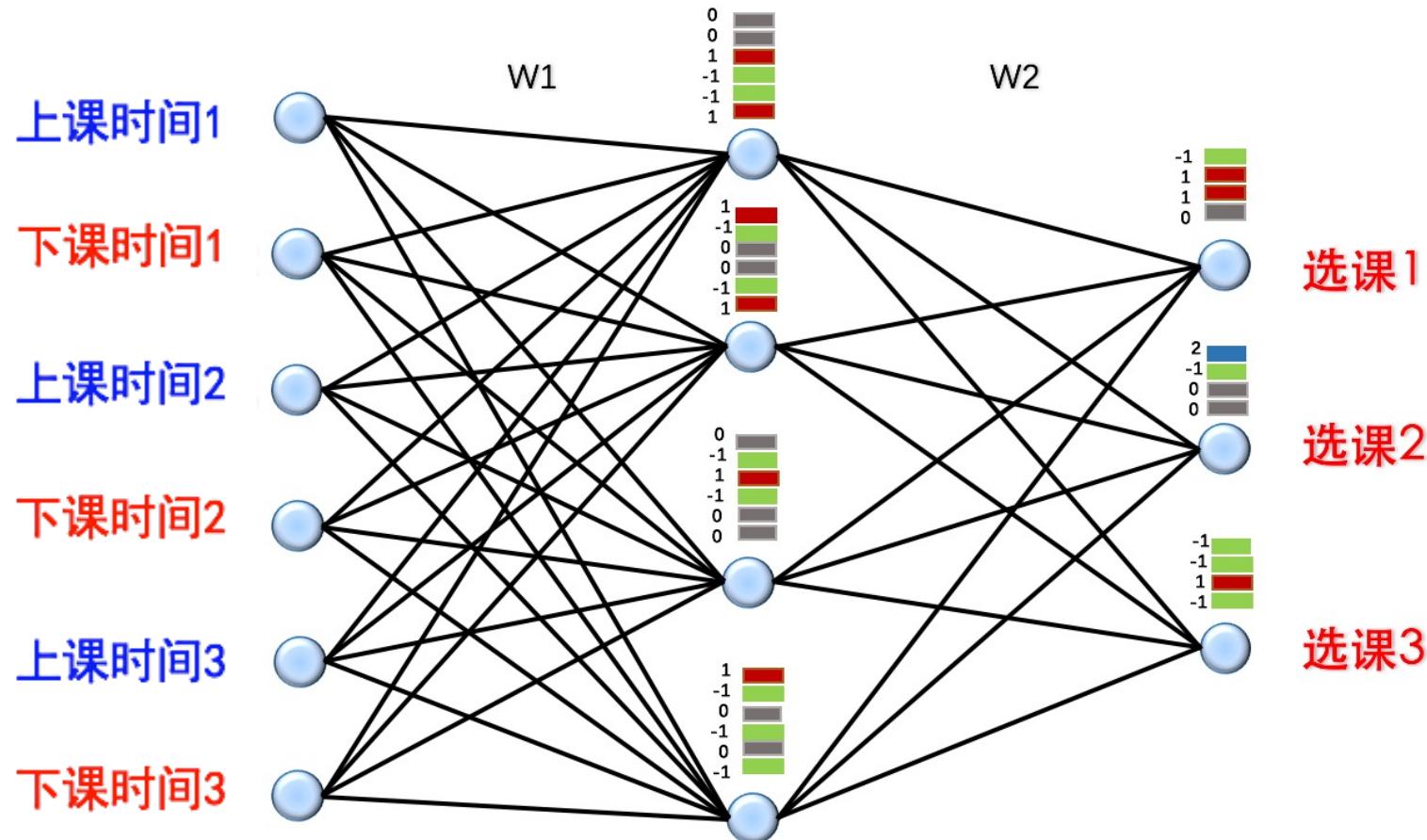


NN学出了规则4和规则5！



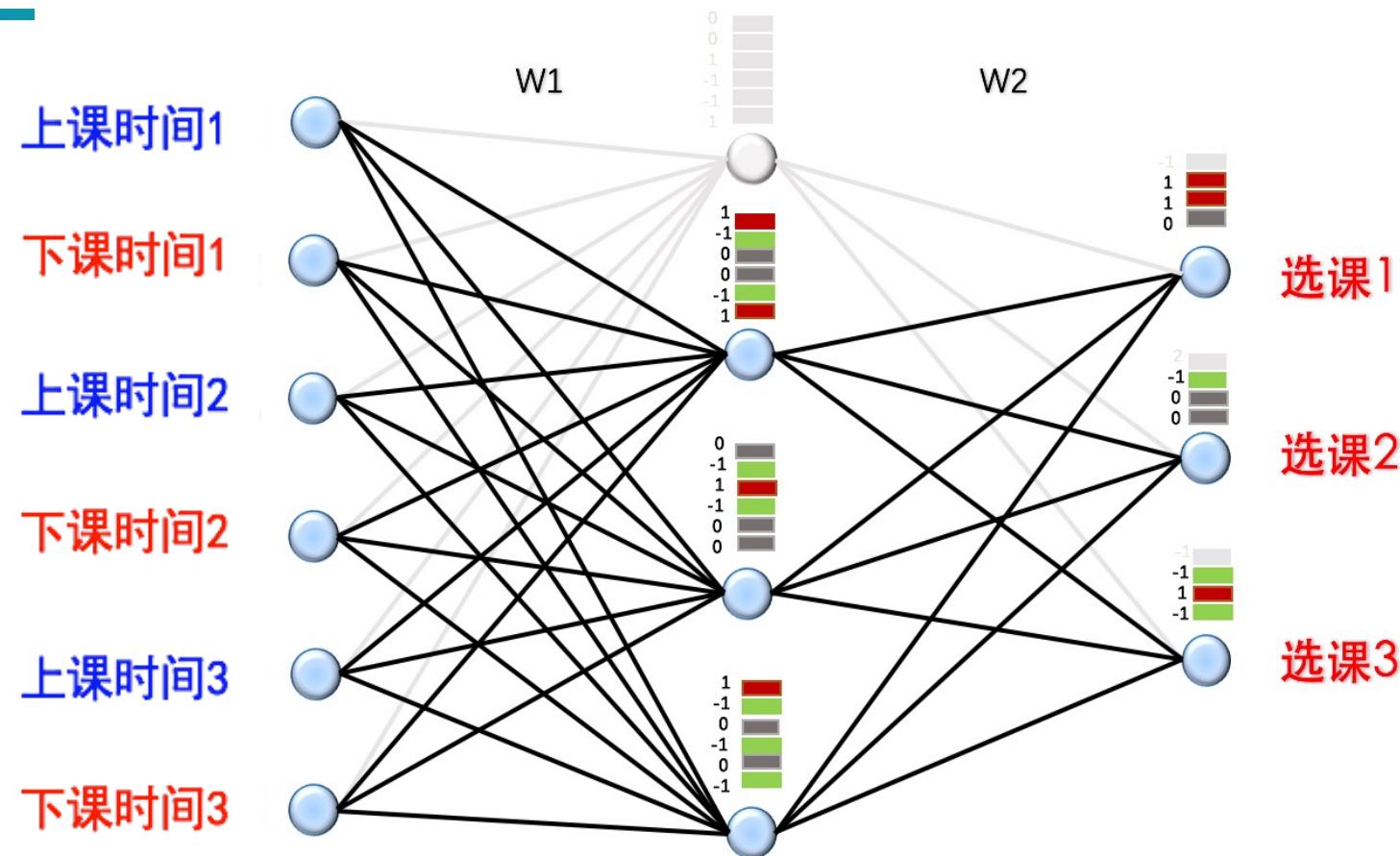
隋京言

NN如何同时掌握贪心规则4和规则5？



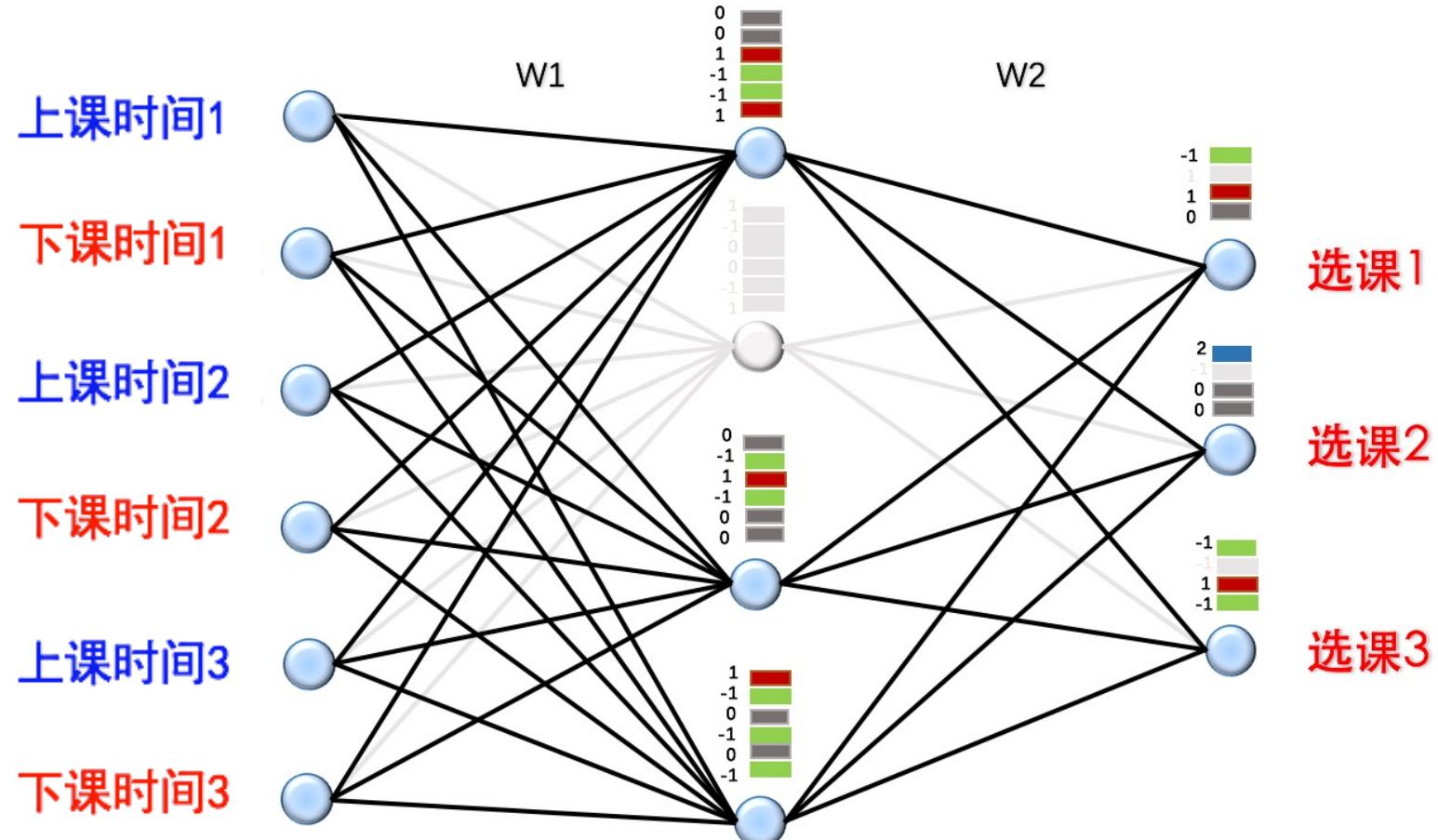
打开黑箱：这是两个网络的复合！

隐层结点2, 3, 4管规则4



去掉结点1，一个会贪心规则4的网络
选最早下课的

隐层结点1，3，4管规则5



去掉结点2，一个会贪心规则5的网络
选最晚上课的

Human设计的贪心规则 vs AI设计的贪心规则

Human algorithmist:

GREEDYINTERVALSCHEDULING($CourseSet$)

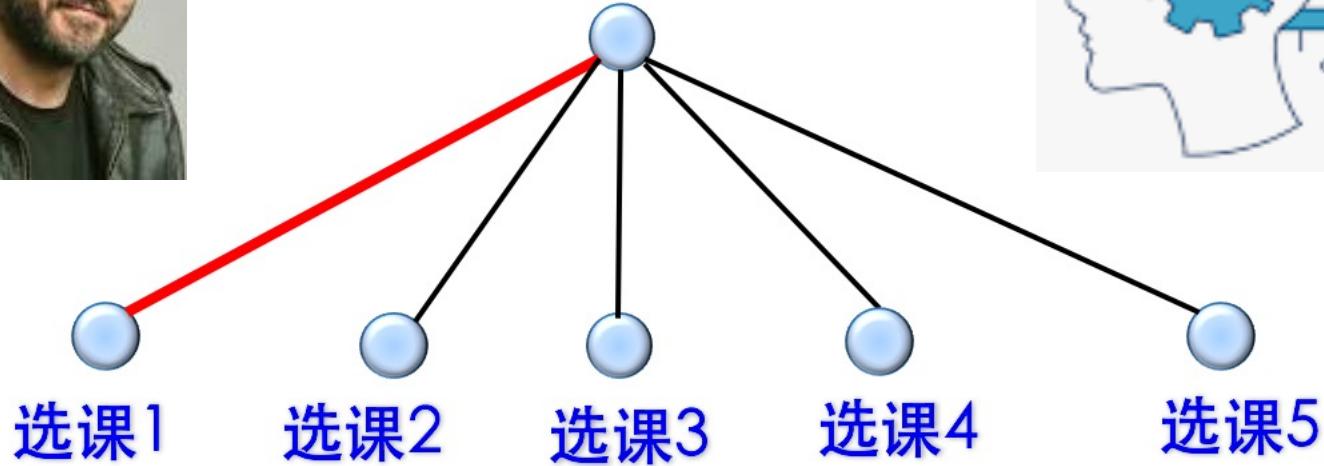
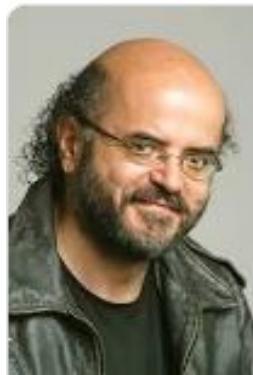
- 1: **while** $CourseSet \neq \emptyset$ **do**
- 2: Select the course C with **earliest finishing time**;
- 3: Remove C and related courses from $CourseSet$;
- 4: **end while**

AI algorithmist:

NNINTERVALSCHEDULING($CourseSet$)

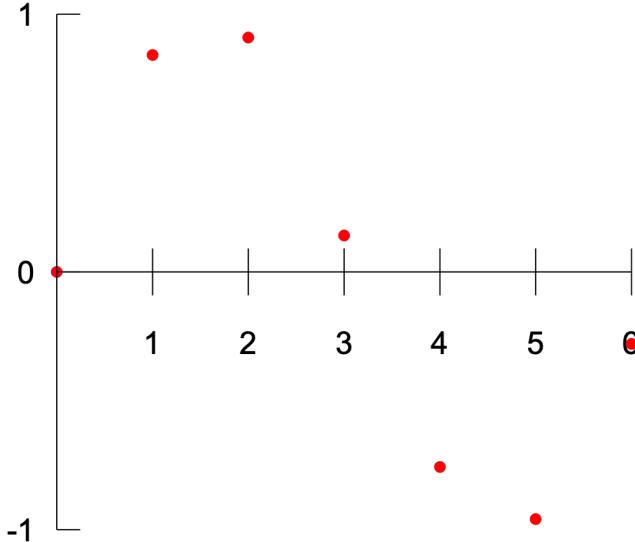
- 1: **while** $CourseSet \neq \emptyset$ **do**
- 2: Select the course C with **highest score by NN**($CourseSet$);
- 3: Remove C and related courses from $CourseSet$;
- 4: **end while**

Human Algorithmist = AI Algorithmist?

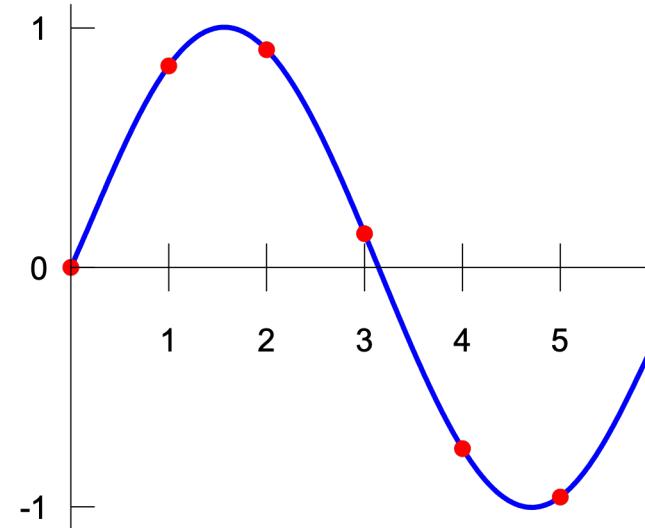


Yes! (在搜索问题上)

进展2：散点插值新算法---NIERT

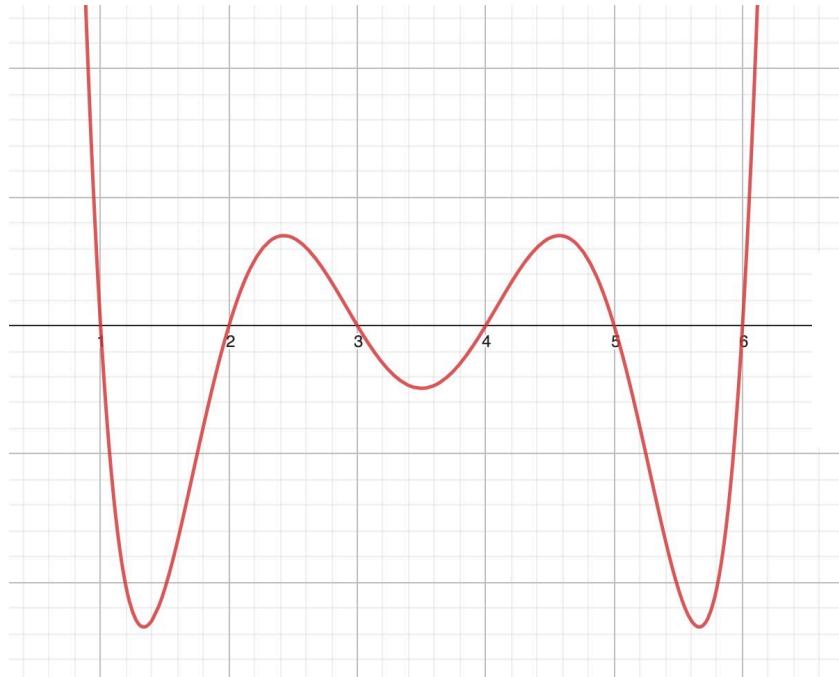


散点插值



- 教科书：拉格朗日多项式 $p(x) = f(0)l_0(x) + f(1)l_1(x) + \dots + f(6)l_6(x)$
- 关键点：基函数咋选？
$$l_0(x) = \frac{x-1}{0-1} \times \frac{x-2}{0-2} \times \dots \times \frac{x-6}{0-6}$$
$$l_1(x) = \frac{x-0}{1-0} \times \frac{x-2}{1-2} \times \dots \times \frac{x-6}{1-6}$$

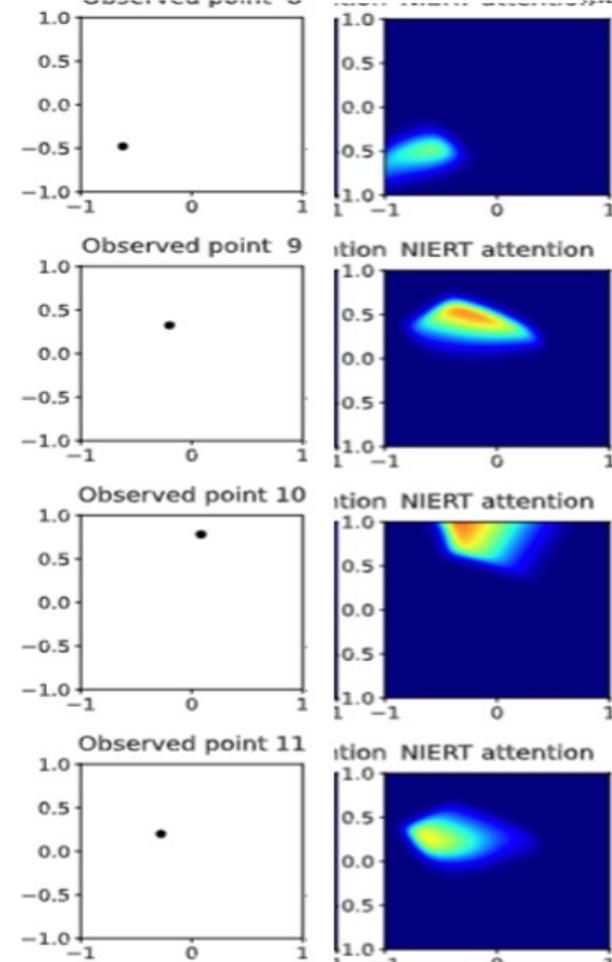
疑问：为何Waring, Euler和Lagrange选这个基函数？



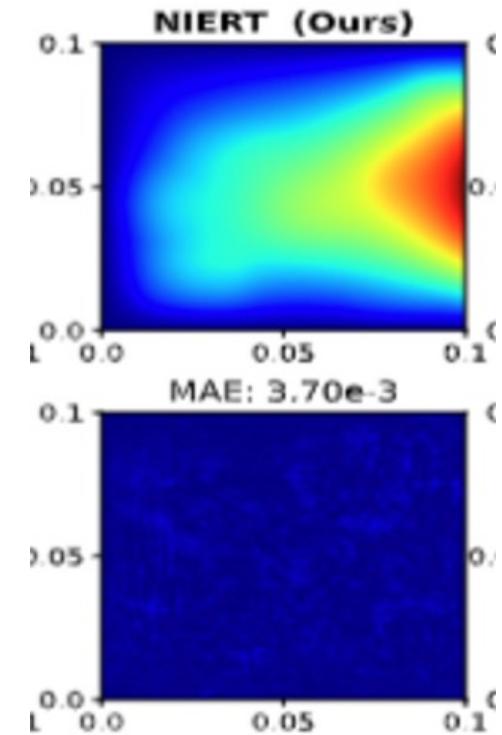
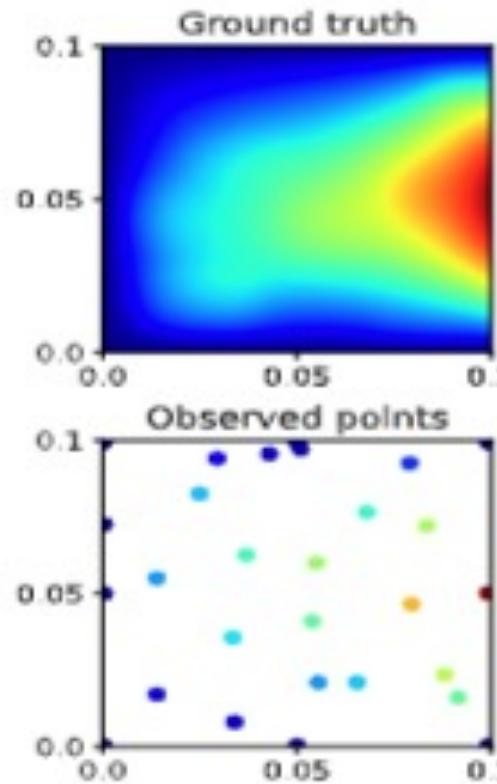
$$l_0(x) = \frac{x-1}{0-1} \times \frac{x-2}{0-2} \times \dots \times \frac{x-6}{0-6}$$

- 对称的基函数就是最好的吗？
- 有没有更好的基函数？

NIERT的诀窍：单方向的attention



学出来的基函数



当前最准的算法



丁诗哲

进展3：整数线性规划，选哪个结点进行分枝？

$$\min \quad x + 2y + 6z$$

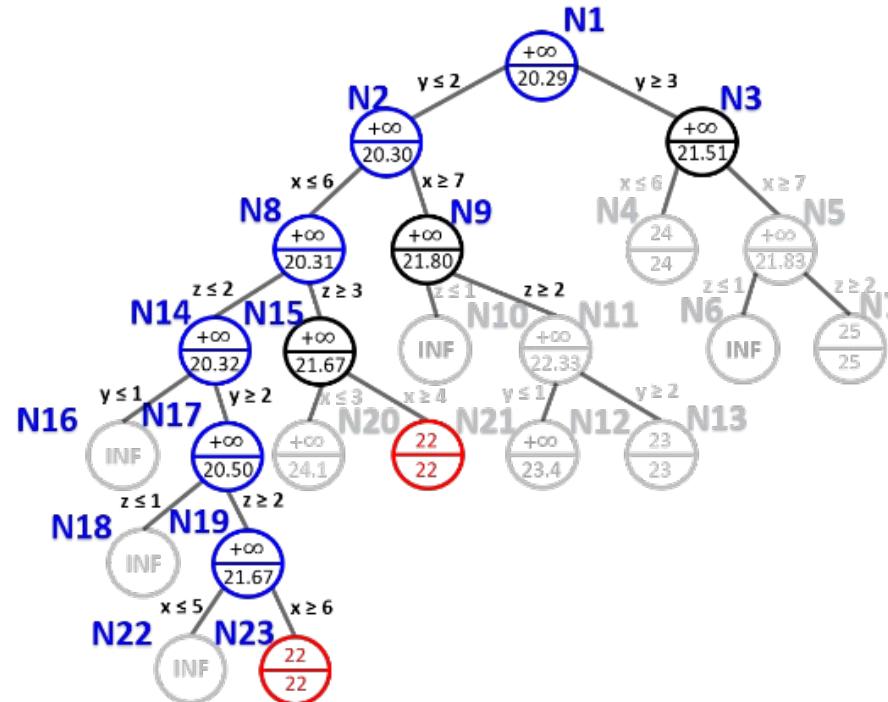
$$s.t. \quad 3x - y + 4z \geq 23$$

$$-x + 6y + 10z \geq 23$$

$$-x - 2y + 53z \geq 65$$

$$x, y, z \geq 0$$

$$x, y, z \in \mathbb{Z}$$

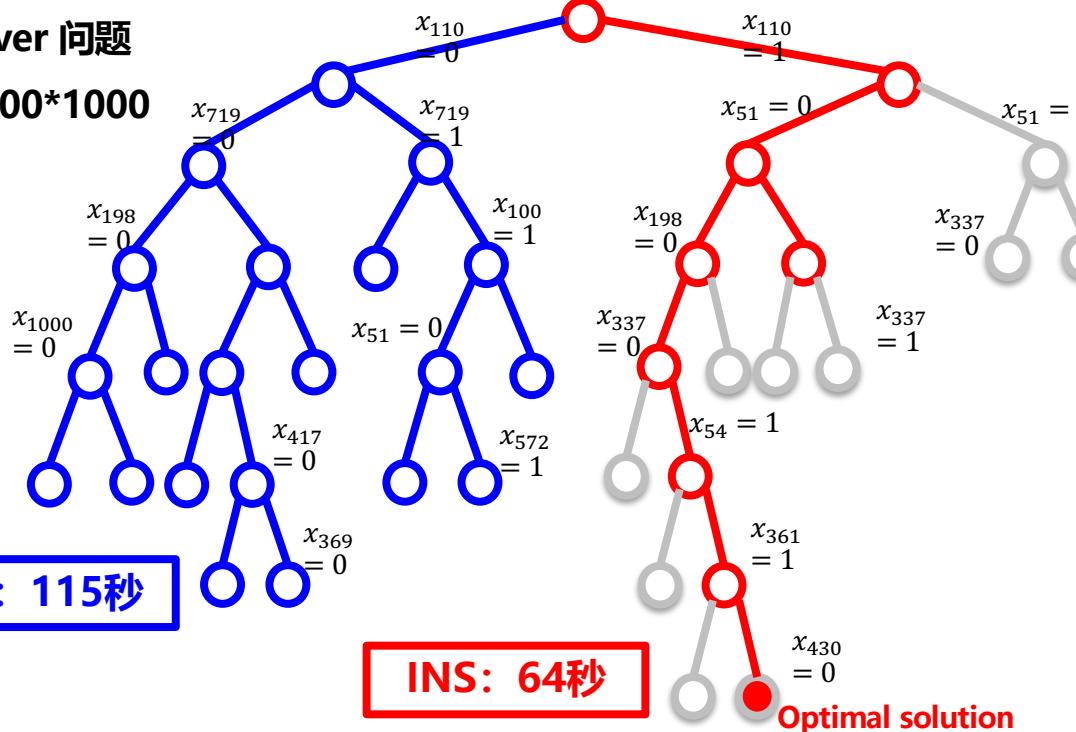


- 聪明人的灵感：选线性松弛解最松的结点（BFS，为当前最好的求解器SCIP采用）
- 不足：选不准，求解慢

AIA-ILP算法：用AI指导选分支节点

Setcover 问题

规模 1000*1000



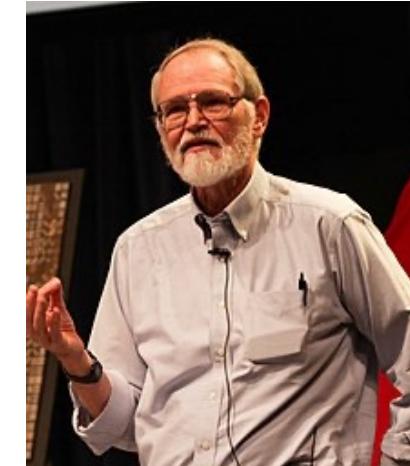
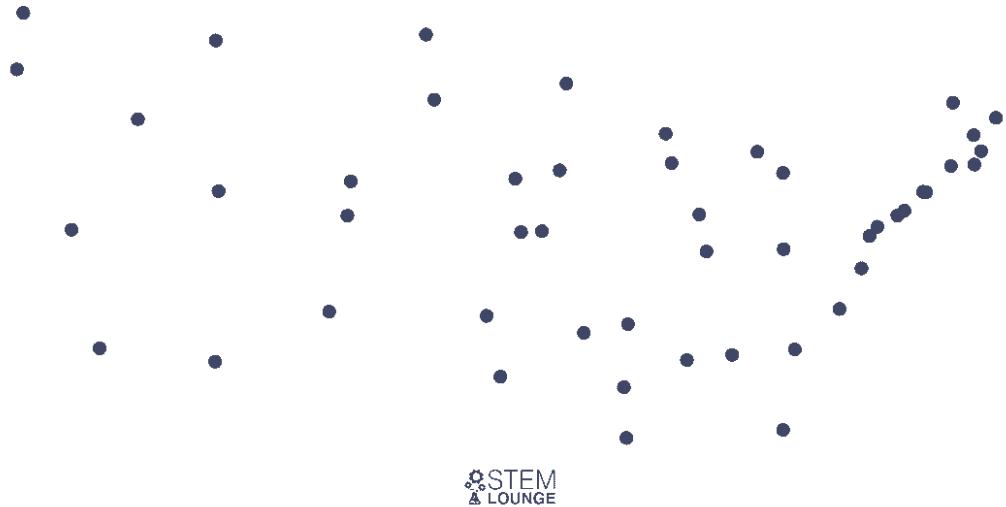
Method	Easy			Medium			Hard	
	Nodes	Bpb-time	Sol-time	Nodes	Bpb-time	Sol-time	Nodes	Wins
SCIP-BFS	22	14.69	21.83	416	216.12	423.41	496	2 / 100
XGBoost	24	14.75	21.36	395	147.37	383.46	451	6 / 100
He et al.[20]	24	15.47	21.98	399	158.29	398.71	483	4 / 100
MLP-Pair	25	15.65	21.21	380	141.77	382.35	424	26 / 100
INS (Ours)	24	15.58	21.04	382	140.32	376.08	435	20 / 100
INS-En (Ours)	23	14.41	21.23	370	118.94	371.39	420	42 / 100
1000×500			1000×1000			1000×2000		
Set Covering								
SCIP-BFS	12	5.77	6.39	93	76.61	103.07	426	5 / 100
XGBoost	13	5.84	7.12	108	78.20	106.44	451	6 / 100
He et al.[20]	13	5.64	6.90	98	69.04	99.60	418	7 / 100
MLP-Pair	13	5.56	7.11	81	59.84	88.29	296	30 / 100
INS (Ours)	13	5.93	7.05	83	58.96	88.27	310	24 / 100
INS-En (Ours)	13	5.42	6.67	82	59.16	88.52	310	28 / 100
100×500			200×1000			300×1000		
Combinatorial Auction								
SCIP-BFS	59	16.15	40.34	77	59.94	172.61	79	18 / 100
XGBoost	63	14.69	39.48	81	53.92	164.97	90	6 / 100
He et al.[20]	64	15.21	40.13	83	57.19	169.33	92	5 / 100
MLP-Pair	57	13.91	36.79	71	56.10	162.35	85	24 / 100
INS (Ours)	55	13.95	36.56	70	58.70	172.05	71	23 / 100
INS-En (Ours)	58	14.78	38.10	67	52.24	157.56	71	24 / 100
100×100			200×100			400×100		
Capacitated Facility Location								

- 实验结果：大数据+NN解组合优化，速度更快，解更优



刘睿之

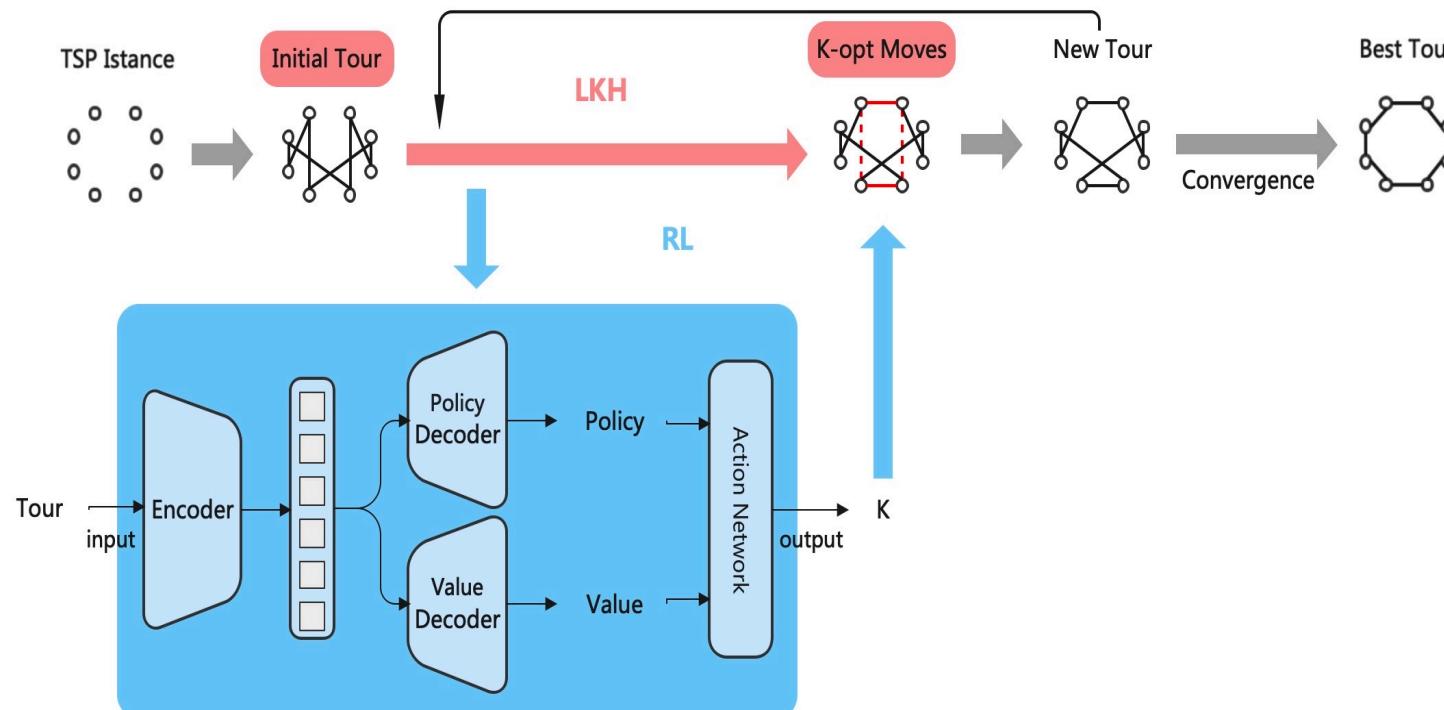
进展4：旅行商问题，去哪些边，加哪些边？



Brian Wilson Kernighan

- LKH三个聪明人的灵感：2条边、3条边、4条边…，逐个试探，见好就换

AIA-TSP：用AI决定换几条边，不用试



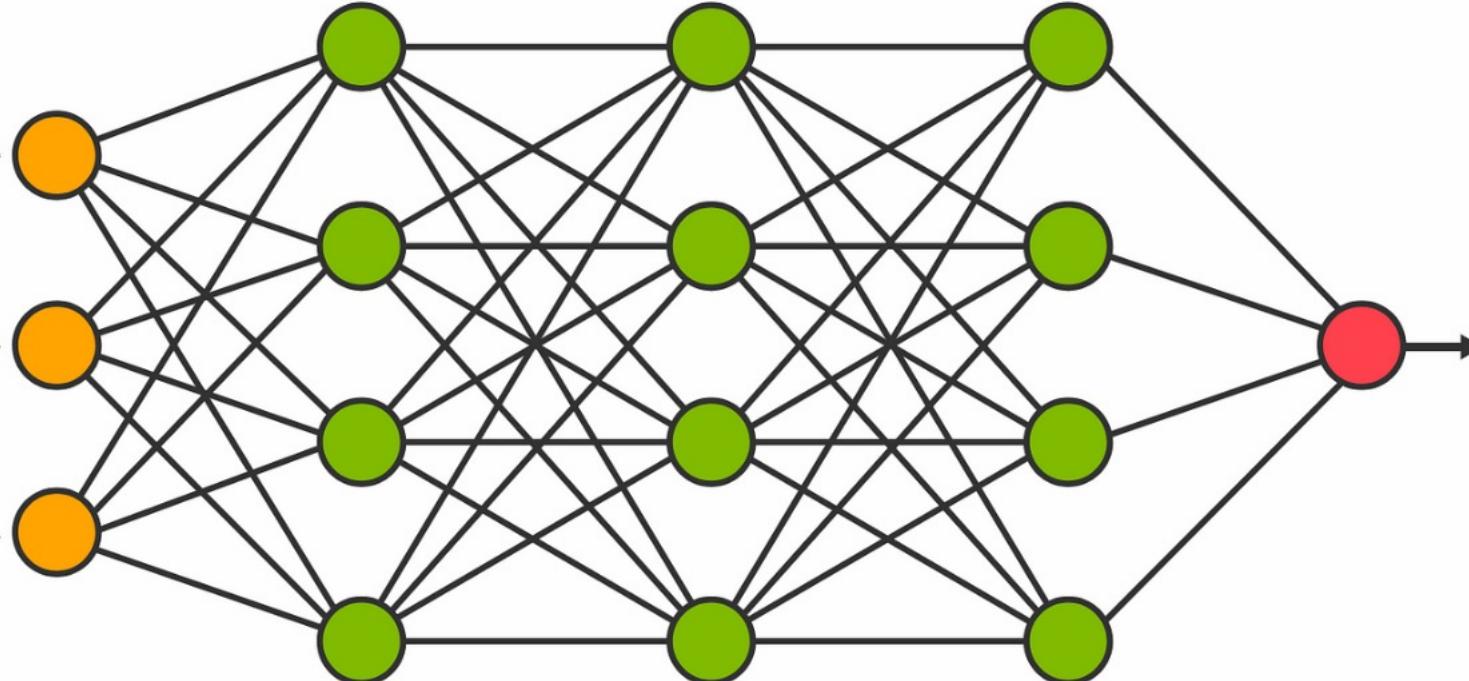
Method	tsp1000		tsp2000	
	Time(s)	Gap(%)	Time(s)	Gap(%)
LKH(10 trial)		0.0532		0.0668
NeuroLKH	633	0.0091	1302	0.0073
RL & LKH		0.0394		0.0592
RL & nLKH		0.0193		0.0306
LKH(100 trial)		0.0204		0.0252
NeuroLKH	932	0.0061	2123	0.0052
RL & LKH		0.0184		0.0329
RL & nLKH		0.0023		0.0055
LKH(1000 trial)		0.0029		0.0047
NeuroLKH	3638	0.0008	7540	0.0018
RL & LKH		0.0031		0.0092
RL & nLKH		0.0000		0.0000

- 优势：速度更快，质量更优，规模大的问题更显著



王睿

优点与不足



优点：NN可有效学出灵感

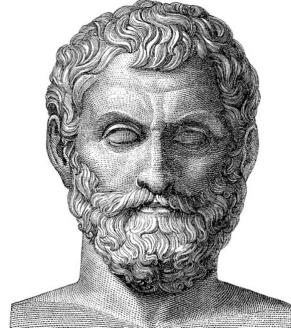
不足：NN很大，部署易受限
可读性差，人难以改进

阶段三：CoPA计划：用LLM写出算法

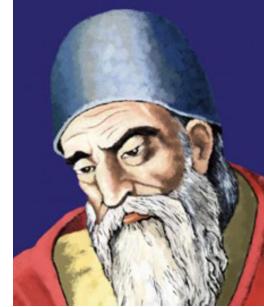
(2023-?)

CoPA: Co-pilot for algorithm

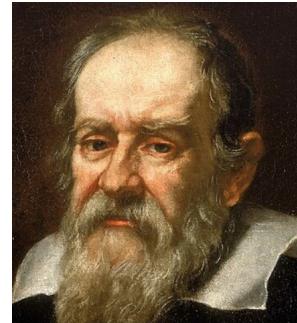
大语言模型引发研究范式嬗变：AI4R



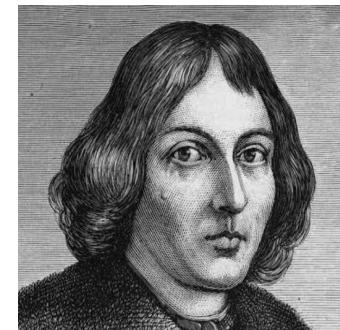
泰勒斯



欧几里得



伽利略



开普勒

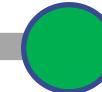


LLM



观察、思辨

演绎



实验、归纳



大数据、计算



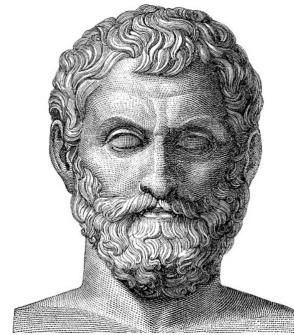
AI4R

---朝永振一郎《物理是什么？》

---纽特《世界的种子》

---李国杰《可》

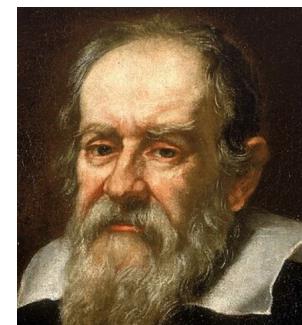
AI4R：人机协同，产生“猜想”



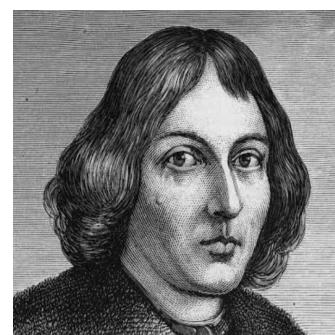
泰勒斯



欧几里得



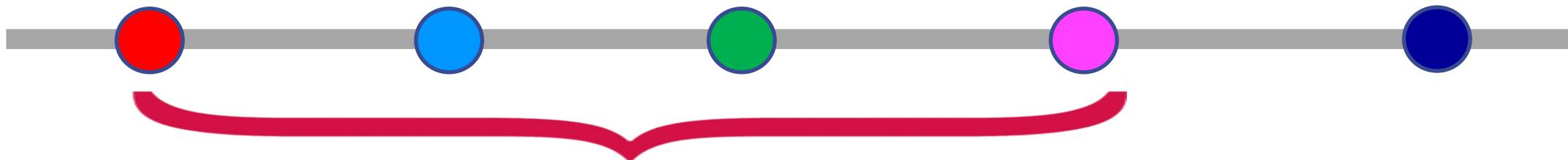
伽利略



开普勒



LLM

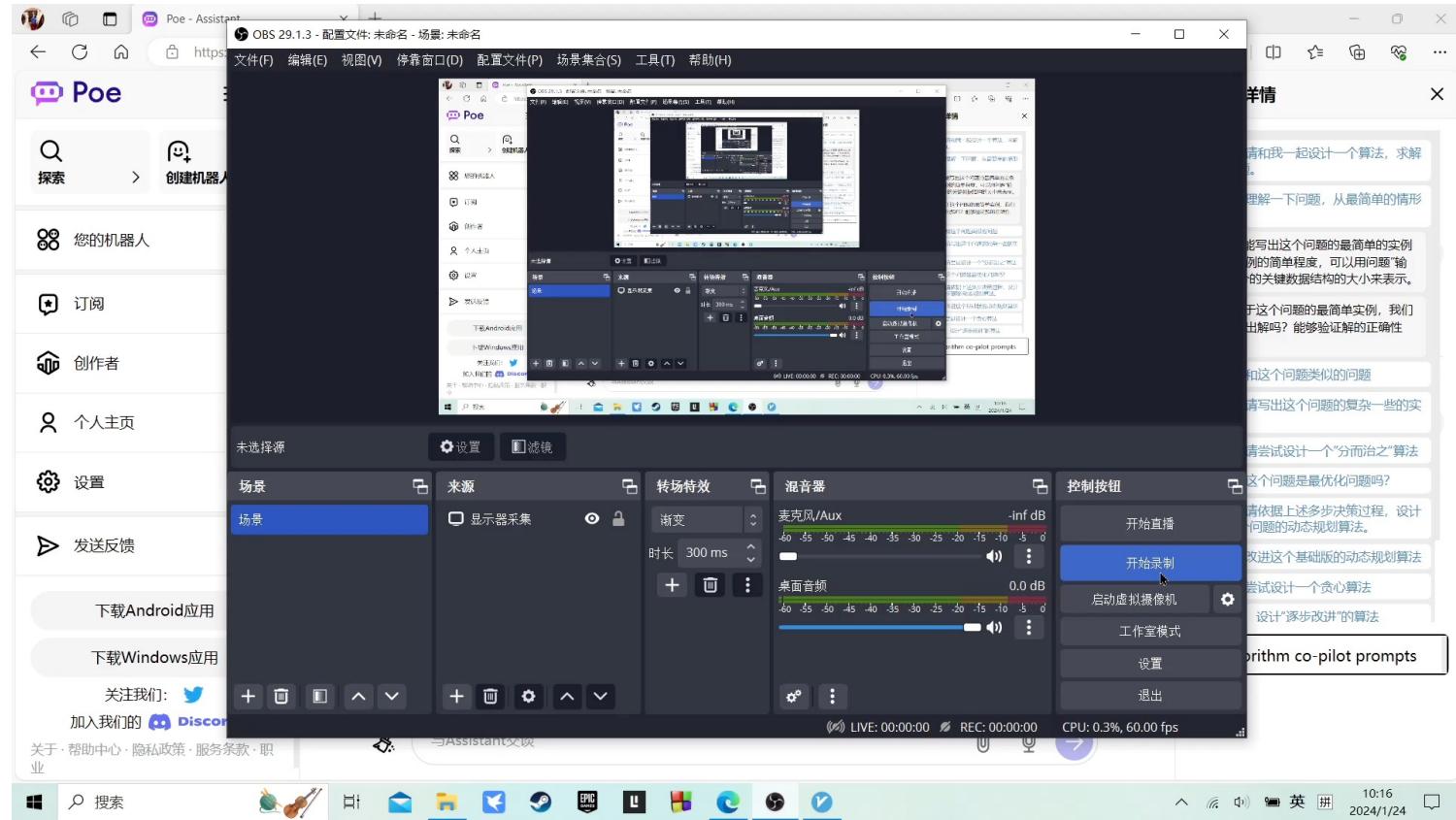


人做探索、提猜想

人机协同做探索、提猜想

- 蒲慕明：如何选题、提猜想？先看综述，少看papers（可否用LLM？）
- 李国杰：LLM是“不确定性计算”，是“猜测机”（需要“验证器”？）
- 杨振宁：特勒一天有十个ideas，其中九个半都是错的（那半个呢？）

思考：prompts对人类学生有效，对LLM有效否？



- **软件：**浏览器插件 **CoPA (Co-pilot for algorithm)**
- **功能：**人机协同，用prompts引导，校验回答，互相启发

初试Llama2：不会做，写出一段重复代码

Q:请设计一个 $O(N^2)$ 时间的算法，求一个n个数的序列的最长单调递增子序列

A: 以下是一个 $O(N^2)$ 时间的算法，用于求一个n个数的序列的最长单调递增子序列：

1. 创建一个长度为n的数组，初始化为空。
2. 遍历数组中的每个元素，将该元素与数组中下一个元素进行比较，如果该元素大于下一个元素，则将该元素与下一个元素交换位置。
3. 遍历数组中的每个元素，将该元素与数组中下一个元素进行比较，如果该元素大于下一个元素，则将该元素与下一个元素交换位置。
4. 遍历数组中的每个元素，将该元素与数组中下一个元素进行比较，如果该元素大于下一个元素，则将该元素与下一个元素交换位置。
5. 遍历数组中的每个元素，将该元素与数组中下一个元素进行比较，如果该元素大于下一个元素，则将该元素与下一个元素交换位置。
6. 遍历数组中的每个元素，将该元素与数组中下一个元素进行比较，如果该元素大于下一个元素，则将该元素与下一个元素交换位置。
7. 遍历数组中的每个元素，将该元素与数组中下一个元素进行比较，如果该元素大于下一个元素，则将该元素与下一个元素交换位置。

再试Llama2：问一个prompt，有些迹象

Q: 请设计一个 $O(N^2)$ 时间的算法，求一个n个数的序列的最长单调递增子序列。该问题能分解成子问题？

A: 可以分解成子问题。设序列为 $[a_1, a_2, \dots, a_n]$ ，其中 $a_1 < a_2 < \dots < a_n$ 。最长单调递增子序列问题可以分解为以下子问题：

1. 最长单调递增子序列问题：求一个n个数的序列的最长单调递增子序列。
2. 最长单调递增子序列问题的子问题：求一个n个数的序列的最长单调递增子序列，其中第n-1个数大于第n-2个数。
3. 最长单调递增子序列问题的子问题：求一个n个数的序列的最长单调递增子序列，其中第n-1个数小于第n-2个数。

因此，可以将最长单调递增子序列问题分解为以上三个子问题，并分别解决。对于第一个子问题，可以使用动态规划的方法，将每个子问题看作一个独立的最长单调递增子序列问题，然后将它们的最长单调递增子序列合并起来。对于第二个子问题，可以使用动态规划的方法，将每个子问题看作一个独立的最长单调递增子序列问题，然后将它们的最长单调递增子序列合并起来。对于第三个子问题，可以使用动态规划的方法，将每个子问题看作一个独立的最长单调递增子序列问题，然后将它们的最长单调递增子序列合并起来。因此，最长单调递增子序列问题可以分解为三个子问题，并分别解决。

三试Llama2：问两个prompts，写出动态规划算法

Q:请设计一个 $O(N^2)$ 时间的算法，求一个n个数的序列的最长单调递增子序列。该问题能分解成子问题？子问题之间是否有最优子结构呢？

A: 最长单调递增子序列问题可以分解成子问题，并且有最优子结构。子问题：

1. 最长单调递增子序列的长度
2. 最长单调递增子序列的起始位置

最优子结构：

1. 最长单调递增子序列的长度
2. 最长单调递增子序列的起始位置 可以使用动态规划的方法来解决这个问题。

具体实现如下：

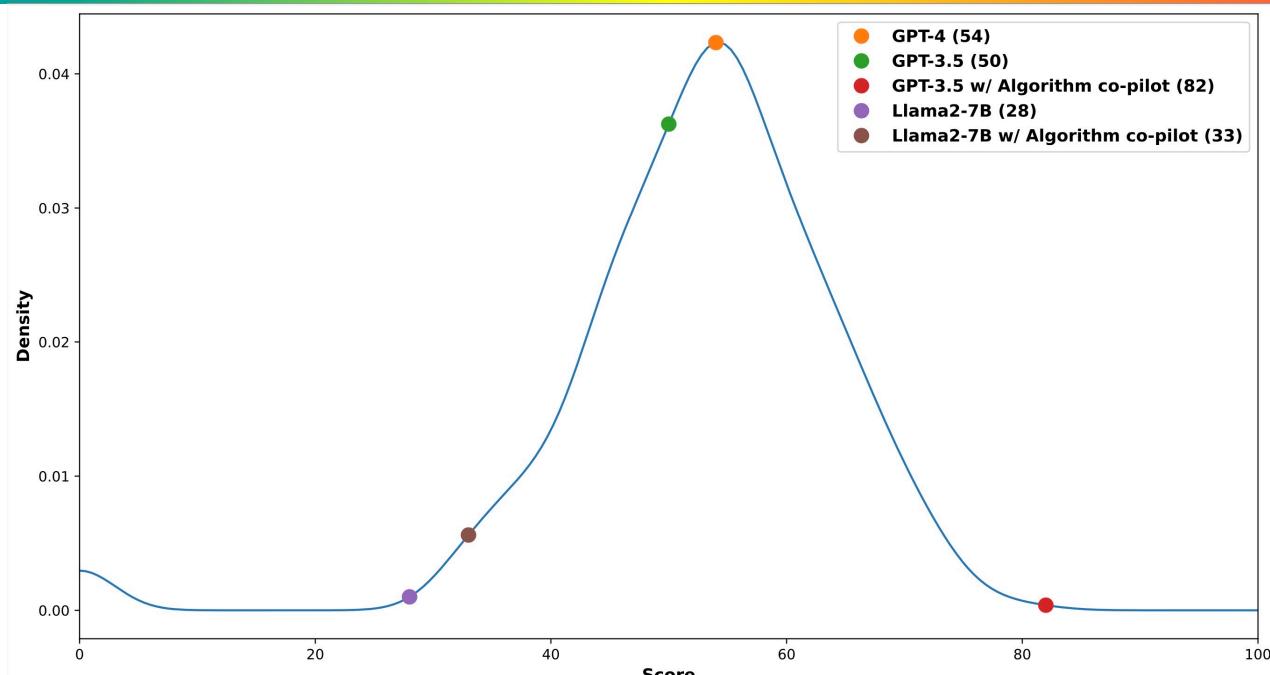
1. 初始化一个长度为n的数组dp，其中 $dp[i]$ 表示数组 $[1, \dots, i]$ 的最长单调递增子序列的长度。
2. 遍历数组 $[1, \dots, n]$ ，对于每个数*i*，判断是否存在一个长度为 $dp[i]$ 的子序列，该子序列的长度为*i*，并且从该子序列的起始位置开始，该子序列的长度为*i*-1。
3. 如果存在，则将 $dp[i]$ 的值更新为该子序列的长度，并将起始位置更新为该子序列的起始位置。
4. 返回 $dp[n]$ 的值。时间复杂度为 $O(n^2)$ 。

2023年的算法课：一个教学实验



- 新同学：AI, #359
- 大模型：用每次课件finetune大模型（寒武纪290）
- 期末：AI和同学们一同考试

期末考试： 和358名同学相比， AI排第几？



- 试题：
 - 分而治之：2，贪心：2，动态规划：2，线性规划：1
- 结果：
 - GPT3.5：第180名左右，50分
 - GPT3.5+CoPA：第1名，82分

更难的测试：3道ACM竞赛题，GPT4

- **选择标准：**
 - 3道题GPT不会
 - 3道题操作者也不会，避免诱供
- **结果：**
 - 人机协同，解出来两道题

我的愿景2024

- **CoPA计划:**
 - 有一位同学用CoPA，人机协同，写出过去写不出的程序
 - 写个agent，自动输入prompts，自动检验答案
- **ProDESIGN蛋白质设计:**
 - 有一位生物学家用VR+AI，人机协同，设计出一个有功能的抗体
- **探索“什么是涌现？”**
 - 涌现就是“最小描述长度的相变”！（与李明老师合作）

写在最后：从万物皆数到万物皆算

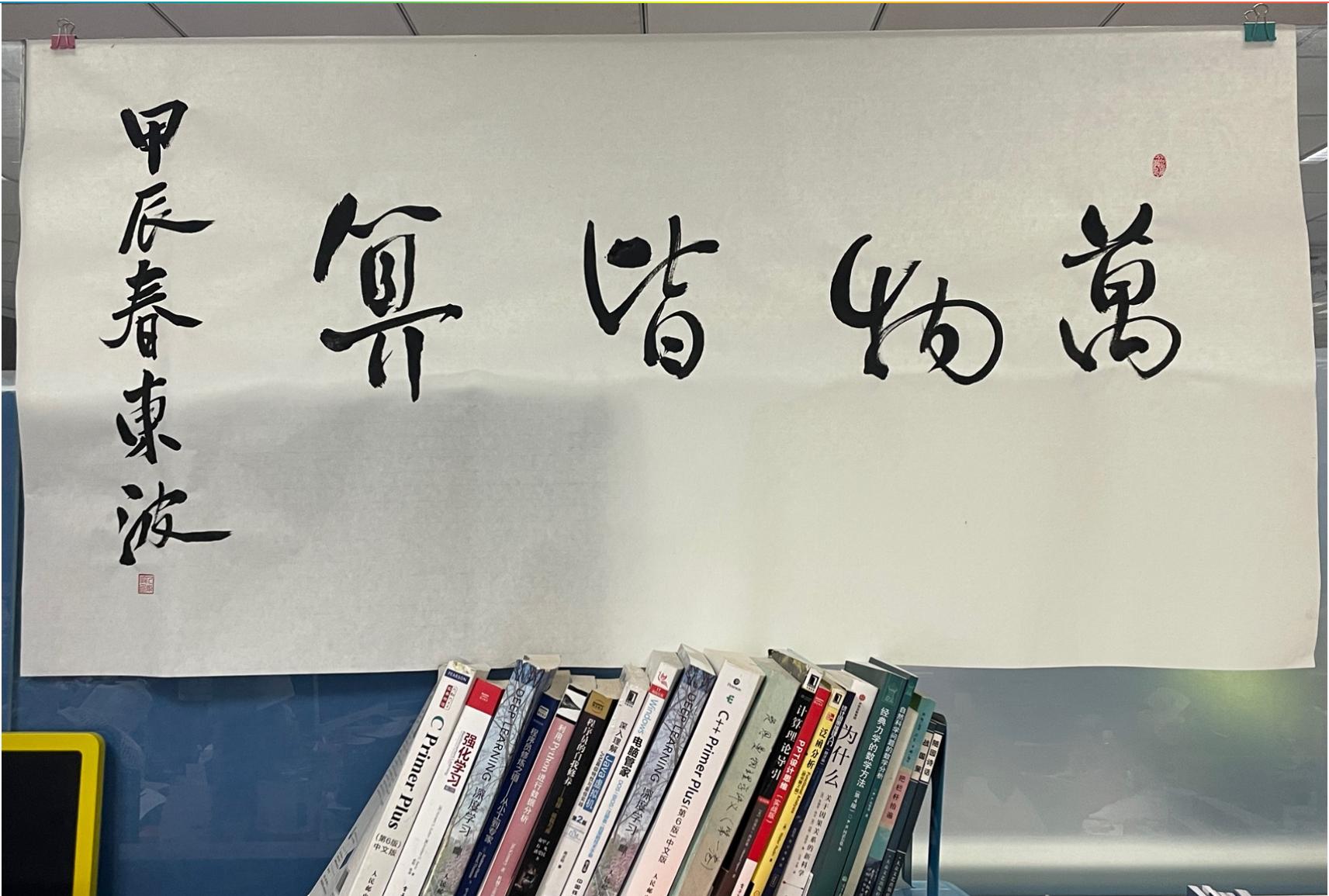


R. Karp

两弦长比 $2:3$, 和弦最是好听

世间万物，尽在一算

寄语我的学生：苦候灵感，莫若一算



END