



中山大學
SUN YAT-SEN UNIVERSITY



多核程序设计与实践

期末大作业要求及选题参考

陶钧

taoj23@mail.sysu.edu.cn

中山大学 数据科学与计算机学院
国家超级计算广州中心

● 评分标准

- | | |
|---|-----|
| – 完成情况 | 50% |
| – 技术难度加分 | 20% |
| • 实现难度（工程量）、优化难度（工程量） | |
| • 例如，使用多GPU进行编程、MPI+CUDA、针对大规模数据的out-of-core计算、CUDA+OpenGL、通过界面提供交互功能等 | |
| – 报告 | 30% |
| • 书面报告：每个小组应由所有成员共同出具一份书面报告，说明大作业项目的选题，预期完成的目标，实际完成的内容，实施的具体方案，在实施过程中解决的问题，并行算法的效率（加速比）等。小组中每个成员应提交一份个人报告，说明个人在小组中发挥的作用，及完成的工作。个人报告以附录形式，附在小组报告后。 | |
| • 现场报告：现场报告与书面报告内容一致，形式不同。 | |

● 关键时间节点

- 提交时间均为当日晚23: 59分前
- 5月6日（11周周一）：选题报告
 - 小组成员名单（每组4-5人：由于现场报告时间限制，为避免分组过多，原则上不允许低于4人组队；由于工作量极大需要5人以上组队的请直接联系我申请）
 - 选题内容（例如，预期实现算法）
- 6月17日（17周周一）：书面报告
 - 无论现场报告时间安排，统一提交书面报告
- 6月18日（17周课上）及6月25日（18周课上）：现场报告
 - 要求所有同学到场

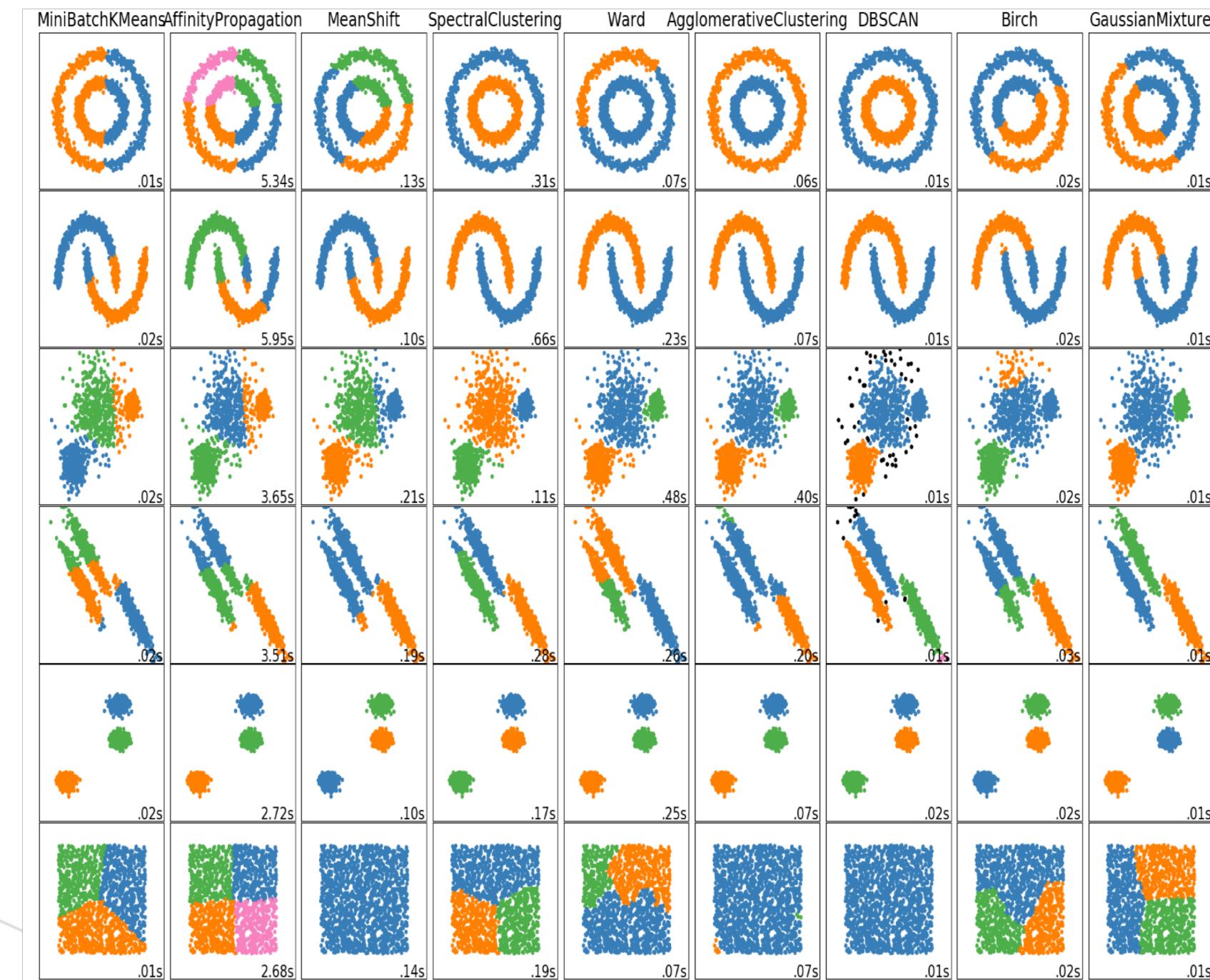
- 实现并行算法或并行化传统算法

- 选题标准：适合GPU并行化
 - 通过相似方式处理大量数据

- 参考选题

- 聚类算法
 - 图布局算法
 - 图采样算法
 - 最近邻搜索
 - 模拟退火
 - 计算机视觉算法（blob detection等）
 - 传统算法并行化（图遍历，最短路径等）

- 目的：按照某种规则将一组数据对象分割成不同的类（clusters）使同一类的对象间相似性尽可能大，同时不同类的对象间差异也尽可能大。



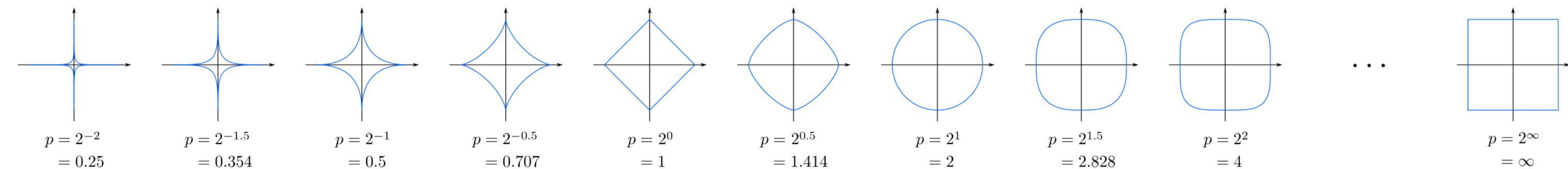
图片来自 <https://scikit-learn.org/stable/modules/clustering.html>

● 相似性度量

– 明氏距离（Minkowski distance/L_p-norm）

$$D(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

- $p = 1$: 曼哈顿距离
- $p = 2$: 欧氏距离
- $p = \infty$: 切比雪夫距离



● 相似性度量

– 余弦相似性 (cosine similarity)

- 根据向量间夹角计算相似性

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

– KL散度 (Kullback-Leibler divergence/相对熵)

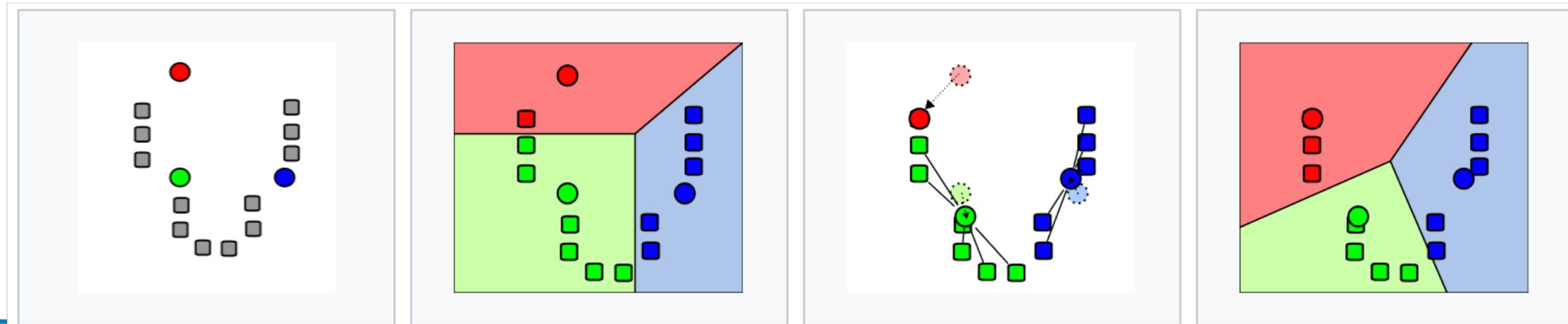
- 基于信息论
 - 使用基于概率分布Q的编码对来自概率分布P的样本编码时所需额外的位元数
- 常用来衡量两个概率分布的差别 (非对称)

$$D_{\text{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

● 常见聚类算法

- k-means、DBSCAN、层次聚类（hierarchical clustering）、混合高斯模型（GMM）聚类、affinity propagation等
- 举例：k-means
 - 基于中心（centroid-based）聚类
 - 1. 随机选择中心
 - 2. 根据到中心的距离将数据对象分配至不同类
 - 3. 更新中心位置，并重复步骤3

图片来自https://en.wikipedia.org/wiki/K-means_clustering

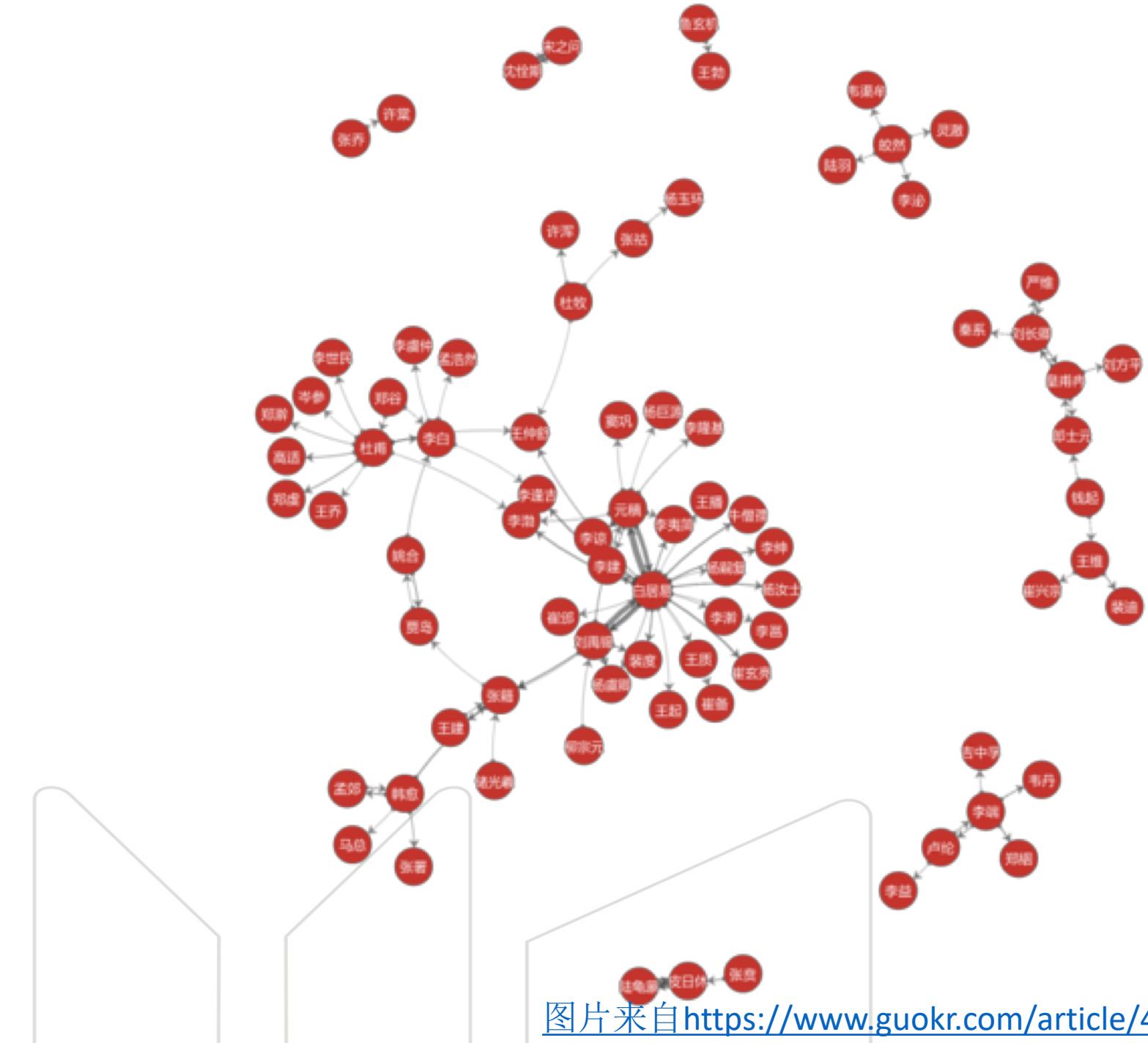


● 参考资料

- Scikit-learn (python库) 的聚类算法页面
 - <https://scikit-learn.org/stable/modules/clustering.html>
- 维基百科：聚类算法
 - https://en.wikipedia.org/wiki/Cluster_analysis
- 维基百科：k-means聚类
 - https://en.wikipedia.org/wiki/K-means_clustering
- 维基百科：余弦相似性
 - https://en.wikipedia.org/wiki/Cosine_similarity
- 维基百科：KL散度
 - https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

- 目标：将数据点根据一定规则（例如，距离）放置在二维平面上，使数据点在平面上的位置关系能反映其在原始空间中的关系

— 右图：唐朝诗人关系网

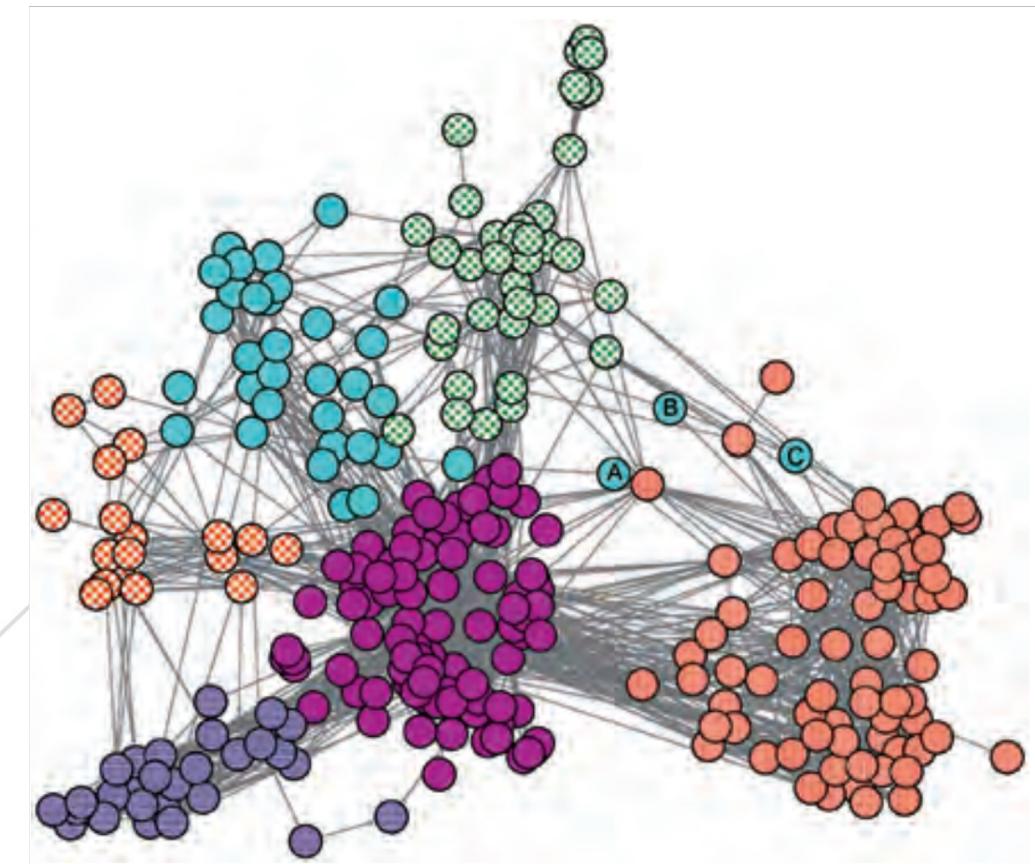


● 常见算法

– 力导向布局(force-directed graph layout)

- FR-layout, KW-layout, stress majorization
- 最小化能量方程

$$E_s = \sum_{i=1}^n \sum_{j=1}^n \frac{1}{2} k(d(i,j) - s(i,j))^2$$

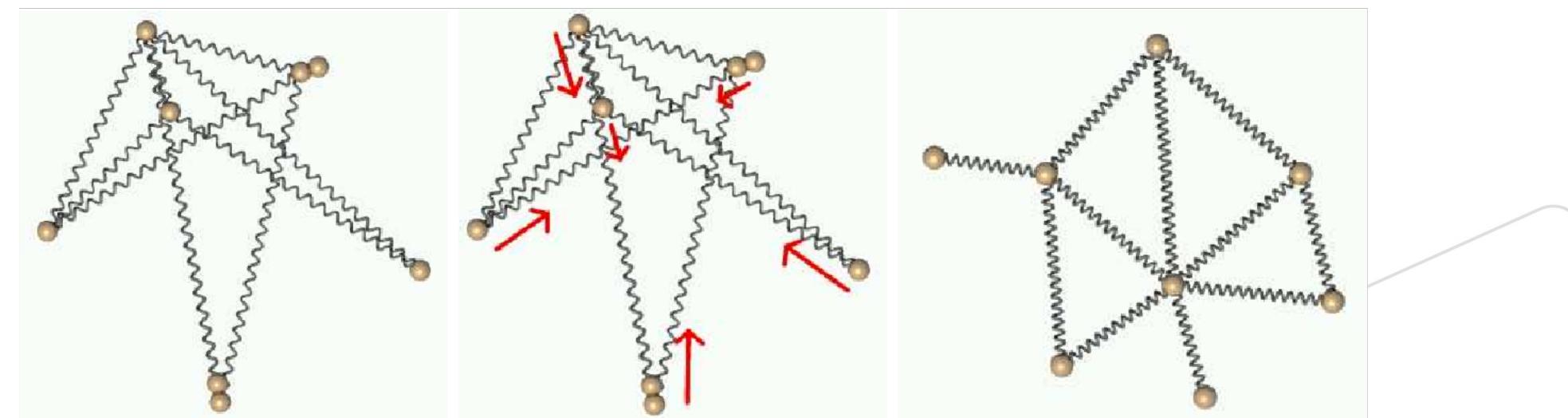


● 常见算法

– 力导向布局(force-directed graph layout)

- FR-layout, KW-layout, stress majorization
- 使用弹簧系统模拟图布局
 - 当两点在平面上距离小于目标距离时产生排斥力
 - 当两点在平面上距离大于目标距离时产生牵引力

– 降维算法（Dimension reduction）：MDS、LLE、t-SNE等



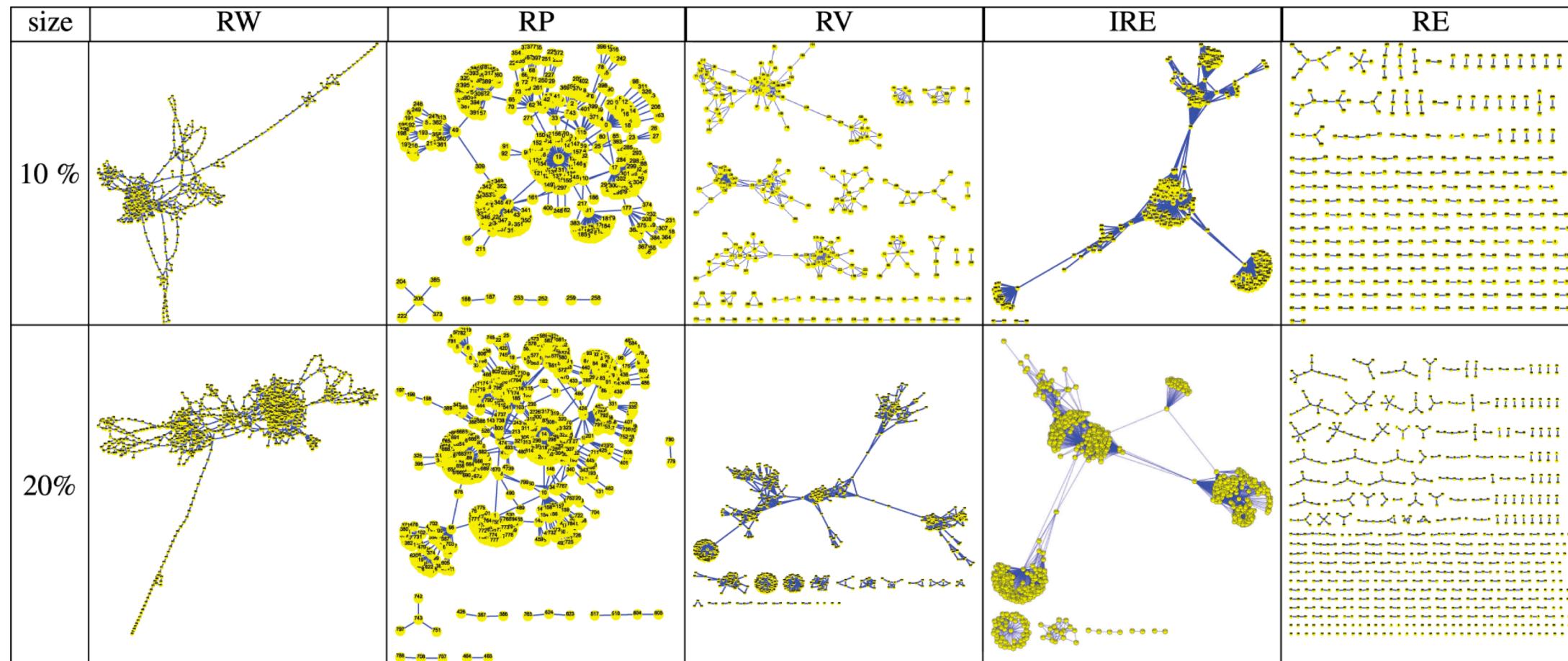
Kobourov, Spring Embedders and Force Directed Graph Drawing Algorithms, ArXiv, 2012

● 参考资料

- Force-Directed Drawing Algorithms
 - <http://cs.brown.edu/people/rtamassi/gdhandbook/chapters/force-directed.pdf>
- Gibson et al., A Survey of Two-Dimensional Graph Layout Techniques for Information Visualization
 - <http://www.leonidzhukov.net/hse/2015/sna/papers/gibson2013>
- Cui, A Survey of Graph Visualization
 - <https://pdfs.semanticscholar.org/2a52/cd195942cd901d73c118d6a66c97934df3fb.pdf>

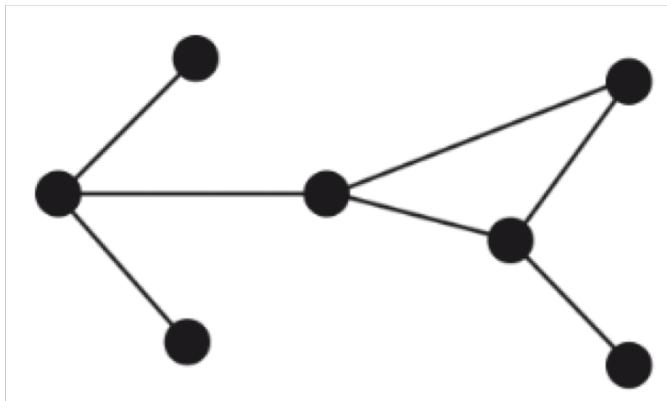


- 目标：对大规模图中的节点及边进行采样，使采样得到的图尽可能反映原图的结构或特征

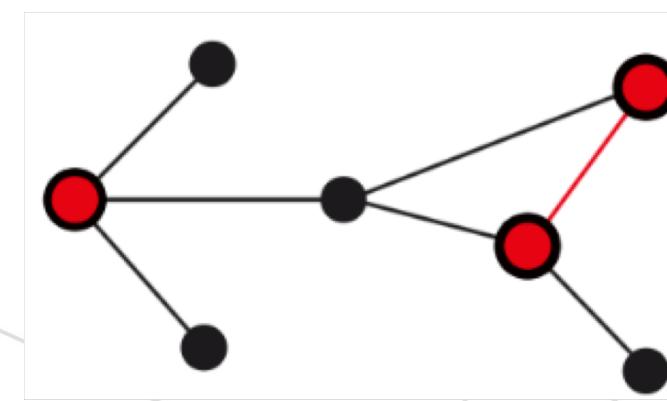


● 常见方法

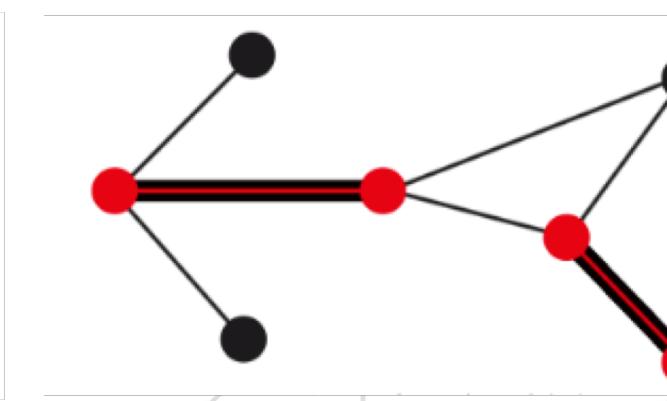
- 随机点采样、随机边采样
- 基于图遍历的算法
 - Random walk, metropolis-hasting random walk, snow-ball sampling, forest fire sampling



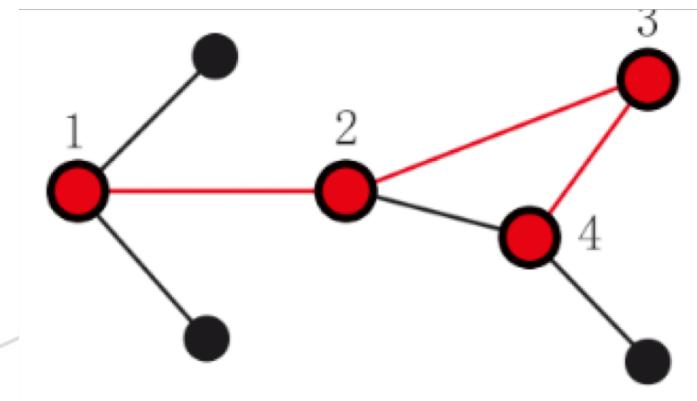
原图



随机点采样



随机边采样



Random walk

● 参考资料

- A Survey and Taxonomy of Graph Sampling
 - 主要参考第四节 (Traversal Based Sampling)
 - <https://arxiv.org/pdf/1308.5865.pdf>
- 上海交通大学复杂网络与控制实验室，相关采样算法
 - <http://cnc.sjtu.edu.cn/sampling/method.html>



- 目标：在尺度空间中给定一个点集 S 和一个目标点 p ，在 S 中找出距离 p 最近的点
 - 模式识别，计算机视觉，碰撞检测，光线追踪，几何体的距离度量（MCP距离、Hausdorff距离等），N-body模拟等
 - k-NN (k-nearest neighbors)：在 S 中找出离 p 最近的 k 个点

- 直接思路

- 线性查找：计算 p 与 S 中每一个点的距离，并找出距离最小的（一个或 k 个）点
 - 并行计算距离，并行归约

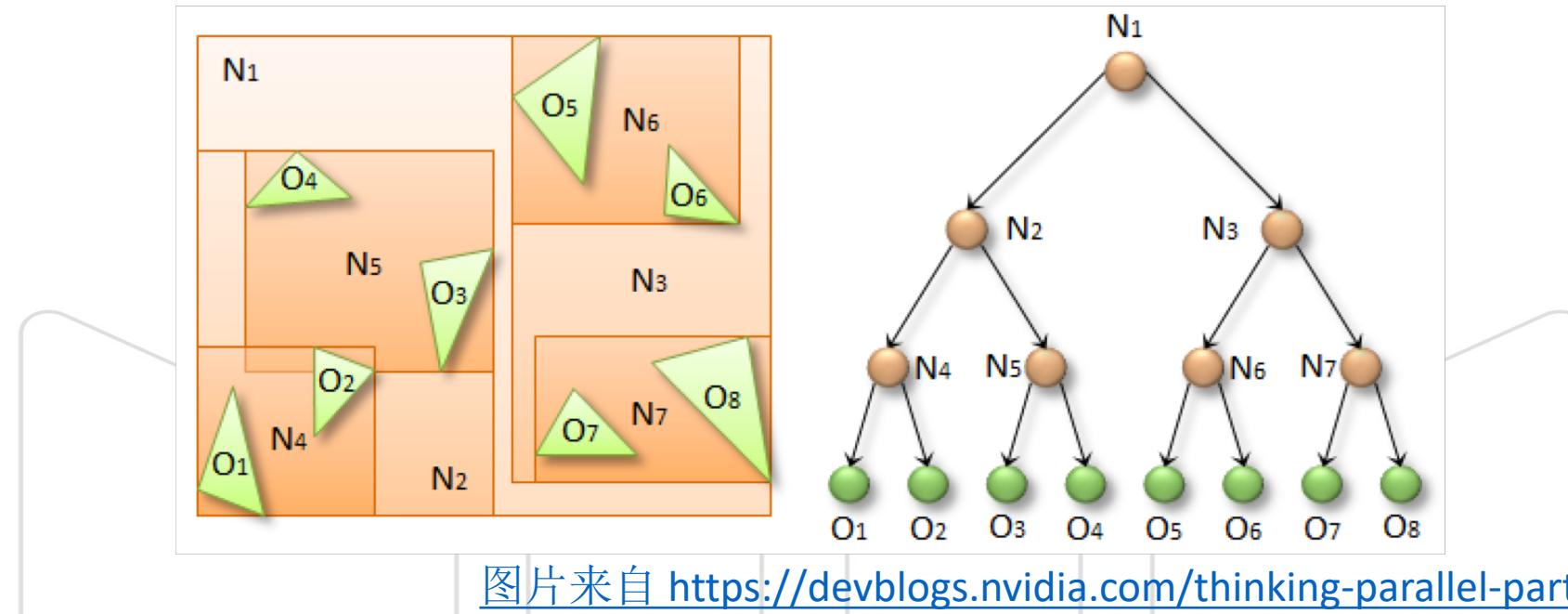
● 通过树结构进行查找

- 常见数据结构

- bounding volume hierarchy (BVH)-tree, octree, kd-tree等

- 举例：BVH-tree

- 使用包围盒 (bounding volume) 将物体分配到树的节点中
- 在查找最近邻时，先判断与包围盒的距离



图片来自 <https://devblogs.nvidia.com/thinking-parallel-part-ii-tree-traversal-gpu/>

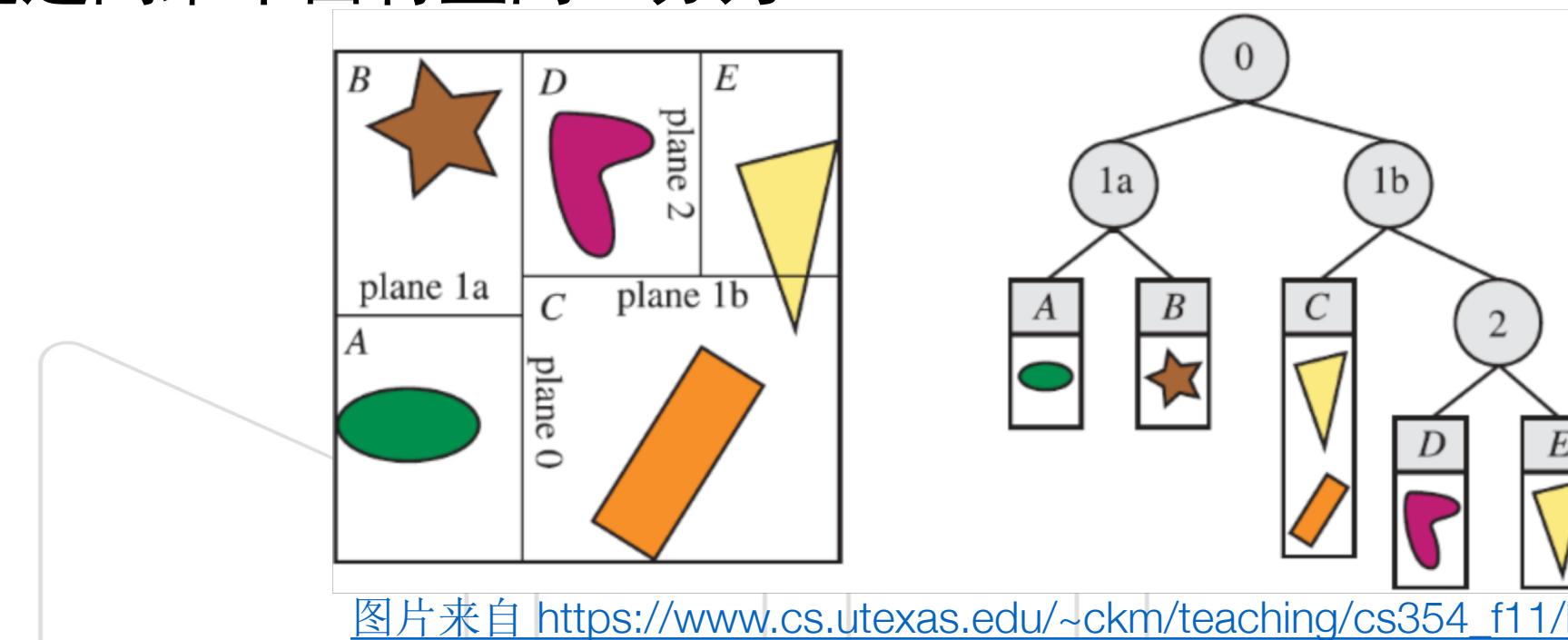
- 通过树结构进行查找

- 常见数据结构

- bounding volume hierarchy (BVH)-tree, octree, kd-tree等

- 举例：kd-tree

- Octree (八叉树) 的高维扩展
 - 每次通过高维平面将空间一分为二



图片来自 https://www.cs.utexas.edu/~ckm/teaching/cs354_f11/lectures/Lecture21.pdf 19

● 参考资料

- Brown and Snoeyink, GPU Nearest Neighbor Searches Using a Minimal kd-Tree
 - <http://on-demand.gputechconf.com/gtc/2010/presentations/S12140-Superfast-Nearest-Neighbor-Searches-Using-Minimal-kd-Tree.pdf>
- Karras, Maximizing Parallelism in the Construction of BVHs, Octrees, and k-d Trees
 - https://research.nvidia.com/sites/default/files/publications/karras2012hpg_paper.pdf
- Lauterbach et al., Fast BVH Construction on GPUs
 - http://graphics.snu.ac.kr/class/graphics2011/references/2007_lauterbach.pdf

- 目标：寻找目标函数的全局最优解

- 采取随机行动

- 随机行动范围随着算法运行逐渐减小（退火）
 - 可能使用较差的解代替当前解

Let $s = s_0$

For $k = 0$ through k_{\max} (exclusive):

$T \leftarrow \text{temperature}(k / k_{\max})$

Pick a random neighbour, $s_{\text{new}} \leftarrow \text{neighbour}(s)$

If $P(E(s), E(s_{\text{new}}), T) \geq \text{random}(0, 1)$:

$s \leftarrow s_{\text{new}}$

Output: the final state s

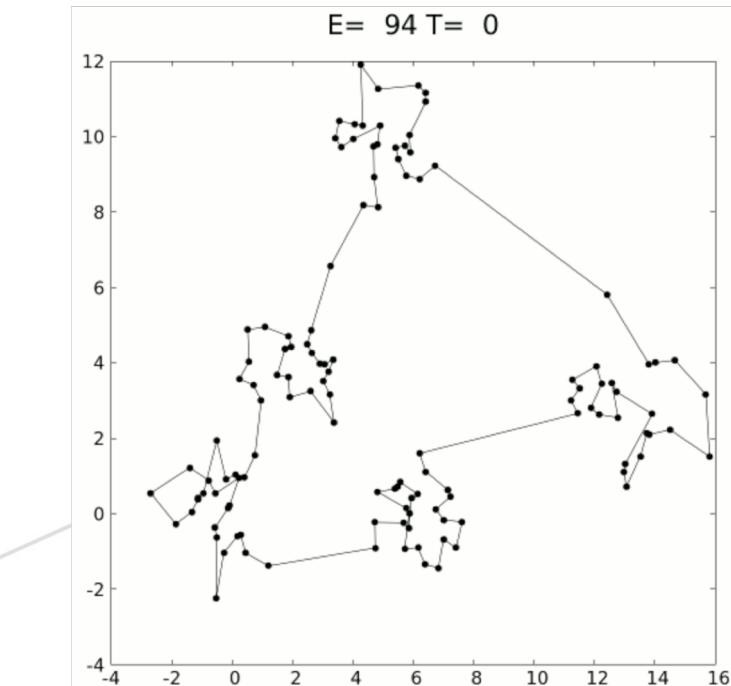
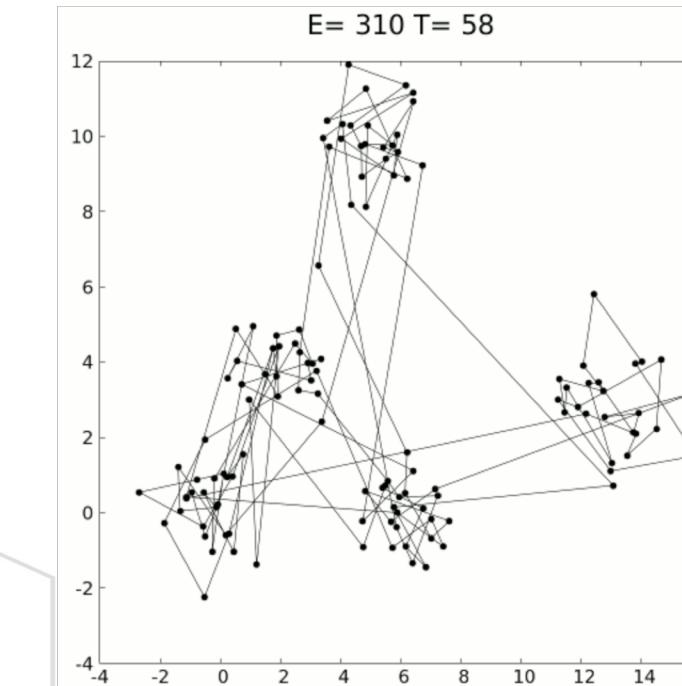
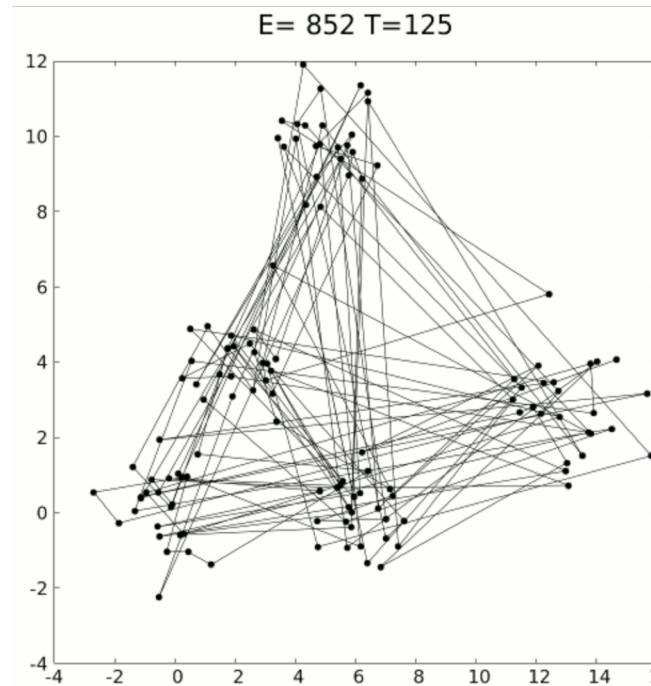


● 应用：

– 旅行推销员问题 (travelling salesman problem)

- 给定一组点及每两个点之间的距离，求访问每个点一次并回到起始点的最短回路
- NP问题
- 使用模拟退火随机改变路径

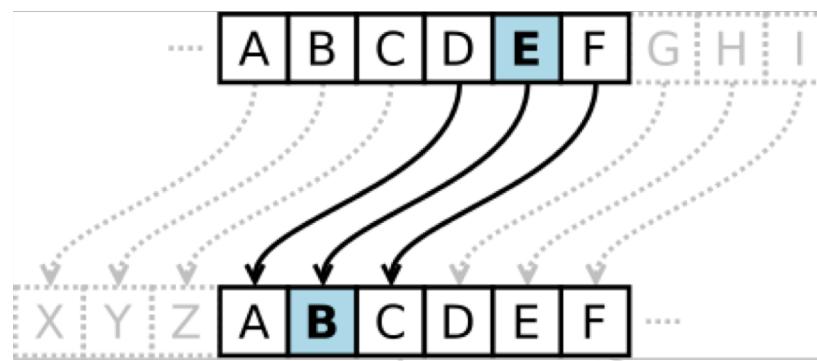
图片来自 https://en.wikipedia.org/wiki/Simulated_annealing



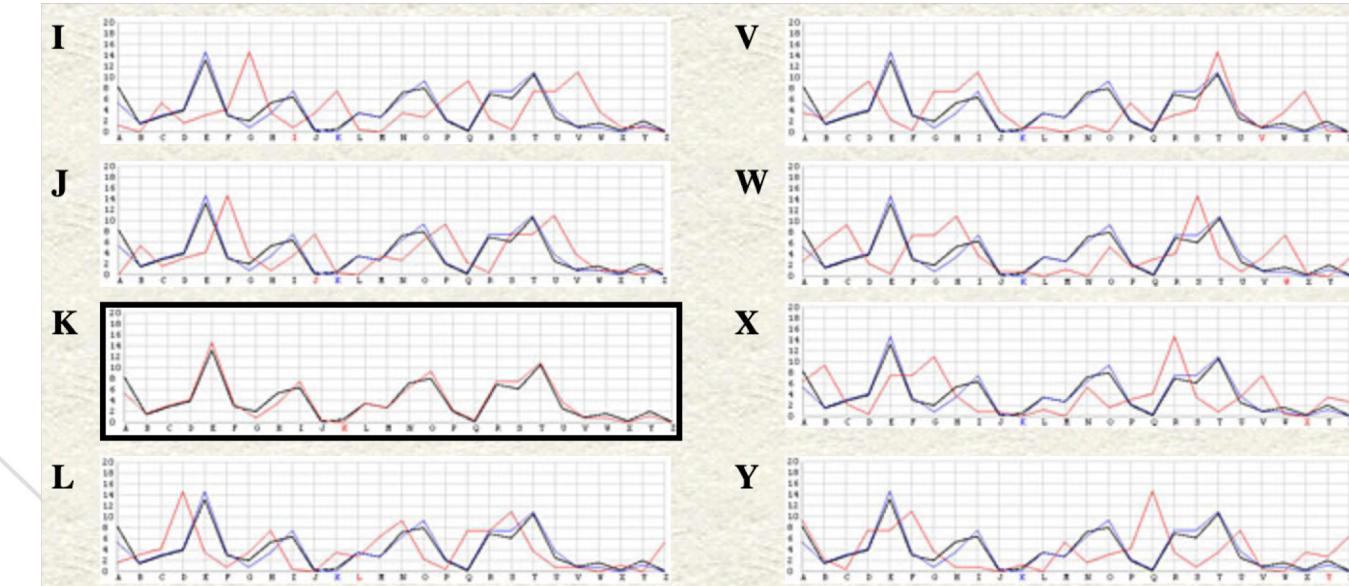
● 应用：

– 破译经典加密算法

- Caesar cipher, Vigenère cipher, Playfair cipher
- Caesar cipher: 每个字母移动固定步数
 - 不改变单个字母分布pattern（只是平移）
 - 不需要模拟退火，但可并行
 - 使用 χ^2 距离计算两个分布间的距离



图片来自 https://en.wikipedia.org/wiki/Caesar_cipher



图片来自 <https://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/Vig-Frequency-Analysis.html>

● 应用：

– 破译经典加密算法

- Vigenère cipher：每个字母移动由秘钥指定的固定步数
 - 例子：使用秘钥LEMON
 - 改变单个字母总体分布pattern，但不改变隔步长（秘钥长度）的分布
 - 先确定秘钥长度，再确定每一位上的字母

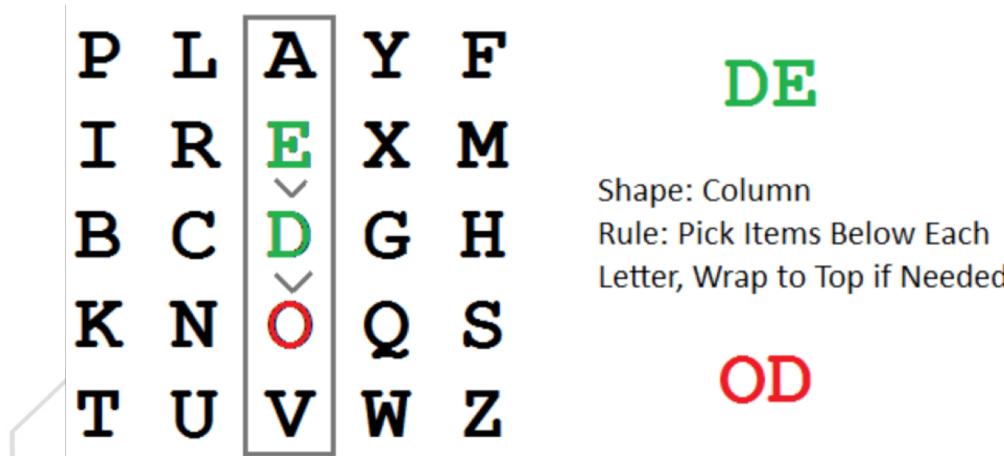
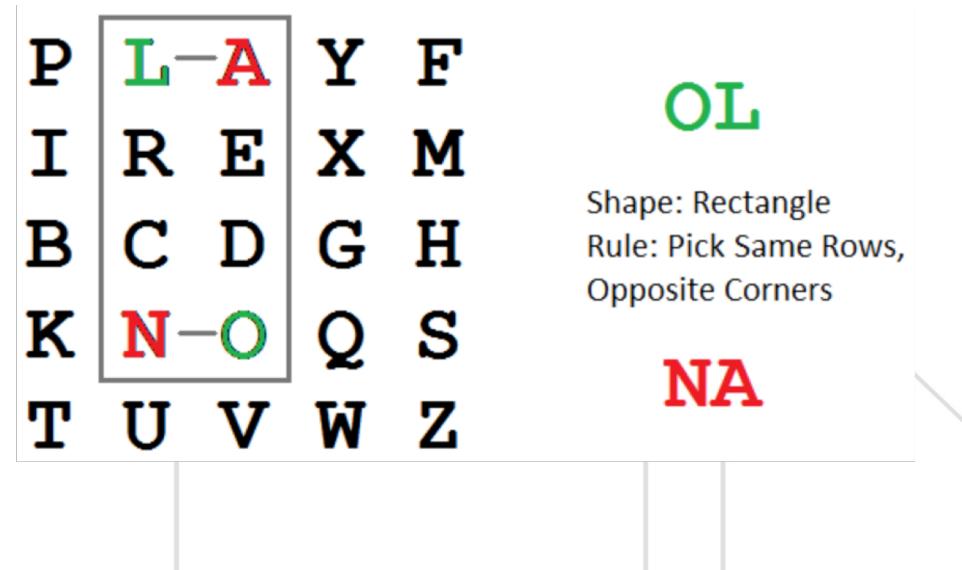
Plaintext:	ATTACKATDAWN
Key:	LEMONLEMONLE
Ciphertext:	LXFOPVEFRNHR

图片来自 https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher

● 应用：

– 破译经典加密算法

- Playfair cipher基于一个5x5的字母矩阵将原文中的一对字母加密为密文中的一个字母
 - 打破单个字母的分布pattern
 - 但字母对的出现频率将提示解密是否正确
 - » 使用距离计算解密得到的字母对概率分布与标准英文的字母对分布之间的差异
 - » 随机修改矩阵不断降低差异



图片来自 https://en.wikipedia.org/wiki/Playfair_cipher

● 参考资料

- Wikipedia, Simulated Annealing
 - [https://en.wikipedia.org/wiki/Simulated annealing](https://en.wikipedia.org/wiki/Simulated_annealing)
- Simulated Annealing, The Travelling Salesman Problem
 - <https://www.fourmilab.ch/documents/travelling/anneal/>
- Wikipedia, Playfair Cipher
 - [https://en.wikipedia.org/wiki/Playfair cipher](https://en.wikipedia.org/wiki/Playfair_cipher)
- Cowan, Breaking Short Playfair Cipher with the Simulated Annealing Algorithm
 - <https://pdfs.semanticscholar.org/1b7d/937237bd17e49301045df20d7f31a818698f.pdf>
- Shene, Cryptography Visualization Tools: A Tutorial
 - <https://pages.mtu.edu/~shene/NSF-4/Tutorial/index.html>

● 拓宽选题思路

- 计算机视觉算法（blob detection）
- 传统算法或数据结构的并行化
- 选取感兴趣的论文，并将其内容并行化
- 任何适合GPU计算模式的选题
 - 通过GPU能获得效率提升



Questions?

