Randomised algorithms

April 30, 2026

1 Inclusions of complexity classes

Exercise 1.

Show that $ZPP = RP \cap co-RP$.

Exercise 2.

Verify the following inclusions: $RP \subseteq BPP \subseteq PP$.

Exercise 3.

Show the following inclusions: $RP \subseteq NP \subseteq PP$.
Hint: for $NP \subseteq PP$, prove that 3-SAT belongs to PP.

2 Finding minimal and maximal cuts

Exercise 4.

Let G be a graph. A *cut* in G is a bipartition of the vertices $A \sqcup B = V(G)$. The weight of a cut $A \sqcup B$ is the total number of edges going from A to B. The problem MAXCUT consists in finding a cut of maximum weight.

Design a randomised 1/2-approx for MAXCUT.

Exercise 5. Karger's algorihtm

Given a multi-graph G, we are interested in finding a cut of minimum weight (the weight of a multi-edge between u and v is the number of edges between u and v). Let uv be an edge G. Contracting uv consists in merging the vertices u and v into one vertex w. More precisely, it removes u and v, adds a new vertex w and reattaches each edge that was incident to u or v to w (note that the common neighbours of u and v will have a multi-edge to w after this operation). We denote G/uv the graph obtained from G after contracting uv.

Karger's algorithm for the minimum cut problem constructs a sequence of graphs $G_0 = G, \dots G_{n-2}$ such that each G_i has $n - i$ vertices and is obtained from G_{i-1} by selecting an edge uv uniformly at random in G_{i-1} and setting $G_i = G_{i-1}/uv$. The graph G_n has 2 vertices a and b, the algorithm returns $A \sqcup B$ where A (resp. b) is the set of vertices that were contracted into a (resp. b).

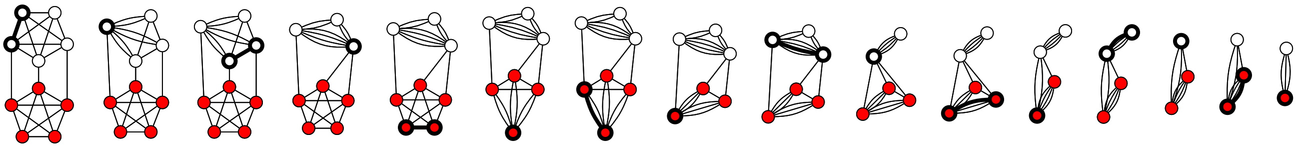

Figure 1: An execution of Karger's algorithm returning a minimum cut (credit: Thore Husfelt)

1. Argue that for every i, the weight of a minimum cut of G_i is at least that of a minimum cut of G.
2. Let $A \sqcup B$ be a cut of minimum weight in G. Show that Karger's algorithm returns $A \sqcup B$ with probability at least $\binom{n}{2}^{-1}$

3. How many times should we run Karger's algorithm to obtain a minimum cut with constant probability?
4. Suppose that instead of choosing an edge uniformly at random, one chooses a pair of adjacent vertices uniformly at random (i.e. not taking into account the number of edges between u and v) and merge them. Show that the probability of returning a minimum cut can then be exponentially small.

★ Exercise 6. Goemans-Williamson Maxcut approximation scheme

A semi-definite program is a problem of the form:

$$\underset{x^{(1)},\dots x^{(n)}\in\mathbb{R}^n}{\text{maximise}} \sum_{i,j\in[n]} c_{i,j}(x^{(i)}\cdot x^{(j)}) \qquad\qquad \text{subject to } \sum_{i,j} a_{i,j,k}(x^{(i)}\cdot x^{(j)}) \leq b_k \text{ for every } k$$

where $x \cdot y = \sum_{i=1}^{n} x_i y_i$ is the scalar product between the vectors x and $y \in \mathbb{R}^n$.

We are interested in finding a Monte-Carlo approximation algorithm for the MAXCUT problem. To do so, we follow an approach similar to algorithm described for MAXSAT in the lecture: first express the problem as an integer integer quadratic program (instead of integer linear program for MAXSAT), then relax this integer quadratic program (IQP) into a semi-definite program (SDP) and solve it, finally, round randomly the solution of this SDP to approximate the solution of the IQP.

1. Let G be a graph on n vertices, let $A \sqcup B$ be a cut of G. For each vertex u, let $x^{(u)} = 1$ if $u \in A$ and $x^{(u)} = -1$ if $u \in B$. Express the weight of the cut $A \sqcup B$ as a polynomial in the variables $(x^{(u)})_{u\in V(G)}$. Let $f((x^{(u)})_{u\in V(G)})$ be the corresponding objective function.
2. For each vertex u, let $y^{(u)} = (1, 0, \dots 0) \in \mathbb{R}^n$ if $u \in A$ and $y^{(u)} = (-1, 0 \dots 0) \in \mathbb{R}^n$ if $u \in B$. Express the weight of the cut $A \sqcup B$ as a polynomial in the variables $(y^{(u)} \cdot y^{(v)})_{u,v\in V(G)}$. Let $g((y^{(u)})_{u\in V(G)})$ be the corresponding objective function.
3. We now relax the constraint $y^{(u)} = (\pm 1, 0, \dots 0)$ and replace it by $\left\| y^{(u)} \right\|_2 = 1$. Argue that the corresponding optimisation problem is a semi-definite program.
4. Denote $(\hat{y}^{(u)})_{u\in V(G)}$ a solution of this semi-definite program. Informally, when is some edge uv contributing to the objective function?
5. We now do the randomised rounding step. Define a way of sampling random variables $X^{(u)} \in \{-1, +1\}$ for $u \in V(G)$ such that $\mathbb{E}[f((X^{(u)})_{u\in V(G)})] \geq 0.878 \cdot g((\hat{y}^{(u)})_{u\in V(G)})$.
6. Conclude.