“Sapienza” University of Rome

Faculty of Information Engineering, Informatics and Statistics

Department of Computer Science

Advanced Algorithms

Lecture notes integrated with the book "My book",
Author 1, ...

Author

Alessio Bandiera

February 27, 2025

Contents

Information and Contacts

Personal notes and summaries collected as part of the *Advanced Algorithms* course offered by the degree in Computer Science of the University of Rome "La Sapienza".

Further information and notes can be found at the following link:
https://github.com/aflaag-notes. Anyone can feel free to report inaccuracies, improvements or requests through the Issue system provided by GitHub itself or by contacting the author privately:

- Email: alessio.bandiera02@gmail.com
- LinkedIn: [Alessio Bandiera](#)

The notes are constantly being updated, so please check if the changes have already been made in the most recent version.

Suggested prerequisites:

- Progettazione degli Algoritmi

Licence:

These documents are distributed under the [GNU Free Documentation License](#), a form of copyleft intended for use on a manual, textbook or other documents. Material licensed under the current version of the license can be used for any purpose, as long as the use meets certain conditions:

- All previous authors of the work must be **attributed**.
- All changes to the work must be **logged**.
- All derivative works must be **licensed under the same license**.
- The full text of the license, unmodified invariant sections as defined by the author if any, and any other added warranty disclaimers (such as a general disclaimer alerting readers that the document may not be accurate for example) and copyright notices from previous versions must be maintained.
- Technical measures such as DRM may not be used to control or obstruct distribution or editing of the document.

<div align="right">

1

TODO

</div>

1.1 TODO

1.1.1 The Max Cut problem

The **Max Cut problem** — in the unweighted case — is a classic combinatorial optimization problem in the branch of graph theory, in which we seek to partition the vertices of an undirected graph into two disjoint subsets while maximizing the number of edges that have endpoints in both subsets. More formally, we will define a **cut** of a graph as follows.

> **Definition 1.1: Cut of a graph**
>
> Given an undirected graph $G = (V, E)$, and a subset of its vertices $S \subseteq V$, the **cut** induced by S on G is defined as follows
>
> $$\mathrm{cut}(S) := \{e \in E \mid |S \cap e| = 1\}$$

Note that in the definition above we are defining the cut of a graph through the intersection between a set of vertices S and edges in E; this is because, in the undirected case, we will consider the edges of a graph $G = (V, E)$ as sets of 2 elements

$$E = \{\{u, v\} \mid u, v \in V\}$$

Therefore, given a set of vertices S, the cut induced by S is simply the set of edges that have only one endpoint in S (implying that the other one will be in $V - S$).
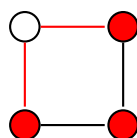

Figure 1.1: Given the set of red vertices S, the green edges represent $\mathrm{cut}(S)$.

With this definition, we can introduce the **Max Cut** problem, which is defined as follows.

> **Definition 1.2: Max Cut problem**
>
> Given an undirected graph $G = (V, E)$, determine the set $S \subseteq V$ that maximizes $|\text{cut}(S)|$.

Although this problem is known to be APX-Hard [1], approximation algorithms and heuristic methods like greedy algorithms and local search are commonly used to find near-optimal solutions.

For now, we present the following **randomized algorithm**, which provides a straightforward $\frac{1}{2}$-approximation for the Max Cut problem. This algorithm runs in polynomial time and achieves the approximation guarantee with high probability.

> **Algorithm 1.1: Random Cut**
>
> Given an undirected graph $G = (V, E)$, the algorithm returns a cut of the graph.
>
> ---
>
> 1: **function** RANDOMCUT(G)
> 2: $S := \varnothing$
> 3:   **for** $v \in V$ **do**
> 4:   Let c_v be the outcome of the flip of an independent fair coin
> 5:   **if** $c_v ==$ heads **then**
> 6: $S = S \cup \{v\}$
> 7:   **end if**
> 8:   **end for**
> 9:   **return** S
> 10: **end function**

Note that this algorithm does not care about the structure of the graph in input, since the output is completely determined by the coin flips performed in the `for` loop. Now we will prove that this algorithm is a correct **expected $\frac{1}{2}$-approximation** of the Max Cut problem.

> **Theorem 1.1: Expected approximation ratio of RANDOMCUT**
>
> Given a graph $G = (V, E)$, and an optimal solution S^* to the Max Cut problem on G. Let $S = \text{RANDOMCUT}(G)$; then, it holds that
>
> $$\mathbb{E}[|\text{cut}(S)|] \geq \frac{|\text{cut}(S^*)|}{2}$$

Proof. By definition, note that

$$\forall e \in E \quad e \in \operatorname{cut}(S) \iff |S \cap e| = 1$$

Consider an edge $e = \{v, w\} \in E$; then, by definition

$$\{v, w\} \in \operatorname{cut}(S) \iff (v \in S \wedge w \notin S) \vee (v \notin S \wedge w \in S)$$

and let ξ_1 and ξ_2 be these last two events respectively. Then

$$\Pr[\xi_1] = \Pr[c_v = \text{heads} \wedge c_w = \text{tails}]$$

by definition of the algorithm, and by independence of the flips of the fair coins we have that

$$\Pr[\xi_1] = \Pr[c_v = \text{heads}] \cdot \Pr[c_w = \text{tails}] = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

Analogously, we can show that

$$\Pr[\xi_2] = \frac{1}{4}$$

This implies that

$$\Pr[e \in \operatorname{cut}(S)] = \Pr[\xi_1 \vee \xi_2] = \Pr[\xi_1] + \Pr[\xi_2] - \Pr[\xi_1 \wedge \xi_2] = \frac{1}{4} + \frac{1}{4} - 0 = \frac{1}{2}$$

Hence, we have that

$$\mathbb{E}[|\operatorname{cut}(S)|] = \sum_{e \in E} 1 \cdot \Pr[e \in \operatorname{cut}(S)] = \frac{|E|}{2} \geq \frac{|\operatorname{cut}(S^*)|}{2}$$

where the last inequality directly follows from the definition of cut of a graph. $\square$

Bibliography

[1]   Sanjeev Arora et al. "Proof verification and the hardness of approximation problems".
      In: *Journal of the ACM* 45.3 (May 1998), 501–555. ISSN: 1557-735X. DOI: 10.1145/
      278298.278306. URL: http://dx.doi.org/10.1145/278298.278306.