Approximation Algorithms

Serge Gaspers

Contents

1 Approximation Algorithms

Optimisation problems

Definition 1. An *optimisation problem* is characterised by

- a set of input instances
- a set of *feasible solutions* for each input instance
- a *value* for each feasible solution

In a *maximisation* problem (resp., a *minimisation*) problem, the goal is to find a feasible solution with maximum (resp., minimum) value.

Example: In the Vertex Cover minimisation problem, the input is a graph G, the feasible solutions are all the vertex covers of G, and the value of a vertex cover is its size.

Approximation algorithm

Definition 2. An *approximation algorithm* A for an optimisation problem Π is a polynomial time algorithm that returns a feasible solution. Denote by $A(I)$ the value of the feasible solution returned by the approximation algorithm A for an instance I and by $\mathrm{OPT}(I)$ the value of the optimum solution. If Π is a minimisation problem, then the *approximation ratio* of A is r if

$$\frac{A(I)}{\mathrm{OPT}(I)} \leq r \text{ for every instance } I.$$

If Π is a maximisation problem, then the *approximation ratio* of A is r if

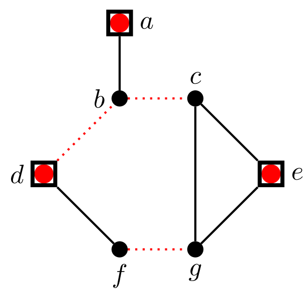
$$\frac{\mathrm{OPT}(I)}{A(I)} \leq r \text{ for every instance } I.$$

We say that A is an r-approximation algorithm if it has approximation ratio r.

2 Multiway Cut

Problem Definition

MULTIWAY CUT	
Input:	A connected graph $G = (V, E)$ and a set of terminals $S = \{s_1, \dots, s_k\}$
Feasible Solution:	A multiway cut, i.e., an edge subset $X \subseteq E$ such that the graph $(V, E \setminus X)$ has no path between any two distinct terminals
Objective:	Minimize the size of the multiway cut.



Complexity

MULTIWAY CUT is NP-complete, even when $k = 3$ (Dahlhaus et al., 1994).
MULTIWAY CUT can be solved in polynomial time when $k = 2$ by a maximum flow algorithm.

Approximation algorithm

Algorithm Greedy-MC

- For each $i \in \{1, \dots, k\}$, compute a smallest edge set C_i, separating s_i from the other terminals. (This can be done by computing a smallest cut between s_i and s_{-i} in the graph obtained from G by merging all the vertices in $S \setminus \{s_i\}$ into a new vertex s_{-i}.)
- Return $\bigcup_{i \in \{1,\dots,k\}} C_i$.

Approximation ratio

Theorem 3 ((Dahlhaus et al., 1994))**.** *Greedy-MC is a 2-approximation algorithm for* MULTIWAY CUT.

Proof. First, note that the algorithm runs in polynomial time. To show that its approximation ratio is at most 2, let us compare the size of the solution it returns, $C = \bigcup_{i \in \{1,\dots,k\}} C_i$, to the size of an optimal solution, A. The graph $(V, E \setminus A)$ has k connected components $G_1, \dots, G_k$, one for each $s_1, \dots, s_k$. Let $A_i \subseteq A$ denote the edges with one endpoint in G_i. Observe that $A = \bigcup A_i$. Since each edge of A is incident to two of the connected components, we have that

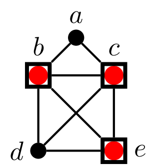
$$2 \cdot |A| = \sum_{i=1}^{k} |A_i| \geq \sum_{i=1}^{k} |C_i| \geq |C|$$

Since $|C| \leq 2 \cdot |A|$, Greedy-MC is a 2-approximation algorithm. □

3 Vertex Cover

Recall: A *vertex cover* of a graph $G = (V, E)$ is a subset of vertices $S \subseteq V$ such that for each edge $\{u, v\} \in E$, we have $u \in S$ or $v \in S$.

VERTEX COVER	
Input:	A graph $G = (V, E)$ and an integer k
Parameter:	k
Question:	Does G have a vertex cover of size at most k?



3.1 Preprocessing

VC-preprocess
Input: A graph G and an integer k.
Output: A graph G' and an integer k' such that G has a vertex cover of size at most k if and only if G' has a vertex cover of size at most k'.

$G' \leftarrow G$
$k' \leftarrow k$
repeat
| Execute simplification rules (Degree-0), (Degree-1), (Large Degree), and (Number of Edges) for (G', k')
until *no simplification rule applies*
return (G', k')

Claim: It is easy to add some bookkeeping to this preprocessing algorithm so that it outputs a set of $k - k'$ vertices such that any vertex cover S' for G' can be extended to a vertex cover for G by adding these $k - k'$ vertices.

Approximation algorithm for Vertex Cover
Since VC-preprocess returns an equivalent instance (G', k') of size $O(k^2)$, we have that

Corollary 4. *The* Vertex Cover *optimisation problem has an approximation algorithm with approximation ratio* $O(\text{OPT})$.

Proof sketch. We start from $k = 0$ and increment k until a solution is returned

- For a given value of k, kernelize.
- If (Number of Edges) does not return No, then return a vertex cover containing all the vertices of the kernelized graph, along with the vertices determined by the bookkeeping of the kernelization procedure.

This procedure returns a vertex cover of size $O(\text{OPT}^2)$. □

Can we obtain a constant approximation ratio?

4 Another kernel / approximation algorithm for Vertex Cover

Integer Linear Program for Vertex Cover
The Vertex Cover problem can be written as an Integer Linear Program (ILP). For an instance $(G = (V, E), k)$ for Vertex Cover with $V = \{v_1, \dots, v_n\}$, create a variable x_i for each vertex v_i, $1 \le i \le n$. Let $X = \{x_1, \dots, x_n\}$.

$$\text{ILP}_{\text{VC}}(G) = \boxed{\begin{array}{ll} \text{Minimize} \displaystyle\sum_{i=1}^{n} x_i & \\ x_i + x_j \ge 1 & \text{for each } \{v_i, v_j\} \in E \\ x_i \in \{0, 1\} & \text{for each } i \in \{1, \dots, n\} \end{array}}$$

Then, (G, k) is a Yes-instance iff the objective value of $\text{ILP}_{\text{VC}}(G)$ is at most k.

Note: Since we just reduced the NP-complete Vertex Cover problem to ILP, we conclude that ILP is NP-hard.

LP relaxation for Vertex Cover

$$\mathrm{LP_{VC}}(G)= \boxed{\begin{array}{ll} \text{Minimize} \sum_{i=1}^{n} x_i & \\ x_i + x_j \geq 1 & \text{for each } \{v_i, v_j\} \in E \\ x_i \geq 0 & \text{for each } i \in \{1, \ldots, n\} \end{array}}$$

Note: the value of an optimal solution for the Linear Program $\mathrm{LP_{VC}}(G)$ is at most the value of an optimal solution for $\mathrm{ILP_{VC}}(G)$

Note 2: Linear Programs (LP) can be solved in polynomial time (Cohen, Lee, and Song, 2019).

Properties of LP optimal solution

- Let $\alpha : X \to \mathbb{R}_{\geq 0}$ be an optimal solution for $\mathrm{LP_{VC}}(G)$. Let

$$\begin{aligned} V_- &= \{v_i : \alpha(x_i) < 1/2\} \\ V_{1/2} &= \{v_i : \alpha(x_i) = 1/2\} \\ V_+ &= \{v_i : \alpha(x_i) > 1/2\} \end{aligned}$$

Lemma 5. *For each $i, 1 \leq i \leq n$, we have that $\alpha(x_i) \leq 1$.*

Lemma 6. *V_- is an independent set.*

Lemma 7. *$N_G(V_-) = V_+$.*

Lemma 8. *For each $S \subseteq V_+$ we have that $|S| \leq |N_G(S) \cap V_-|$.*

Proof. For the sake of contradiction, suppose there is a set $S \subseteq V_+$ such that $|S| > |N_G(S) \cap V_-|$. Let $\epsilon = \min_{v_i \in S}\{\alpha(x_i) - 1/2\}$ and $\alpha' : X \to \mathbb{R}_{\geq 0}$ s.t.

$$\alpha'(x_i) = \begin{cases} \alpha(x_i) & \text{if } v_i \notin S \cup (N_G(S) \cap V_-) \\ \alpha(x_i) - \epsilon & \text{if } v_i \in S \\ \alpha(x_i) + \epsilon & \text{if } v_i \in N_G(S) \cap V_- \end{cases}$$

Note that α' is an improved solution for $\mathrm{LP_{VC}}(G)$, contradicting that α is optimal. ☐

Theorem 9 (Hall's marriage theorem)**.** *A bipartite graph $G = (V \uplus U, E)$ has a matching saturating $S \subseteq V$ if and only if for every subset $W \subseteq S$ we have $|W| \leq |N_G(W)|$.* [1]

Consider the bipartite graph $B = (V_- \uplus V_+, \{\{u, v\} \in E : u \in V_-, v \in V_+\})$.

Lemma 10. *There exists a matching M in B of size $|V_+|$.*

Proof. The lemma follows from the previous lemma and Hall's marriage theorem. ☐

Crown Decomposition: Definition

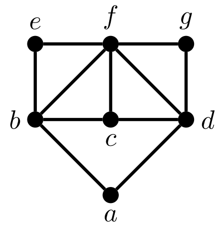
Definition 11 (Crown Decomposition)**.** A crown decomposition (C, H, B) of a graph $G = (V, E)$ is a partition of V into sets C, H, and B such that

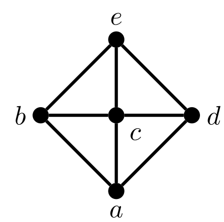
- the crown C is a non-empty independent set,
- the head $H = N_G(C)$,
- the body $B = V \setminus (C \cup H)$, and
- there is a matching of size $|H|$ in $G[H \cup C]$.

By the previous lemmas, we obtain a crown decomposition $(V_-, V_+, V_{1/2})$ of G if $V_- \neq \emptyset$.

[1] A *matching* M in a graph G is a set of edges such that no two edges in M have a common endpoint. A matching *saturates* a set of vertices S if each vertex in S is an end point of an edge in M.

Crown Decomposition: Examples





crown decomposition
$(\{a, e, g\}, \{b, d, f\}, \{c\})$

has no crown decomposition

Using the crown decomposition

Lemma 12. *Suppose that $G = (V, E)$ has a crown decomposition (C, H, B). Then,*

$$vc(G) \leq k \quad \Leftrightarrow \quad vc(G[B]) \leq k - |H|,$$

where $vc(G)$ denotes the size of the smallest vertex cover of G.

Proof. ($\Rightarrow$): Let S be a vertex cover of G with $|S| \leq k$. Since S contains at least one vertex for each edge of a matching, $|S \cap (C \cup H)| \geq |H|$. Therefore, $S \cap B$ is a vertex cover for $G[B]$ of size at most $k - |H|$.

($\Leftarrow$): Let S be a vertex cover of $G[B]$ with $|S| \leq k - |H|$. Then, $S \cup H$ is a vertex cover of G of size at most k, since each edge that is in G but not in $G[B]$ is incident to a vertex in H. □

Nemhauser-Trotter

Corollary 13 ((Nemhauser and Trotter Jr., 1974))**.** *There exists a smallest vertex cover S of G such that $S \cap V_- = \emptyset$ and $V_+ \subseteq S$.*

Corollary 14 ((Nemhauser and Trotter Jr., 1974))**.** Vertex Cover *has a 2-approximation algorithm.*

Crown reduction

(Crown Reduction)
If solving $\text{LP}_{VC}(G)$ gives an optimal solution with $V_- \neq \emptyset$, then return $(G - (V_- \cup V_+), k - |V_+|)$.

(Number of Vertices)
If solving $\text{LP}_{VC}(G)$ gives an optimal solution with $V_- = \emptyset$ and $|V| > 2k$, then return No.

Lemma 15. *(Crown Reduction) and (Number of Vertices) are sound.*

Proof. (Crown Reduction) is sound by previous Lemmas. Let α be an optimal solution for $\text{LP}_{VC}(G)$ and suppose $V_- = \emptyset$. The value of this solution is at least $|V|/2$. Thus, the value of an optimal solution for $\text{ILP}_{\text{VC}}(G)$ is at least $|V|/2$. Since G has no vertex cover of size less than $|V|/2$, we have a No-instance if $k < |V|/2$. □

Linear vertex-kernel for Vertex Cover

Theorem 16. Vertex Cover *has a kernel with $2k$ vertices and $O(k^2)$ edges.*

This is the smallest known kernel for Vertex Cover. See `http://fpt.wikidot.com/fpt-races` for the current smallest kernels for various problems.

5 More on Crown Decompositions

Crown Lemma

Lemma 17 (Crown Lemma)**.** *Let $G = (V, E)$ be a graph without isolated vertices and with $|V| \geq 3k+1$. There is a polynomial time algorithm that either*

- *finds a matching of size $k+1$ in G, or*
- *finds a crown decomposition of G.*

To prove the lemma, we need Kőnig's Theorem

Theorem 18 ((Kőnig, 1931))**.** *In every bipartite graph the size of a maximum matching is equal to the size of a minimum vertex cover.*

Proof of the Crown Lemma. Compute a maximum matching M of G. If $|M| \geq k+1$, we are done. Note that $I := V \setminus V(M)$ is an independent set with $|V| - |V(M)| \geq k+1$ vertices. Consider the bipartite graph B formed by edges with one endpoint in $V(M)$ and the other in I. Compute a minimum vertex cover X and a maximum matching M' of B. We know: $|X| = |M'| \leq |M| \leq k$. Hence, $X \cap V(M) \neq \emptyset$. Let $M^* = \{e \in M' : e \cap (X \cap V(M)) \neq \emptyset\}$. We obtain a crown decomposition with

- crown $C = V(M^*) \cap I$
- head $H = X \cap V(M) = X \cap V(M^*)$, and
- body $B = V \setminus (C \cup H)$.

As an exercise, verify that (C, H, B) is indeed a crown decomposition. □

After computing a kernel ...

- ... we can use any algorithm to compute an actual solution.
- Brute-force, faster exponential-time algorithms, parameterized algorithms, often also approximation algorithms

Kernels

- A parameterized problem may not have a kernelization algorithm
  - Example, Coloring[2] parameterized by k has no kernelization algorithm unless P = NP.
  - A kernelization would lead to a polynomial time algorithm for the NP-complete 3-Coloring problem
- Only exponential kernels may be known for a parameterized problem
- There is a theory of kernel lower bounds, establishing exponential lower bounds on the kernel size of certain parameterized problems.

Approximation algorithms

Besides constant factor approximation algorithms, positive results include:

- additive approximation (rare)
- polynomial time approximation schemes (PTAS): able to achive an approximation ratio $1+\epsilon$ for any constant ϵ in polynomial time, but the running time depends on $1/\epsilon$. Restrictions include EPTAS (Efficient PTAS) and FPTAS (Fully PTAS), restricting how the running time may depend on the parameter $1/\epsilon$.

Negative results include

- no factor-c approximation algorithm unless P = NP / unless the Unique Games conjecture fails, etc.
- APX-hardness, ruling out PTASs

[2] Can one color the vertices of an input graph G with k colors such that no two adjacent vertices receive the same color?

6 Further Reading

- Vazirani's textbook (Vazirani, 2003)
- Fellows et al.'s survey on Vertex Cover kernelization (Fellows et al., 2018)

References

Michael B. Cohen, Yin Tat Lee, and Zhao Song (2019). "Solving linear programs in the current matrix multiplication time". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*. ACM, pp. 938–942.

Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis (1994). "The Complexity of Multiterminal Cuts". In: *SIAM Journal on Computing* 23.4, pp. 864–894. DOI: 10.1017/9781107415157.

Michael R. Fellows, Lars Jaffke, Aliz Izabella Király, Frances A. Rosamond, and Mathias Weller (2018). "What Is Known About Vertex Cover Kernelization?" In: *Adventures Between Lower Bounds and Higher Altitudes: Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday*. Ed. by Hans-Joachim Böckenhauer, Dennis Komm, and Walter Unger. Springer, pp. 330–356.

Dénes Kőnig (1931). "Gráfok és mátrixok". In: *Matematikai és Fizikai Lapok* 38, pp. 116–119.

George L. Nemhauser and Leslie E. Trotter Jr. (1974). "Properties of vertex packing and independence system polyhedra". In: *Math. Program.* 6.1, pp. 48–61.

Vijay V. Vazirani (2003). *Approximation Algorithms*. Springer.