CS 4980: Capstone Research Notes

Professor Mark Floryan,
compiled by Grady Hollar

Fall 2025

X. Basic Definitions

1 Basic Definitions

What is an approximation algorithm?

1.1 Min-max relations

X. Some basic examples

Vertex cover, set cover.

2 Randomization and LP Methods

2.1 A simple MAX-3SAT algorithm

(we assume each clause contains exactly 3 distinct variables + no clause contains both a variable and its negation (you can relax this second one tho?))

Definition (Randomized Approximation Algorithm). We say that a randomized algorithm for a problem has an *approximation ratio* of $\rho(n)$ if, for any input of size n, the expected cost of the solution produced by the randomized algorithm is within a factor of $\rho(n)$ of the cost of an optimal solution.

Optimization Problem (MAX-3SAT) Given a 3-CNF formula ϕ, find an assignment of ϕ that satisfies the largest number of clauses.

Algorithm 1 Approx-MAX-3SAT(ϕ)

---

for $x_i \in \phi$ **do**
  Assign x_i randomly to be 0 or 1
end for
return the assignment

Theorem 2.1. Algorithm 1 is an 8/7-approximation for MAX-3SAT.

Proof. Let ϕ be a 3-CNF formula with n variables $x_1, x_2, \ldots, x_n$ and m clauses. For $1 \leq i \leq m$, define the random variable

$$Y_i = \begin{cases} 1, & \text{clause } i \text{ is satisfied} \\ 0, & \text{otherwise} \end{cases}.$$

Then, the number of clauses satisfied overall is modeled by the random variable defined by

$$Y = \sum_{i=1}^{m} Y_i.$$

So, it suffices to compute $E[Y]$. Since each variable is set to 1 with probability 1/2 and 0 with probability 1/2, and a clause is not satisfied only if all three of its literals are set to 0, we have

$$P[Y_i = 0] = (1/2)^3 = 1/8,$$

so that

$$P[Y_i = 1] = 1 - 1/8 = 7/8.$$

Computing $E[Y_i]$ then gives us

$$E[Y_i] = 1 \cdot 7/8 + 0 \cdot 1/8 = 7/8.$$

We may now compute

$$\begin{aligned} E[Y] &= E\left[\sum_{i=1}^{m} Y_i\right] \\ &= \sum_{i=1}^{m} E[Y_i] \\ &= \sum_{i=1}^{m} 7/8 \\ &= 7m/8. \end{aligned}$$

Since the maximum amount of clauses that can be satisfied in ϕ is m, the approximation ratio is at most $m/(7m/8) = 8/7$. $\square$

2.2 The weighted vertex cover problem

Optimization Problem (Minimum Weight Vertex Cover) Given an undirected graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a vertex cover $C \subseteq V$ of minimum weight $w(C) = \sum_{v \in C} w(v)$.

Consider the following linear program:

minimize

$$\sum_{v \in V} w(v) \cdot x_v$$

subject to

$$\begin{aligned} x_u + x_v &\geq 1 \quad \forall (u, v) \in E \\ x_v &\leq 1 \quad \forall v \in V \\ x_v &\geq 0 \quad \forall v \in V \end{aligned}$$

Algorithm 2 Approx-Min-Weight-VC(G, w)

```
C = ∅
compute an optimal solution x̄ to the above linear program.
for v ∈ V do
    if x̄_v ≥ 1/2 then
        C = C ∪ {v}
    end if
end for
return C
```

Theorem 2.2. Algorithm 2 is a 2-approximation algorithm for the minimum weight vertex cover problem.

Proof. Want to show:

Why is C a cover?

Compare an optimal cover C^* to the objective function value for optimal solution of the LP z^*. Obtain $z^* \leq w(C^*)$. (C^* is a feasible solution to the LP)

Obtain $z^* \geq w(C)/2$. Key step:

$$z^* \geq \sum_{v \in V : \overline{x}_v \geq 1/2} w(v) \cdot \overline{x}_v$$

Combine inequalities to get $w(C) \leq 2w(C^*)$.

□