- 教材讨论
  - JH第5章第3节第6、7小节

问题1：RSMS

- RSMS要解决的问题是什么？
- 基本思路是什么？
- 近似比是多少？为什么？
- 对于不同的输入，效果的优劣分别如何？为什么？

Algorithm 5.3.6.1. RSMS (Random Sampling for Max-Sat)

Input: A Boolean formula Φ over the set of variables $\{x_1, \ldots, x_n\}$, $n \in \mathbb{N}$.
Step 1: Choose uniformly at random $\alpha_1, \ldots, \alpha_n \in \{0, 1\}$.
Step 2: **output**$(\alpha_1, \ldots, \alpha_n)$.
Output: an assignment to $\{x_1, \ldots, x_n\}$.

问题2：RRRMS

- RRRMS的基本思路是什么？
- MAX-SAT是如何被先后规约为ILP和LP的？
- 引理5.3.6.3证明的基本过程是什么？
- 对于不同的输入，效果的优劣分别如何？为什么？

Algorithm 5.3.6.2. RRRMS (RELAXATION WITH RANDOM ROUNDING FOR MAX-SAT)

Input: A formula $\Phi = F_1 \wedge F_2 \wedge \cdots \wedge F_m$ over $X = \{x_1, \ldots, x_n\}$ in CNF, $n, m \in \mathbb{N}$.

Step 1: Formulate the MAX-SAT problem for Φ as the integer linear program $\mathrm{LP}(\Phi)$ maximizing $\sum_{j=1}^{m} z_j$ by the constraints (5.19) and (5.20).

Step 2: Solve the relaxed version of $\mathrm{LP}(\Phi)$ according to (5.21). Let $\alpha(z_1)$, $\alpha(z_2), \ldots, \alpha(z_m), \alpha(y_1), \ldots, \alpha(y_n) \in [0, 1]$ be an optimal solution of the relaxed $\mathrm{LP}(\Phi)$.

Step 3: Choose n values $\gamma_1, \ldots, \gamma_n$ uniformly at random from $[0, 1]$.
for $i = 1$ **to** n **do**
  if $\gamma_i \in [0, \alpha(y_i)]$ **then set** $x_i = 1$
  else set $x_i = 0$
{Observe that Step 3 realizes the random choice of the value 1 for x_i with the probability $\alpha(y_i)$.}

Output: An assignment to X.

问题2：RRRMS (续)

$$\text{maximize} \sum_{j=1}^{m} z_j$$

$$\text{subject to} \sum_{i \in In^+(F_j)} y_i + \sum_{i \in In^-(F_j)} (1 - y_i) \geq z_j \ \forall j \in \{1, \ldots, m\} \quad (5.19)$$

$$\text{where } y_i, z_j \in \{0, 1\} \text{ for all } i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}. \quad (5.20)$$

Lemma 5.3.6.3. *Let k be a positive integer, and let F_j be a clause of Φ with k literals. Let $\alpha(y_1), \ldots, \alpha(y_n), \alpha(z_1), \ldots, \alpha(z_m)$ be the solution of* LP(Φ) *by* RRRMS. *The probability that the assignment computed by the algorithm* RRRMS *satisfies F_j is at least*

$$\left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot \alpha(z_j).$$

Proof. Since one considers the clause F_j independently from other clauses, one can assume without loss of generality that it contains only uncomplemented variables and that it is of the form $x_1 \vee x_2 \vee \cdots \vee x_k$. By the constraint (5.19) of LP(Φ) we have

$$y_1 + y_2 + \cdots + y_k \geq z_j. \tag{5.23}$$

The clause F_j remains unsatisfied if and only if all of the variables $y_1, y_2, \ldots, y_k$ are set to zero. Following Step 3 of RRRMS and the fact that each variable is rounded independently, this occurs with probability

$$\prod_{i=1}^{k} (1 - \alpha(y_i)).$$

So, F_j is satisfied by the output of RRRMS with probability

$$1 - \prod_{i=1}^{k}(1 - \alpha(y_i)). \tag{5.24}$$

Under the constraint (5.23), (5.24) is minimized when $\alpha(y_i) = \alpha(z_j)/k$ for all $i = 1, \ldots, k$. Thus,

$$Prob(F_j \text{ is satisfied}) \geq 1 - \prod_{i=1}^{k}(1 - \alpha(z_j)/k). \tag{5.25}$$

To complete the proof it suffices to show, for every positive integer k, that

$$f(r) = 1 - (1 - r/k)^k \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot r = g(r) \tag{5.26}$$

for every $r \in [0, 1]$ (and so for every $\alpha(z_j)$). Since f is a concave function in r, and g is a linear function in r (Fig. 5.4), it suffices to verify the inequality at the endpoints $r = 0$ and $r = 1$. Since $f(0) = 0 = g(0)$ and $f(1) = 1 - (1 - 1/k)^k = g(1)$, the inequality (5.26) holds. Setting $r = \alpha(z_j)$ in (5.26) and inserting (5.26) into (5.25) the proof is done. □

问题2：RRRMS (续)

	一般情况（特别是短句子）	长句子
RSMS	2	$2^k/(2^k-1)$
RRRMS	$e/(e-1)$	$\frac{k^k}{k^k-(k-1)^k}$

问题3：Schoening算法

- Schoening算法要解决的问题是什么？
- 基本思路是什么？时间复杂度是多少？
- 为什么是单边Monte Carlo算法？

Algorithm 5.3.7.1. SCHÖNING'S ALGORITHM

Input: A formula F in 3CNF over a set of n Boolean variables.
Step 1: $K := 0$;
$UPPER := \lceil 20 \cdot \sqrt{3\pi n} \cdot \left(\frac{4}{3}\right)^n \rceil$
$S := FALSE$.
Step 2: **while** $K < UPPER$ and $S := FALSE$ **do**
  begin $K := K + 1$;
    Generate uniformly at random an assignment $\alpha \in \{0,1\}^n$;
    if F is satisfied by α **then** $S := TRUE$;
    $M := 0$;
    while $M < 3n$ and $S = FALSE$ **do**
      begin $M := M + 1$;
        Find a clause C that is not satisfied by α;
        Pick one of the literals of C at random, and flip its value in order to get a new assignment α;
        if F is satisfied by α **then** $S := TRUE$
    end
  end
Step 3: **if** $S = TRUE$ **output** "F is satisfiable"
else output "F is not satisfiable".

问题3：Schoening算法 (续)

Now we analyze the failure probability of SCHÖNING'S ALGORITHM for a given formula F in 3CNF. If F is not satisfiable, then the algorithm outputs the right answer with certaincy.

Now consider that F is satisfiable. Let α^* be an assignment that satisfies F. Let p be the probability that one local search procedure that executes at most $3n$ local steps from a random assignment generates α^*. Obviously, p is a lower bound on finding an assignment that satisfies F in one run of the local search procedure (the inner cycle **while** in Step 2). The crucial point of this analysis is to show that

$$p \geq \frac{1}{2\sqrt{3 \cdot \pi \cdot n}} \cdot \left(\frac{3}{4}\right)^n . \tag{5.32}$$

The main idea behind is that the number $UPPER \gg p$ of independent attempts is sufficient to increase the probability of success to $1 - e^{-10}$.

In what follows we consider the distance between two assignments α and β as the number of bits in which α and β differ (i.e., the number of flips that the local search needs to move from α to β). Now, partition all assignments from $\{0,1\}^n$ into $n+1$ classes

$$Class(j) = \{\beta \in \{0,1\}^n \mid distance(\alpha^*, \beta) = j\}$$

according to their distance j to α^* for $j = 0, 1, \ldots, n$. Obviously, $|Class(j)| = \binom{n}{j}$, and the probability to uniformly generate an assignment from $Class(j)$ at random is exactly

$$p_j = \binom{n}{j} \Big/ 2^n. \tag{5.33}$$

问题3：Schoening算法 (续)

Now let us analyze the behavior of the local search. If $\alpha \in Class(j)$ does not satisfy F, then there exists at least one clause C that is not satisfied by α. Since α^* satisfies C, there exists a variable that occurs in C and whose flipping results in a $\beta \in Class(j-1)$. Thus, there exists a possibility to get an assignment β with a smaller distance to α than $distance(\alpha^*, \alpha)$. Since C consists of at most three literals and the algorithm chooses one of them randomly, we have a probability of at least 1/3 "to move in the direction" to α^* (to decrease the distance to α^* by 1) in one step (i.e., we have a probability of at most 2/3 to increase the distance to α^* by 1 in one step). Let, for all i, j, $i \le j \le n$, $q_{j,i}$ denote the probability to reach α^* from an $\alpha \in Class(j)$ by $j+i$ moves in the direction to α^* and i moves from the direction of α^* (i.e., in overall $j+2i$ steps). Then

$$q_{j,i} = \binom{j+2i}{i} \cdot \frac{j}{j+2i} \cdot \left(\frac{1}{3}\right)^{j+i} \cdot \left(\frac{2}{3}\right)^{i}$$

can be established by a short combinatorial calculation.[71] Obviously, the probability q_j to reach α^* from an $\alpha \in Class(j)$ is at least $\sum_{i=0}^{j} q_{j,i}$. Observe that SCHÖNING'S ALGORITHM allows $3n$ steps and so $j+2i$ steps can be executed for all $j \in \{0, 1, \ldots, n\}$ and $i \le j$. Thus,

$$\begin{aligned} q_j &\ge \sum_{i=0}^{j} \left[\binom{j+2i}{i} \cdot \frac{j}{j+2i} \cdot \left(\frac{1}{3}\right)^{j+i} \cdot \left(\frac{2}{3}\right)^{i} \right] \\ &\ge \frac{1}{3} \sum_{i=0}^{j} \left[\binom{j+2i}{i} \cdot \left(\frac{1}{3}\right)^{j+i} \cdot \left(\frac{2}{3}\right)^{i} \right] \\ &> \frac{1}{3} \binom{3j}{j} \cdot \left(\frac{1}{3}\right)^{2j} \cdot \left(\frac{2}{3}\right)^{j}. \end{aligned}$$

[71] Note that $\binom{j+2i}{i} \cdot \frac{j}{j+2i}$ is the number of different paths from $\alpha \in Class(j)$ to α^* where a path is determined by a word over the two-letter alphabet $\{+, -\}$ where $+$ means a movement in the direction to α^* and $-$ means a movement that increases the distance to α^*. Every such word must satisfy that each suffix of w contains more $+$ symbols than $-$ symbols.

问题3：Schoening算法 (续)

Now we are ready to perform the final calculation for p. Clearly,

$$p \geq \sum_{j=0}^{n} p_j \cdot q_j.$$

问题3：Schoening算法 (续)

- 你理解这句总结了吗？

The crucial point is that the probability of success in one attempt is at least $1/Exp(n)$, where $Exp(n)$ is an exponential function that grows substantially slower than 2^n. Thus, performing $O(Exp(n))$ random attempts one can find a satisfying assignment with a probability almost 1 in time $O(|F| \cdot n \cdot Exp(n))$.