计算机问题求解 – 论题4-16

- 不同方法的结合

课程研讨

- 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？
- 你能解释引理5.3.6.3的证明吗？
- 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 LP(Φ) maximizing $\sum_{j=1}^{m} z_j$ by the constraints (5.19) and (5.20).

Step 2: Solve the relaxed version of LP(Φ) 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 LP(Φ).

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)

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

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

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

基于严格约束的关联决策协同圆整

- 决策变量圆整之后，需要满足某些严格约束
  - 如：决策变量加权和必须大于阈值。
- 关联协同圆整法通过两两决策变量间的“此消彼长”，不断将其中一个变量的实数值转移到另一个变量
- 精心设计的协同圆整法能在满足约束的条件下，保持原有决策变量圆整为1的期望与输入值一致。

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. □

RRRMS

Theorem 5.3.6.4. *The algorithm* RRRMS *is a polynomial-time randomized* $(e/(e-1))$*-expected approximation algorithm for* MAX-SAT *and a polynomial time randomized* $(k^k/(k^k-(k-1)^k))$*-expected approximation algorithm for* MAX-EkSAT.

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	$e/(e-1)$	$\frac{k^k}{k^k-(k-1)^k}$

D&C Algorithm for 3Sat

Algorithm 3.5.2.1 (D&C-3SAT(F)).

Input: A formula F in 3CNF.

Step 1: **if** $F \in 3CNF(3, k)$ or $F \in 3CNF(m, 2)$ for some $m, k \in \mathbb{N} - \{0\}$,
then decide whether $F \in$ 3SAT or not by testing all assignments to the variables of F;
if $F \in$ 3SAT **output**(1) **else output**(0).

Step 2: Let H be one of the shortest clauses of F.
if $H = (l)$ **then output**(D&C-3SAT($F(l = 1)$));
if $H = (l_1 \vee l_2)$
then output(D&C-3SAT($F(l_1 = 1)$)
$\vee$D&C-3SAT($F(l_1 = 0, l_2 = 1)$));
if $H = (l_1 \vee l_2 \vee l_3)$
then output(D&C-3SAT($F(l_1 = 1)$)
$\vee$D&C-3SAT($F(l_1 = 0, l_2 = 1)$)
$\vee$D&C-3SAT($F(l_1 = 0, l_2 = 0, l_3 = 1)$)).

$Time_{\text{D\&C-3SAT}}(F) = O(r \cdot 1.84^n)$

3CNF(n, r) $= \{\Phi \mid \Phi$ is a formula over at most n variables in 3CNF and Φ contains at most r clauses$\}$.

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？ $O(|F| \cdot n^{3/2} \cdot (4/3)^n)$
- 为什么是单边Monte Carlo算法？你能解释它的证明吗？

Algorithm 5.3.7.1. SCHÖNING'S ALGORITHM

Input: A formula F in 3CNF over a set of n Boolean variables.
Step 1: $K := 0$;
$UPPER := \lceil 20 \cdot \sqrt{3\pi n} \cdot (\frac{4}{3})^n \rceil$
$S := FALSE$.
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)

- 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 $O(|F| \cdot n \cdot Exp(n))$.

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