Capítulo 6

Algoritmos Probabilísticos

Neste capítulo vamos supor que temos à nossa disposição um gerador de *bits* aleatórios (*random bits generator*), ou seja, um algoritmo ideal Rand que recebe um número racional ρ no intervalo fechado $[0, 1]$, devolve 1 com probabilidade ρ e 0 com probabilidade $1 - \rho$. Existem implementações aproximadas do algoritmo Rand (veja Knuth [Knu98]).

Rand

Um algoritmo que utiliza Rand é chamado de **probabilístico** (*randomized*). O comportamento de um algoritmo probabilístico depende não apenas da instância do problema mas também dos números devolvidos por Rand. Dizemos que um algoritmo probabilístico é **polinomial** se existe um polinômio p tal que, para toda instância I do problema, o número de chamadas ao algoritmo Rand e o consumo de tempo das demais operações são limitados por $p(\langle I \rangle)$.

Valor esperado ou Esperança

Vamos generalizar a definição de algoritmos de aproximação (seção 1.2) para englobar algoritmos probabilísticos. Considere um algoritmo probabilístico A para um problema de otimização. Para qualquer instância I do problema, seja X_I a variável aleatória (apêndice D) cujo valor é o valor da solução produzida por A para esta instância[1]. Dizemos que A é uma **α-aproximação probabilística** para o problema em questão se $\mathbf{E}[X_I] \geq \alpha \operatorname{opt}(I)$ no caso de problema de maximização e $\mathbf{E}[X_I] \leq \alpha \operatorname{opt}(I)$ no caso de problema de minimização. No presente capítulo, α é um número que não depende de I, embora em geral α possa

Algoritmo probabilístico aproximativo

[1] Os algoritmos abordados neste capítulo, para cada instância do problema, provocam um número fixo t de chamadas de Rand, que não depende dos valores devolvidos por Rand, sendo que a i-ésima chamada de Rand tem como parâmetro sempre um mesmo número, digamos ρ_i. Denotando por Ω o conjunto $\{0, 1\}$, seja $\mathcal{P}_i$ a medida de probabilidade sobre Ω dada por $\mathcal{P}_i(1) = \rho_i$ e $\mathcal{P}_i(0) = 1 - \rho_i$. O espaço de probabilidade no qual X_I está definida neste caso é o espaço produto $(\Omega, \mathcal{P}_1) \times \cdots \times (\Omega, \mathcal{P}_t)$.

ser uma função de I. Dizemos que α é uma **razão de aproximação esperada** de A.

Vamos mostrar que, em certas condições, uma α-aproximação probabilística A pode ser usada para produzir, com alta probabilidade, soluções aproximadas do problema. Suponha que o problema é de minimização e que a variável aleatória X_I não assume valores negativos (esse é o caso na grande maioria dos problemas combinatórios). Para cada ε positivo, a desigualdade de Markov (lema D.1, apêndice D) garante que

$$\mathbf{Pr}[X_I \geq (\alpha+\varepsilon)\mathrm{opt}(I)] \leq \frac{\mathbf{E}[X_I]}{(\alpha+\varepsilon)\mathrm{opt}(I)} \leq \frac{\alpha\,\mathrm{opt}(I)}{(\alpha+\varepsilon)\mathrm{opt}(I)} = \frac{\alpha}{\alpha+\varepsilon}\,.$$

Para um dado λ no intervalo aberto $(0,1)$, seja $k := \lceil \log_\beta \lambda \rceil$, onde $\beta = \frac{\alpha}{\alpha+\varepsilon}$. Suponha que A é aplicado k vezes à instância I e escolha o resultado de menor valor. Se Y_I é esse menor valor[2], então $\mathbf{Pr}[Y_I \geq (\alpha+\varepsilon)\mathrm{opt}(I)] \leq \lambda$. Para λ pequeno, este esquema produz, portanto, com probabilidade alta (pelo menos $1-\lambda$), uma solução viável com valor menor que $(\alpha+\varepsilon)\mathrm{opt}(I)$.

A probabilidade das k aplicações serem > (alpha+eps)*opt é pequena: <= beta ^k

O mesmo raciocínio pode ser aplicado a problemas de maximização, desde que se conheça uma delimitação superior para o valor ótimo do problema.

6.1 Satisfatibilidade máxima

Vamos usar o problema da satisfatibilidade máxima para exemplificar algoritmos probabilísticos. Para definir o problema, precisamos introduzir alguma notação.

Suponha dado um conjunto finito V de objetos que chamamos **variáveis**. Uma **cláusula booleana** sobre V, ou simplesmente **cláusula**, é um par ordenado de subconjuntos de V disjuntos, um dos quais pelo menos não é vazio. Se C é uma cláusula, denotamos o primeiro componente de C por C_1 e o segundo por C_0. Em terminologia tradicional, C_0 é o conjunto das variáveis "complementadas" e C_1 é o conjunto das variáveis "não-complementadas". O **número de variáveis** em uma cláusula C é $|C_1| + |C_0|$.

[2] Y_I é uma variável aleatória. Denotando por $(\Omega', \mathcal{P}')$ o espaço de probabilidade de X_I, o espaço no qual Y_I está definida é o produto $(\Omega', \mathcal{P}') \times \cdots \times (\Omega', \mathcal{P}')$ de k cópias de $(\Omega', \mathcal{P}')$.

Uma **valoração** das variáveis de V é um vetor x indexado por V com valores em $\{0,1\}$. Uma valoração x **satisfaz** uma cláusula C se $x_v = 1$ para algum v em C_1 ou $x_v = 0$ para algum v em C_0.

Vejamos um exemplo. Suponha que $V = \{a,b,c,d\}$ e sejam C, D e E as cláusulas $(\{a,d\},\{b\})$, $(\{c\},\{a,b\})$ e $(\emptyset,\{b,c,d\})$. Na notação tradicional, a coleção $\{C,D,E\}$ seria denotada por $\{a \vee d \vee \bar{b}, c \vee \bar{a} \vee \bar{b}, \bar{b} \vee \bar{c} \vee \bar{d}\}$.

O **problema da satisfatibilidade máxima** (*maximum satisfiability problem*), denotado pela sigla MAXSAT, consiste no seguinte:

> **Problema** MAXSAT $(V, \mathcal{C})$: *Dada uma coleção $\mathcal{C}$ de cláusulas sobre um conjunto V de variáveis, encontrar uma valoração x de V que satisfaça o maior número possível de cláusulas de $\mathcal{C}$.*

O problema é NP-difícil mesmo quando cada cláusula em $\mathcal{C}$ tem duas variáveis [GJ79].

O problema 2SAT de decisão é polinomial, mas o problema de maximização Max2SAT é NP-Difícil

Algoritmo de Johnson

Eis um algoritmo probabilístico para o problema MAXSAT. O algoritmo é atribuído a Johnson [Joh74]. Ele é tão simples que ignora $\mathcal{C}$:

Algoritmo MAXSAT-JOHNSON (V)

```
1  para cada v em V faça
2      ẋ_v ← RAND(1/2)
3  devolva ẋ
```

Verdadeiro, com probabilidade 1/2

No algoritmo acima, o número de chamadas a RAND e o consumo de tempo das demais operações é $O(|V|)$. Seja $X_{\mathcal{C}}$ a variável aleatória cujo valor é o número de cláusulas de $\mathcal{C}$ satisfeitas por uma valoração produzida por MAXSAT-JOHNSON (V). O espaço de probabilidade de $X_{\mathcal{C}}$ é o espaço produto $(\Omega,\mathcal{P}) \times \cdots \times (\Omega,\mathcal{P})$, onde $(\Omega,\mathcal{P})$ aparece $|V|$ vezes no produto, Ω é o conjunto $\{0,1\}$ e $\mathcal{P}$ é a medida de probabilidade sobre Ω dada por $\mathcal{P}(1) = \mathcal{P}(0) = \frac{1}{2}$. $X_{\mathcal{C}}$

pelo menos k variáveis em cada cláusula

> **Teorema 6.1**: *Se $(V,\mathcal{C})$ é uma instância do problema* MAXSAT *na qual cada cláusula tem pelo menos k variáveis e $X_{\mathcal{C}}$ é a variável aleatória acima definida, então*
>
> $$\mathbf{E}[X_{\mathcal{C}}] \;\geq\; \left(1 - \frac{1}{2^k}\right) \mathrm{opt}(V,\mathcal{C})\,.$$

Demonstração: Para cada cláusula C, defina uma variável aleatória Z_C da seguinte maneira: $Z_C := 1$ se a valoração produzida pelo algoritmo satisfaz C e $Z_C := 0$ em caso contrário. Como C tem pelo menos k variáveis e no algoritmo $\dot{x}_v$ é 0 com probabilidade $1/2$, a probabilidade do evento $[Z_C{=}0]$ é no máximo $1/2^k$ e a probabilidade do evento $[Z_C{=}1]$ é pelo menos $1-1/2^k$. Portanto, como $X_{\mathcal{C}}$ é a soma das variáveis aleatórias Z_C,

$$\mathbf{E}[X_{\mathcal{C}}] \;=\; \mathbf{E}[\textstyle\sum_C Z_C] \;=\; \sum_C \mathbf{E}[Z_C] \;\geq\; (1-\frac{1}{2^k})|\mathcal{C}|$$

e concluímos a prova do teorema, já que $\mathrm{opt}(V,\mathcal{C}) \leq |\mathcal{C}|$. □

Como todas as cláusulas em $\mathcal{C}$ têm pelo menos uma variável, podemos aplicar o teorema acima a $(V,\mathcal{C})$ com $k = 1$ e concluir o seguinte resultado.

Teorema 6.2: *O algoritmo* MAXSAT-JOHNSON *é uma 0,5-aproximação probabilística polinomial para o* MAXSAT.

Seria melhor sem Cláusulas de tamanho 1. Melhoria: assumir que não há duas cláusulas de tam. 1 com X e seu complem. Se um literal está numa cláusula de tam. 1, atribua V com prob. h>1/2. Para demais variáveis atribua V com prob. h>1/2.

Cláusula C qualquer. Se tam. 1, prob. ser V é h. Se tam. k>1, a prob. de ser V é 1-prob de ser F, que é 1-prob. de todos seus literais F. Como h>1-h, essa prob. é >= 1-h^k >= 1-h^2.

Tomando h tal que h=1-h^2, temos: h = 1-h^2 = 0,618

Algoritmo do arredondamento probabilístico

Descrevemos a seguir um algoritmo de aproximação para o problema MAXSAT que envolve o "arredondamento probabilístico" (*randomized rounding*) de uma solução ótima de um programa linear. O **arredondamento probabilístico** de um número ρ do intervalo aberto $(0,1)$ consiste em arredondar ρ "para cima" com probabilidade ρ e "para baixo" com probabilidade $1-\rho$.

Relaxação de um modelo de Programação Inteira exato para MaxSAT

Considere o seguinte problema de programação linear: dada uma coleção $\mathcal{C}$ de cláusulas sobre um conjunto V, encontrar um par (x,z) de vetores, com x indexado por V e z indexado por $\mathcal{C}$, que

Maximizar o número de cláusulas satisfeitas

$$\begin{aligned} &\text{maximize} && \textstyle\sum_{C\in\mathcal{C}} z_C \\ &\text{sob as restrições} \\ &\textstyle\sum_{v\in C_0}(1-x_v) + \sum_{v\in C_1} x_v \;\geq\; z_C && \text{para cada } C \text{ em } \mathcal{C}, \\ &0 \;\leq\; z_C \;\leq\; 1 && \text{para cada } C \text{ em } \mathcal{C}, \\ &0 \;\leq\; x_v \;\leq\; 1 && \text{para cada } v \text{ em } V. \end{aligned} \tag{6.1}$$

Se a cláusula é satisfeita, a soma dos valores de seus literais é >=1

z_C = 1 (cláusula satisfeita)
z_C = 0 (cláusula não sat.)
x_v = 1 (variável com valor V)
x_v = 0 (variável com valor F)

Note a seguinte relação entre este programa linear e o problema MAXSAT. Suponha que temos uma solução ótima x^* do MAXSAT $(V,\mathcal{C})$. Para cada cláusula C em $\mathcal{C}$, faça $z^*_C := 1$ se x^* satisfaz C e $z^*_C := 0$ em caso

contrário. É claro então que o par (x^*, z^*) é uma solução viável de (6.1) e o valor dessa solução é igual ao número de cláusulas de $\mathcal{C}$ satisfeitas por x^*. Portanto, o programa linear (6.1) é uma relaxação do problema MAXSAT $(V, \mathcal{C})$ e

$$\sum_C \hat{z}_C \geq \text{opt}(V, \mathcal{C}) \,, \tag{6.2}$$

para qualquer solução ótima $(\hat{x}, \hat{z})$ de (6.1).

Sendo uma relaxação, sem a restrição de valores inteiros, o valor retornado será maior

O seguinte algoritmo de arredondamento probabilístico, desenvolvido por Goemans e Williamson [GW94], é um algoritmo de aproximação para o MAXSAT.

Algoritmo MAXSAT-GW $(V, \mathcal{C})$
1 seja $(\hat{x}, \hat{z})$ uma solução ótima racional de (6.1)
2 para cada v em V faça
3 $\quad \dot{x}_v \leftarrow \text{RAND}\,(\hat{x}_v)$
4 devolva $\dot{x}$

Obtido da Programação Linear, com valores racionais

Verdadeiro, com probabilidade x_v

O algoritmo acima pode ser implementado de modo que a execução da linha 1 consuma tempo limitado por um polinômio em $|V|$ e $|\mathcal{C}|$, e produza uma solução ótima racional $(\hat{x}, \hat{z})$ com o tamanho de $\hat{x}$ e $\hat{z}$ limitados por um polinômio em $|V|$ e $|\mathcal{C}|$ (fato C.4, apêndice C). Como, além disso, o número de chamadas a RAND é $|V|$, o algoritmo MaxSat-GW é um algoritmo probabilístico polinomial.

Teorema 6.3: *Se $(V, \mathcal{C})$ é uma instância do problema* MAXSAT *na qual cada cláusula tem no máximo k variáveis, com $k \geq 1$, e $X_{\mathcal{C}}$ é a variável aleatória*[3] *cujo valor é o número de cláusulas de $\mathcal{C}$ satisfeitas por uma valoração produzida pelo algoritmo* MAXSAT-GW $(V, \mathcal{C})$, *então*

No máximo k (ao invés de pelo menos k)

$$\mathbf{E}[X_{\mathcal{C}}] \;\geq\; \left(1 - (1 - \frac{1}{k})^k\right) \text{opt}(V, \mathcal{C}) \,.$$

Demonstração: Para cada cláusula C, defina Z_C da maneira usual: Z_C vale 1 se a valoração produzida pelo algoritmo satisfaz C e 0 em caso contrário. A probabilidade de que Z_C seja 0 é o produto de termos da forma $\hat{x}_v$, para v em C_0, e $(1 - \hat{x}_v)$, para v em C_1. Se t é o número de

[3] O espaço de probabilidade é $(\Omega, \mathcal{P}_1) \times \cdots \times (\Omega, \mathcal{P}_{|V|})$, onde $\Omega = \{0, 1\}$ e $\mathcal{P}_i$ é dado por $\mathcal{P}_i(1) = \hat{x}_i$ e $\mathcal{P}_i(0) = 1 - \hat{x}_i$ para cada i em V.

variáveis de C então $t \geq 1$ e

$$\begin{aligned}
\mathbf{Pr}[Z_C{=}1] &= 1 - \textstyle\prod_{v\in C_0} \hat{x}_v \prod_{v\in C_1} (1-\hat{x}_v) \\
&\geq 1 - ((t-\hat{z}_C)/t)^t && (6.3)\\
&= 1 - (1-\hat{z}_C/t)^t \\
&\geq (1-(1-1/t)^t)\hat{z}_C && (6.4)\\
&\geq (1-(1-1/k)^k)\hat{z}_C\,. && (6.5)
\end{aligned}$$

Para justificar (6.3), note que a média geométrica de números não-negativos não ultrapassa a sua média aritmética. Disso temos que

$$\begin{aligned}
\textstyle\prod_{v\in C_0} \hat{x}_v \prod_{v\in C_1} (1-\hat{x}_v) &\leq ((\textstyle\sum_{v\in C_0} \hat{x}_v + \sum_{v\in C_1} (1-\hat{x}_v))/t)^t \\
&= ((t - \textstyle\sum_{v\in C_0} (1-\hat{x}_v) - \sum_{v\in C_1} \hat{x}_v)/t)^t \\
&\leq ((t-\hat{z}_C)/t)^t\,.
\end{aligned}$$

Para verificar (6.4), considere a função $f(z) := 1-(1-z/t)^t$. Observe que f é côncava no intervalo fechado $[0,1]$, pois $f'(z) \geq 0$ e $f''(z) \leq 0$ nesse intervalo. Então, como $f(0) = 0$, temos que $f(z) \geq z\, f(1)$, que se traduz em (6.4). Finalmente, (6.5) vale pois $(1-1/t)^t$ é crescente para $t \geq 1$ e, por hipótese, $t \leq k$.

Podemos agora calcular o valor esperado de $X_{\mathcal{C}}$:

$$\begin{aligned}
\mathbf{E}[X_{\mathcal{C}}] &= \mathbf{E}[\textstyle\sum_C Z_C] \\
&= \textstyle\sum_C \mathbf{E}[Z_C] \\
&= \textstyle\sum_C \mathbf{Pr}[Z_C{=}1] \\
&\geq \textstyle\sum_C (1-(1-1/k)^k)\hat{z}_C \\
&\geq (1-(1-1/k)^k)\,\mathrm{opt}(V,\mathcal{C})\,,
\end{aligned}$$

onde a última desigualdade segue de (6.2). □

Como $(1-1/k)^k$ é não passa de $1/e$ para todo $k \geq 1$, onde e é a base dos logaritmos naturais, temos que $(1-1/k)^k < 1/e < 0{,}37$. Desse fato, e do teorema anterior, é imediato o seguinte resultado.

Teorema 6.4: *O algoritmo* MaxSat-GW *é uma 0,63-aproximação probabilística polinomial para o* MaxSat. □

Algoritmo combinado

O teorema 6.1 permite deduzir uma razão de aproximação melhor do algoritmo MaxSat-Johnson quando todas as cláusulas têm pelo menos

	Razões de aproximação			
Valor de k	1	2	3	4
MaxSat-Johnson	1/2	3/4	7/8 = 0,875	0.94
MaxSat-GW	1	3/4	19/27 = 0,7	0.68

duas variáveis. Já para o algoritmo MAXSAT-GW obtêm-se resultados melhores quando as cláusulas têm no máximo duas variáveis, pelo teorema 6.3. Parece razoável, portanto, combinar os dois algoritmos. O algoritmo abaixo deve-se a Goemans e Williamson [GW94] também.

Algoritmo MAXSAT-COMBINADO-GW $(V, \mathcal{C})$
1 $\dot{x} \leftarrow$ MAXSAT-JOHNSON (V)
2 $\ddot{x} \leftarrow$ MAXSAT-GW $(V, \mathcal{C})$
3 seja $\dot{s}$ o número de cláusulas de $\mathcal{C}$ satisfeitas por $\dot{x}$
4 seja $\ddot{s}$ o número de cláusulas de $\mathcal{C}$ satisfeitas por $\ddot{x}$
5 se $\dot{s} \geq \ddot{s}$
6 então devolva $\dot{x}$
7 senão devolva $\ddot{x}$

Teorema 6.5: *O algoritmo* MAXSAT-COMBINADO-GW *é uma* 0,75*-aproximação probabilística polinomial para o* MAXSAT.

Demonstração: Seja $\mathcal{C}_k$ a coleção das cláusulas de $\mathcal{C}$ que têm exatamente k variáveis. Sejam $X_{\mathcal{C}}$, $\dot{X}_{\mathcal{C}}$ e $\ddot{X}_{\mathcal{C}}$ variáveis aleatórias cujos valores são o número de cláusulas de $\mathcal{C}$ satisfeitas por uma valoração produzida por MAXSAT-COMBINADO-GW $(V, \mathcal{C})$, MAXSAT-JOHNSON (V) e MAXSAT-GW $(V, \mathcal{C})$, respectivamente. Note que $X_{\mathcal{C}} \geq \frac{1}{2}(\dot{X}_{\mathcal{C}} + \ddot{X}_{\mathcal{C}})$. Se refizermos as partes apropriadas das demonstrações dos teoremas 6.1 e 6.3 temos

pois o máximo é maior ou igual a média

$$\begin{aligned}
\mathbf{E}[X_{\mathcal{C}}] &\geq \mathbf{E}[\tfrac{1}{2}(\dot{X}_{\mathcal{C}} + \ddot{X}_{\mathcal{C}})] \\
&= \tfrac{1}{2}(\mathbf{E}[\dot{X}_{\mathcal{C}}] + \mathbf{E}[\ddot{X}_{\mathcal{C}}]) \\
&\geq \tfrac{1}{2}\textstyle\sum_k \sum_{C \in \mathcal{C}_k} ((1 - 2^{-k}) + (1 - (1 - k^{-1})^k)\hat{z}_C) \\
&\geq \tfrac{1}{2}\textstyle\sum_k \sum_{C \in \mathcal{C}_k} (1 - 2^{-k} + 1 - (1 - k^{-1})^k)\hat{z}_C \\
&\geq \tfrac{1}{2}\textstyle\sum_k \sum_{C \in \mathcal{C}_k} \tfrac{3}{2}\,\hat{z}_C \qquad (6.6) \\
&\geq \tfrac{3}{4}\,\mathrm{opt}(V, \mathcal{C})\,. \qquad (6.7)
\end{aligned}$$

Analisar a tabela anterior

onde a desigualdade (6.6) pode ser concluída analisando-se os casos $k = 1$, $k = 2$ e $k \geq 3$ e (6.7) vale em virtude de (6.2). □

6.2 Desaleatorização

Em muitos casos, é possível converter um algoritmo de aproximação probabilístico em um determinístico com uma razão de aproximação

igual à razão esperada do algoritmo probabilístico. Isso pode ser feito através do *método das esperanças condicionais* [AS92, ES74, Spe87]. Para aplicá-lo, precisamos saber calcular eficientemente certas esperanças condicionais, que dependem do problema em foco. É esse o cerne do processo de desaleatorização de um algoritmo probabilístico por esse método. Ilustramos o método aplicando-o ao algoritmo MAXSAT-JOHNSON. Obtemos como resultado a seguinte 0,5-aproximação polinomial determinística, que utiliza um procedimento ESPCOND, detalhado posteriormente.

Algoritmo MAXSAT-JOHNSON-DESALEATORIZADO $(V, \mathcal{C})$

```
1   D ← C
2   para cada v em V faça
3       se EspCond(v, 1, D) ≥ EspCond(v, 0, D)
4           então ẋ_v ← 1
5                 para cada C em D faça
6                     se v ∈ C_1
7                         então D ← D \ {C}
8                         senão C_0 ← C_0 \ {v}
9           senão ẋ_v ← 0
10                para cada C em C faça
11                    se v ∈ C_0
12                        então D ← D \ {C}
13                        senão C_1 ← C_1 \ {v}
14  devolva ẋ
```

Cláusula C satisfeita por v é removida

v é removido de C, pois não pode satisfazê-la

No algoritmo acima, $\mathcal{D}$ é um multiconjunto (um conjunto em que cada elemento aparece com uma certa multiplicidade). No início de cada iteração, $\mathcal{D}$ contém, além de pares $\{\emptyset, \emptyset\}$, que chamaremos de **pseudo-cláusulas**, apenas as cláusulas de $\mathcal{C}$ que a "valoração parcial" $\dot{x}$ do algoritmo ainda não satisfez. Ademais, de cada cláusula de $\mathcal{C}$ em $\mathcal{D}$, foram eliminadas as variáveis cujos valores $\dot{x}$ já fixou.

O procedimento auxiliar ESPCOND recebe uma variável v, um elemento i de $\{0, 1\}$ e um multiconjunto $\mathcal{D}$ de cláusulas e pseudo-cláusulas. Seja $X_{\mathcal{D}}$ a variável aleatória cujo valor é o número de cláusulas em $\mathcal{D}$, levando em conta a multiplicidade, satisfeitas por uma valoração produzida pelo algoritmo MAXSAT-JOHNSON (V). Se $i = 1$ então o procedimento devolve o valor esperado de $X_{\mathcal{D}}$ sob a condição que $\dot{x}_v$=1 no algoritmo

MAXSAT-JOHNSON. Este valor esperado é denotado por $\mathbf{E}[X_{\mathcal{D}}|\dot{x}_v{=}1]$. Se $i = 0$ então o procedimento devolve $\mathbf{E}[X_{\mathcal{D}}|\dot{x}_v{=}0]$, que é definido de maneira análoga.

Procedimento ESPCOND$(v, i, \mathcal{D})$

```
1  esp ← 0
2  para cada C em D faça
3      k ← |C_1| + |C_0|
4      se v ∈ C_i
5          então esp ← esp + 1
6          senão se v ∈ C_{1−i}
7                    então esp ← esp + (1 − 2^{−k+1})
8                    senão esp ← esp + (1 − 2^{−k})
9  devolva esp
```

Calcula a Esperança do número de cláusulas satisfeitas, condicionado a variável v ter valor i=0 ou i=1

Teorema 6.6: *O algoritmo* MAXSAT-JOHNSON-DESALEATORIZADO *é uma* 0,5*-aproximação polinomial para o* MAXSAT.

Demonstração: Seja $X_{\mathcal{C}}$ a variável aleatória cujo valor é o número de cláusulas em $\mathcal{C}$ satisfeitas por uma valoração produzida pelo algoritmo MAXSAT-JOHNSON(V). Como, no algoritmo MAXSAT-JOHNSON, $\dot{x}_v = 1$ com probabilidade $1/2$ e $\dot{x}_v = 0$ com probabilidade $1/2$, temos que $\mathbf{E}[X_{\mathcal{C}}] = \frac{1}{2}\mathbf{E}[X_{\mathcal{C}}|\dot{x}_v{=}1] + \frac{1}{2}\mathbf{E}[X_{\mathcal{C}}|\dot{x}_v{=}0]$, e disso concluímos que $\mathbf{E}[X_{\mathcal{C}}]$ não ultrapassa o maior entre as duas esperanças condicionais, $\mathbf{E}[X_{\mathcal{C}}|\dot{x}_v{=}1]$ e $\mathbf{E}[X_{\mathcal{C}}|\dot{x}_v{=}0]$. Logo, $\mathbf{E}[X_{\mathcal{C}}] \leq \mathbf{E}[X_{\mathcal{C}}|\dot{x}_{v_1}]$, onde o último termo denota o valor esperado de $X_{\mathcal{C}}$ sob a condição que, no algoritmo MAXSAT-JOHNSON, atribua-se a $\dot{x}_{v_1}$ o mesmo valor atribuído no algoritmo MAXSAT-JOHNSON-DESALEATORIZADO. Uma vez fixado o valor de $\dot{x}_{v_1}$, o mesmo raciocínio vale para as demais variáveis e, disso, concluímos que

Em MaxSat-GW, é um pouco diferente

X = (x_v) * E[X | v=1] + (1-x_v)*E[X | v=0]

Pois fomos escolhendo as maiores esperanças condicionais

$$\mathbf{E}[X_{\mathcal{C}}] \leq \mathbf{E}[X_{\mathcal{C}}|\dot{x}_{v_1}] \leq \mathbf{E}[X_{\mathcal{C}}|\dot{x}_{v_1}, \dot{x}_{v_2}] \leq \ldots \leq \mathbf{E}[X_{\mathcal{C}}|\dot{x}_{v_1}, \ldots, \dot{x}_{v_n}]\,.$$

O último termo é o número de cláusulas de $\mathcal{C}$ satisfeitas pela valoração $\dot{x}$ produzida pelo algoritmo MAXSAT-JOHNSON-DESALEATORIZADO. Esse número é pelo menos $0{,}5\,\mathrm{opt}(V, \mathcal{C})$, pois $\mathbf{E}[X_{\mathcal{C}}] \geq 0{,}5\,\mathrm{opt}(V, \mathcal{C})$.

Quanto ao tempo de execução do algoritmo, basta observar que o procedimento ESPCOND$(v, i, \mathcal{D})$ consome tempo $O(|V|)$. Disso, é fácil concluir que o algoritmo MAXSAT-JOHNSON-DESALEATORIZADO é polinomial. □

É um pouco surpreendente que desaleatorizações produzam, mesmo sem levar em conta como as variáveis aleatórias se distribuem em torno do valor esperado, um algoritmo com razão de aproximação igual à razão esperada do algoritmo probabilístico. No caso do algoritmo acima, o procedimento para o cálculo das esperanças condicionadas é bem simples. Em outros problemas, esses cálculos podem ser bastante complicados, inviabilizando o processo de desaleatorização.

6.3 Geradores de números aleatórios

Uma maneira mais geral de definir algoritmos probabilísticos parte de um gerador de números aleatórios no intervalo $[0, 1)$. Nessa definição alternativa, substitui-se o algoritmo RAND por um algoritmo ideal RANDUNI que devolve um número real aleatório com distribuição uniforme no intervalo $[0, 1)$ (apêndice D). Existem implementações aproximadas desse algoritmo RAND [Knu98].

RANDUNI

Os conceitos introduzidos no início deste capítulo (algoritmo de aproximação probabilístico, algoritmo probabilístico polinomial, etc.) podem ser naturalmente adaptados a esta outra definição. Em particular, os algoritmos vistos neste capítulo são probabilísticos polinomiais também de acordo com esta nova definição, pois é fácil implementar RAND(ρ) com uma chamada a RANDUNI: o algoritmo RAND devolve 1 se o número devolvido por RANDUNI for menor que ρ e devolve 0 em caso contrário.

Por outro lado, não se sabe, em geral, converter um algoritmo que chama RANDUNI um número polinomial de vezes em um que chama RAND um número polinomial de vezes. Por isso é razoável dizer que esta nova definição é uma generalização da original. A desvantagem da nova definição é que ela requer manipulação números reais.

Exercícios

6.1 Considere o seguinte algoritmo para o MAXSAT $(V, \mathcal{C})$.

```
Algoritmo MaxSat-CaraCoroa(V)
1  b ← Rand(1/2)
2  para cada v em V faça x_v ← b
3  devolva x
```

Mostre que o algoritmo acima é uma 0,5-aproximação probabilística polinomial para o MAXSAT.

- Seja C uma cláusula qualquer. Qual a probabilidade de C ser verdadeira?
- Se C possui literais todos positivos (não complementados), então a probabilidade de todos serem verdadeiros é 1/2.
- Se C possui literais todos negativos (complementados), então a probabilidade de todos serem falsos é 1/2.
- Se C possui literais positivos e negativos, então a probabilidade de C ser verdadeira é 1.
- Portanto, qualquer cláusula tem pelo menos 1/2 de probabilidade de ser verdadeira, e o número de cláusulas satisfeitas é em média metade do total de cláusulas. Essa número esperado é maior ou igual a metade do ótimo (número máximo de cláusulas satisfeitas).

Vamos criar um conjunto A aleatoriamente. Para cada vértice v de G, coloque v em A com probabilidade 1/2. Seja B = V(G)-A. Vamos analisar o peso do corte (A,B) de G.

Cada aresta (u,v) tem 1/2 de probabilidade de estar entre A e B:
u em A e v em B ---> prob. 1/4
u em B e v em A ---> prob. 1/4

O peso esperado do corte (A,B) é metade do peso total das arestas.
Esse peso esperado é maior ou igual à metade do peso do corte máximo.

6.2 Inspire-se no algoritmo MAXSAT-JOHNSON e projete uma 0,5-aproximação probabilística polinomial para o **problema do corte máximo**:

> **Problema** MAXCUT (G, w): *Dado um grafo G e um peso w_e em $\mathbb{Q}_{\geq}$ para cada aresta e de G, encontrar um corte R que maximize $w(R)$.*

Apresente uma versão desaleatorizada do seu algoritmo.

6.3 Projete uma 0,5-aproximação probabilística polinomial para a seguinte variante do MAXCUT (exercício 6.2), que denotamos por MAXkCUT: dados um grafo G, um peso w_e em $\mathbb{Q}_{\geq}$ para cada aresta e e um inteiro $k \geq 2$, determinar uma partição de V_G em k partes que maximize a soma do peso das arestas com extremos em partes distintas da partição.

6.4 Projete uma 0,5-aproximação probabilística polinomial para o problema da cobertura máxima (exercício 2.4). Mostre que a versão desaleatorizada deste algoritmo é exatamente o algoritmo guloso sugerido no exercício 2.4.

6.5 Escreva uma versão desaleatorizada do algoritmo MAXSAT-GW. Apresente-a da forma mais simples que puder omitindo, se possível, todos os vestígios "probabilísticos" do algoritmo original. Prove, sem usar argumentos probabilísticos, que o algoritmo obtido é uma 0,63-aproximação para MAXSAT.

É possível também obter um algoritmo MaxSat-Combinado-GW-Desaleatorizado com fator de aprox 3/4 determinístico

6.6 Considere o seguinte algoritmo probabilístico para o MAXSAT:

> **Algoritmo** MAXSAT-COMBINADO-PROBABILÍSTICO $(V, \mathcal{C})$
> 1 $b \leftarrow$ RAND $(\frac{1}{2})$
> 2 se $b = 0$
> 3 então $\dot{x} \leftarrow$ MAXSAT-JOHNSON (V)
> 4 senão $\dot{x} \leftarrow$ MAXSAT-GW $(V, \mathcal{C})$
> 5 devolva $\dot{x}$

Prove que esse algoritmo é uma 0,5-aproximação probabilística polinomial para o MAXSAT. Escreva e comente a versão desaleatorizada deste algoritmo. O que muda se alterarmos o $\frac{1}{2}$ da linha 1 do algoritmo para um outro valor no intervalo $(0, 1)$?

6.7 Considere a seguinte versão ponderada do MAXSAT. Além do conjunto V e da coleção $\mathcal{C}$ de cláusulas sobre V, é dado também um peso w_C em $\mathbb{Q}_{\geq}$ para cada cláusula C em $\mathcal{C}$; o problema consiste em encontrar uma valoração que maximize a soma dos pesos das cláu-

O algoritmo Graham-aleatório é 2-aproximativo, pois Escalonamento-Graham é 2-aproximativo.

É possível obter instâncias que se aproximam do fator 2 de aproximação esperado.

Instância limite: n tarefas de tempo n e (n+1)n^2 tarefas de tempo 1 em n(n+2) máquinas.
Ótimo: tempo n em cada máquina.

Ordem aleatória: Sorteia uniformemente número em [0,1] para cada tarefa.
A ordem desses números determina a ordem das tarefas.

Ideia: Order Statistics: Número esperado da última tarefa grande n/(n+1)
Aproximadamente (n/(n+1)) * (n+1)n^2 = n^3 tarefas pequenas antes da última tarefa grande.

Algoritmo Escalonamento-Graham: tempo n + n^3/(n(n+2)) ---> 2n = 2 * opt

(ver slides da disciplina)

sulas satisfeitas. Modifique os três algoritmos apresentados neste capítulo, bem como suas análises, para o MaxSat ponderado. Que razão de aproximação tem cada um dos algoritmos modificados?

6.8 Considere o seguinte algoritmo para o Escalonamento (seção 2.1): escolha aleatoriamente a próxima tarefa a ser escalonada e aplique o critério de Graham. Estime o valor esperado da solução assim obtida.

Notas bibliográficas

A discussão do algoritmo MaxSat-GW é um resumo de um artigo de Goemans e Williamson [GW94]; resumos semelhantes aparecem no livro de Motwani e Raghavan [MR95b] e no livro editado por Hochbaum [Hoc97]. Os exercícios 6.1, 6.3 e 6.4 foram extraídos do livro de Vazirani [Vaz01].

O livro de Motwani e Raghavan [MR95b] é um texto abrangente sobre algoritmos probabilísticos, com muitos exercícios e problemas de pesquisa.

O algoritmo MaxSat-Johnson é atribuído a Johnson [Joh74], embora este tenha escrito a versão não-probabilística do algoritmo. A primeira 0,75-aproximação polinomial para o MaxSat deve-se a Yannakakis [Yan94]. Zwick [Zwi99] projetou um algoritmo de aproximação para um caso particular do MaxSat e conjecturou, baseando-se em experimentos númericos, que seu algoritmo é uma 0,797-aproximação para o problema. Recentemente, Asano e Williamson [AW00] projetaram uma 0,784-aproximação, combinando vários algoritmos conhecidos para o problema [GW94, FG95, KZ97]. No momento, esse é o melhor algoritmo de aproximação conhecido para o MaxSat. Utilizando o algoritmo de Zwick [Zwi99], Asano e Williamson obtêm uma 0,833-aproximação para o MaxSat, desde que a conjectura de Zwick seja confirmada.

A técnica de arredondamento probabilístico foi introduzida por Raghavan e Thompson [RT87, Rag88] e vem sendo aplicada com bastante sucesso a vários problemas [BTV99, CKR00, Fei99, JV00].

O método das esperanças condicionais aparece implicitamente em um artigo de Erdős e Selfridge [ES73]. A conexão com algoritmos polinomiais determinísticos foi feita por Spencer [Spe87].

Uma outra técnica bem sucedida de desaleatorização de algoritmos probabilísticos foi desenvolvida por Luby [Lub86] e por Alon *et al.* [ABI86]. Uma sucessão de chamadas ao algoritmo Rand produz

uma seqüência de *bits* aleatórios. Podemos imaginar então que, além dos dados do problema, um algoritmo probabilístico recebe uma seqüência de *bits* aleatórios, de comprimento conveniente. Freqüentemente, basta que os *bits* nessa seqüência sejam k-mutuamente independentes. Para explicar este conceito, denote por $b_1, \ldots, b_n$ os *bits* produzidos por n chamadas de RAND. Dizemos que $b_1, \ldots, b_n$ são **k-mutuamente independentes** se, para qualquer subseqüência $b_{i_1}, \ldots, b_{i_k}$ de $b_1, \ldots, b_n$, a probabilidade de que $b_{i_1}, \ldots, b_{i_k}$ coincida com qualquer das 2^k seqüências de k *bits* é 2^{-k}. Enquanto independência plena requer um espaço de probabilidade composto de 2^n vetores binários, pode-se construir espaços de probabilidade com $n^{k/2}$ vetores binários cujos *bits* são k-mutuamente independentes. Se k é constante, um algoritmo probabilístico pode ser desaleatorizado da seguinte forma: executa-se o algoritmo para cada um dos vetores neste espaço de probabilidade e devolve-se a melhor das soluções produzidas. Isso resulta em um algoritmo polinomial, determinístico, e com razão de aproximação igual à razão esperada do algoritmo probabilístico original. Para saber mais sobre isso, veja o artigo de Chor e Goldreich [CG89], bem como o relatório técnico de Luby e Wigderson [LW95].

A respeito de geradores de números pseudo-aleatórios, veja os livros de Johnson [Joh87], Devroye [Dev86] e Knuth [Knu98]. Knuth descreve implementações aproximadas do algoritmo RANDUNI que fornecem números pseudo-aleatórios satisfatórios em tempo probabilístico constante: o tempo *esperado* de execução de tais implementações é limitado por uma constante.