Spis rzeczy

6 Mapy fizyczne genomów **89**
6.2 Problem podwójnego trawienia 89
6.2.1 Metoda symulowanego wyżażania 90
6.2.2 Zastosowanie algorytmu Metropolisa do problemu komiwojażera 95
6.2.3 Zastosowanie algorytmu Metropolisa do DDP 96

6 Mapy fizyczne genomów

6.2 Problem podwójnego trawienia

Problem *podwójnego trawienia* (DDP) polega na następującym zadaniu. Mamy dane trzy multizbiory[1] liczb rzeczywistych

$$A = \{a_1, \ldots, a_n\} \subseteq \mathbb{R}^+,$$

$$B = \{b_1, \ldots, b_m\} \subseteq \mathbb{R}^+,$$

$$A \wedge B = \{c_1, \ldots, c_k\} \subseteq \mathbb{R}^+.$$

Multizbiór A zawiera długości wszystkich odcików otrzymanych przez pocięcie genomu enzymem A. Podobnie dla B. Multizbiór $A \wedge B$ zawiera wszystkie odcinki powstałe w wyniku cięcia genomu obydwoma enzymami. Musimy dobrać takie uporządkowanie A oraz B, aby otrzymać mapę zgodną z $A \wedge B$. Oczywiście warunkiem koniecznym jest równość sum:

$$\sum_{i=1}^{n} a_i = \sum_{i=1}^{m} b_i = \sum_{i=1}^{k} c_i.$$

Ścisłe sformułowanie tego problemu jest następujące. Znaleźć permutacje σ na zbiorze $\{1, \ldots, n\}$ oraz μ na zbiorze $\{1, \ldots, m\}$ tak, aby zachodził

[1] Multizbiór tym się różni od zbioru, że pewne elementy mogą występować wiecej niż jeden raz, natomiast kolejność elementów nie ma znaczenia.

następujący warunek. Niech

$$S = \{\sum_{i=1}^{j} a_{\sigma(i)} \mid 0 \leq j \leq n\} \cup \{\sum_{i=1}^{j} b_{\mu(i)} \mid 0 \leq j \leq m\}.$$

Uporządkujmy zbiór S rosnąco: $S = \{s_0 \leq s_1 \leq \ldots \leq s_l\}$. Niech $C_{\sigma,\mu} = \{s_i - s_{i-1} \mid 1 \leq i \leq l\}$. Powiemy, że para permutacji (σ, μ) jest *rozwiązaniem* problemu DDP, gdy

$$A \wedge B = C_{\sigma,\mu}.$$

Problem DDP polega stwierdzeniu czy istnieją permutacje (σ, μ) będące rozwiązaniem dla danych $A, B, A \wedge B$.

Twierdzenie 6.2.1 *DDP jest problemem NP-zupełnym.*

Dowód: Oczywiście DDP należy do klasy NP. Dla dowodu NP-trudności zredukujemy problem podziału do DDP. *Problem podziału* definiuje się następująco. Dany skończony zbiór F oraz funkcja $f : F \to \mathbb{N}$. Czy istnieje $F' \subseteq F$ taki, że

$$\sum_{x \in F'} f(x) = \sum_{x \in F - F'} f(x).$$

Wiadomo, że problem podziału jest NP-zupełny.

Niech $f : F \to \mathbb{N}$ będzie instancją problemu DDP. Definiujemy

$$A = A \wedge B = \{f(x) \mid x \in F\},$$

$$B = \{\sum_{x \in F} f(x)/2, \sum_{x \in F} f(x)/2\},$$

Wówczas, jak łatwo jest pokazać $A, B, A \wedge B$ ma rozwiązanie (jako DDP) wtedy i tylko wtedy, gdy F można rozbić na dwa podzbiory o równych sumach. ∎

6.2.1 Metoda symulowanego wyżażania

Jest to metoda zapożyczona z fizyki statystycznej. Została ona zaproponowana przez Metropolisa w 1953r. Metoda symulowanego wyżażania (simulated annealing)[2] jest stosowana do rozwiązywania problemu znajdowania

[2]Dosłowne tłumaczenie 'simulated annealing' to symulowane schładzanie; nazwa 'symulowane wyżażanie' przyjęła się w języku polskim.

minimum funkcji $f : V \to \mathbb{R}$, gdzie V jest pewnym skończonym zbiorem. Metoda ta jest ulepszeniem tzw. metody Monte Carlo znajdowania minimum. Ta ostatnia metoda polega na iteracyjnym losowaniu punktów $v \in V$ (z prawdopodobieństwem $1/|V|$), zapamiętując $f(v)$ w przypadku, gdy jest ona najmniejszą dotąd znalezioną wartością. Przy dużym zbiorze V małe jest prawdopodobieństwa znalezienia minimum.

Metoda Metropolisa polega na rozważeniu rozkładów prawdopodobieństwa π_T (tzw. *rozkłady Gibbsa*) indeksowanych parametrem $T > 0$. T pełni rolę temperatury środowiska. Dla $u \in V$ definujemy

$$\pi_T(u) = \frac{e^{-f(u)/T}}{\sum_{v \in V} e^{-f(v)/T}}.$$

Intuicyjnie π_T przedstawia rozkład energii przy temperaturze T. Zauważmy, że dla dużych wartości T, rozkład π_T staje się coraz bardziej zbliżony do rozkładu jednorodnego:

$$\pi_\infty(u) = \lim_{T \to \infty} \pi_T(u) = 1/|V|.$$

Natomiast przy wartościach T zbiżających się do 0 (z prawej strony), rozkład π_T staje się coraz bardziej skupiony w wartościach minimalnych funkcji f. Niech $\mathrm{MIN}_f = \{w \in V \mid \forall v f(w) \leq f(v)\}$. Wówczas

$$\pi_0 = \lim_{T \to 0^+} \pi_t(u) = \begin{cases} 0, & \text{gdy } u \notin \mathrm{MIN}_f \\ \frac{1}{|\mathrm{MIN}_f|}, & \text{gdy } u \in \mathrm{MIN}_f \end{cases}$$

Wynika to z faktu, że

$$\lim_{T \to 0^+} e^{a/T} = \begin{cases} +\infty, & a > 0 \\ 1, & a = 0 \\ 0, & a < 0 \end{cases}$$

Technicznie problem polega na tym jak symulować rozkłady π_T bez konieczności obliczania wszystkich wartości $f(v)$, dla $v \in V$.

Zastosujemy łańcuchy Markowa. Rozważmy łańcuch Markowa o $|V|$ stanach $\{1, \ldots, |V|\}$. Macierz przejścia dla tego łańcucha jest zadana przez prawdopodobieństwa $p(i, j)$ (przejście ze stanu i do stanu j), dla $1 \leq i, j \leq |V|$. Ponadto mamy *początkowy rozkład* μ, gdzie $\mu(i)$ jest prawdopodobieństwem znalezienia się łancucha w chwili początkowej w stanie i. Wówczas, dla $n \in \mathbb{N}$, n-ta potęga P^n macierzy P opisuje prawdopodobieństwa przejścia pomiędzy stanami w n krokach. Tak więc, $\mu \cdot P^n$ (traktowane jako mnożenie wektora przez macierz) jest rozkładem prawdopodobieństwa znalezienia się

układu w danym stanie po n krokach, zakładając rozkład początkowy μ. Niech $p^n(i,j)$ oznacza (i,j) element macierzy P^n. Mamy następującą równość dla $1 \leq i \leq |V|$,

$$(\mu \cdot P^n)(i) = \sum_{j=1}^{|V|} \mu(j) \cdot p^n(j,i).$$

Powiemy, że stany i, j łańcucha Markowa *komunikują się ze sobą*, gdy istnieją ścieżki o dodatnim prawdopodobieństwie łączące i z j oraz j z i.

Niech i będzie stanem. Rozważmy wszystkie liczby $k \geq 1$ takie, że $p^k(i,i) > 0$. *Okresem* stanu i nazwiemy największy wspólny podzielnik liczb $\{k \mid p^k(i,i) > 0\}$. Zauważmy, że jeśli i, j komunikują się ze sobą, to mają one ten sam okres. Powiemy, że łańcuch Markowa jest *aperiodyczny*, gdy okres każdego stanu wynosi 1.

Twierdzenie 6.2.2 (O rozkładzie stacjonarnym)
Niech P będzie macierzą przejść pewnego aperiodycznego łańcucha Markowa, w którym każde dwa stany komunikują się ze sobą. Wówczas istnieje dokładnie jeden rozkład π spełniający następujące warunki:
(i) *Dla każdego początkowego rozkładu μ,* $\lim_{n\to\infty} \mu \cdot P^n = \pi$.
(ii) $\pi \cdot P = \pi$.

W powyższym twierdzeniu (i) oznacza, że bez względu na rozkład początkowy, po dostatecznie dużej liczbie kroków łańcucha, rozkład prawdopodobieństwa stanów będzie zbiżony do rozkładu stacjonarnego π. Natomiast (ii) pozwala łatwo sprawdzać, czy dany rozkład jest stacjonarny.

Główny pomysł algorytmu Metropolisa polega na zdefiniowaniu łańcucha Markowa o zbiorze stanów V, którego rozkład stacjonarny jest π_T.

Przypuśćmy, że dla każdego $v \in V$ mamy określony zbiór $N_v \subseteq V$ tak, że przyporządkowanie $v \mapsto N_v$ spełnia następujące warunki dla dowolnych $w, v \in V$,
(i) $|N_v| = |N_w|$,
(ii) $v \in N_w \iff w \in N_v$,
(iii) istnieje $k \geq 1$ oraz ciąg stanów $v_1, \ldots, v_k$ spełniający
$v_1 \in N_w, v_2 \in N_{v_1}, \ldots, v_k \in N_{v_{k-1}}, v \in N_{v_k}$,
(iv) $v \in N_v$.

Wówczas jeśli przyjmiemy prawdopodobieństa przejść tak, aby

$$p_T(w,v) > 0, \quad \text{dla } v \in N_w,$$

to z (iii) wynika, że każde dwa stany komunikują się ze sobą, a z (iv) wynika, że łańcuch jest aperiodyczny.

Dla $a \in \mathbb{R}$ niech $a^+ = \max\{a, 0\}$. Zdefiniujemy teraz prawdopodobieństwa przejść:
I. Dla $v \notin N_w$, $p_T(w, v) = 0$;
II. Dla $v \in N_w - \{w\}$,

$$p_T(w, v) = \frac{e^{-(f(v)-f(w))^+/T}}{|N_w|};$$

III. $p_T(w, w)$ jest wyznaczone przez warunek

$$\sum_{v \in N_w} p_T(w, v) = 1.$$

Twierdzenie 6.2.3 (Metropolis)
Rozkład stacjonarny opisanego łańcucha Markowa jest równy rozkładowi Gibbsa, tzn.

$$\pi_T(u) = \frac{e^{-f(u)/T}}{\sum_{v \in V} e^{-f(v)/T}}.$$

Dowód: Wystarczy sprawdzić warunek (ii) Twierdzenia 6.2.2. Najpierw pokażemy, że π_T spełnia następujący warunek równowagi, dla $w, v \in V$:

$$p_T(w, v) \cdot \pi_T(w) = p_T(v, w) \cdot \pi_T(v). \tag{6.6.1}$$

Oznaczmy przez S sumę $\sum_v \in V e^{-f(v)/T}$. Niech $v \in N_w$, $v \neq w$. Wówczas

$$p_T(w, v) \cdot \pi_T(w) = \frac{e^{-(f(v)-f(w))^+/T} \cdot e^{-f(w)/T}}{|N_w| \cdot S}$$

$$= \frac{e^{-[(f(v)-f(w))^+ + f(w)]/T}}{|N_w| \cdot S}$$

$$= \frac{e^{-\max\{f(v), f(w)\}}}{|N_w| \cdot S}.$$

Ponieważ na mocy założenia (i) $|N_w| = |N_v|$, to powyższy warunek jest symetryczny za względu na w i v. Stąd wynika (6.6.1). Mamy więc

$$(\pi_T \cdot P)(v) = \sum_{w \in V} p_T(w, v) \cdot \pi_T(w) = \sum_{w \in V} p_T(v, w) \cdot \pi_T(v)$$
$$= \pi_T(v) \cdot \sum_{w \in V} p_T(v, w) = \pi_T(v).$$

Z Twierdzenia 6.2.2(ii) wynika, że π_T jest rozkładem stacjonarnym. ∎

Zatem aby probabilistycznie znajdować minimum funkcji, symulujemy powyższy łańcuch Markowa dla małych wartości $T > 0$. Po dostatecznie dużej liczbie kroków, rozkład będzie się zbliżał do stacjonarnego rozkładu π_T, który dla małych T jest coraz bardziej skupiony na punktach minimalnych funkcji. Obliczanie wartości funkcji może być kosztowne. Aby tego uniknąć, symulacja łańcucha wygląda następująco. Będąc w stanie $w \in V$, wybieramy losowo (jednorodny rozkład) stan $v \in N_w$. Obliczamy $f(v)$. Przejście do v następuje z prawdopodobieństwem

$$e^{\frac{-(f(v)-f(w))^+}{T}}.$$

Powyższą metodę można połączyć z jednoczesnym obniżaniem temperatury tak, aby w granicy zbliżyć się do rozkładu π_0, skupionego w punktach minimalnych. Okazuje się, że jeśli nie obniżamy temperatury zbyt szybko, to proces ten będzie zbiegał do żądanego rozkładu.

Twierdzenie 6.2.4 (Geman, Geman)
Dla algorytmu Metropolisa niech T_n będzie temperaturą na n-tym kroku procesu symulacji. Niech X_n będzie zmienną losową przedstawiającą stan układu po n krokach. Niech c będzie stałą taką, że $c \geq f(v) - f(w)$, dla $v, w \in V$. Jeśli ciąg T_n jest taki, że $T_n / \log n \geq c$ oraz $\lim_{n\to\infty} T_n = 0$, to dla każdego $v \in V$,

$$\lim_{n\to\infty} \mathbb{P}(X_n = v) = \pi_0(v).$$

Dowód: (*Szkic*) Pokażemy tylko, że prawdopodobieństwo wyskoczenia z lokalnego minimum jest równe 1. Niech $\Delta = f(v) - f(w) > 0$. Wówczas prawdopodobieństwo skoku z w do v wynosi

$$e^{-\Delta/T_n} = e^{\frac{-(f(v)-f(w))^+}{T_n}}.$$

Zatem prawdopodobieństwo tego, że lokalne minimum nigdy nie zostanie opuszczone wynosi

$$\prod_{n=1}^{\infty}(1 - e^{-\Delta/T_n}).$$

Niech $p_n = e^{-\Delta/T_n}$. Ponieważ

$$1 - x \leq e^{-x},$$

dla $x \in \mathbb{R}$, to

$$\prod_{n=1}^{\infty}(1-p_n) \leq \prod_{n=1}^{\infty} e^{-p_n} = e^{-\sum_n p_n}.$$

Chcemy aby $\sum_{n+1}^{\infty} p_n = +\infty$, bo wtedy powyższy iloczyn będzie równy 0. Ponieważ $-\Delta/T_n \geq -\frac{\Delta \cdot \log n}{c}$, to

$$\sum_{n=1}^{\infty} p_n = \sum_{n=1}^{\infty} e^{-\Delta/T_n} \geq \sum_{n=1}^{\infty} e^{-\frac{\Delta \cdot \log n}{c}} = \sum_{n=1}^{\infty} n^{-\Delta/c}.$$

Ponieważ $0 < \Delta/c \leq 1$, to

$$\sum_{n=1}^{\infty} n^{-\Delta/c} = +\infty.$$

■

Skuteczność metody symulowanego wyżażania zależy od wyboru otoczeń N_w oraz od wyboru funkcji energii f.

6.2.2 Zastosowanie algorytmu Metropolisa do problemu komiwojażera

Mamy dane n miast $\{1, \ldots, n\}$ z kosztem połączeń pomiędzy każdymi dwoma miastami. *Problem komiwojażera* polega na tym aby znaleźć permutację $\sigma \in S_n$ (przez S_n oznaczamy zbiór wszystkich permutacji zbioru n-elemntowego), dla której koszt odwiedzenia każdego miasta[3] jest minimalny.

Zastosujemy algorytm Metropolisa. Jako zbiór stanów przyjmujemy S_n. Dla permutacji $\sigma = (i_1, \ldots, i_n)$ definiujemy otoczenie

$$N(\sigma) = \{\tau \in S_n \mid \exists 1 \leq j \leq k \leq n : \tau = (i_1, \ldots, i_{j-1}, i_k, i_{k-1}, \ldots, i_{j+1}, i_j, i_{k+1}, \ldots, i_n)\}.$$

Oczywiście $|N(\sigma)| = |N(\tau)|$, dla $\sigma, \tau \in S_n$. Ponadto $\sigma \in N(\sigma)$. Dla sprawdzenia warunku komunikowalności weźmy dowolne permutacje $\sigma, \tau \in S_n$. definiujemy ciąg $\sigma_0, \sigma_1, \ldots, \sigma_n$ taki, że σ_k zgadza się z τ na pierwszych k pozycjach:
$\sigma_0 = \sigma$.

[3]Odwiedzamy miasta według kolejności podanej przez σ i wracamy do punktu startu.

Jeśli $\sigma_k = (i_1, \ldots, i_n)$ oraz i_* znajduje się w τ na $(k+1)$-szym miejscu, to niech j będzie miejscem, na którym i_* znajduje się w σ_k. Ponieważ pierwsze k miejsc w σ_k oraz τ są identyczne, to $j > k$. Wówczas

$$\sigma_{k+1} = (i_1, \ldots, i_k, i_j \ldots, i_{k+1}, i_{j+1}, \ldots, i_n).$$

Tak więc $\sigma_n = \tau$ oraz $\sigma_k \in N(\sigma_{k+1})$, dla $k = 0, \ldots, n-1$. Oczywiscie $\sigma \in N(\tau) \iff \tau \in N(\sigma$.

Jako funkcję kosztu na zbiorze S_n przyjmujemy koszt $f(\sigma)$ drogi wyznaczonej przez premutację σ.

6.2.3 Zastosowanie algorytmu Metropolisa do DDP

Stosujemy notację z początku tej sekcji. Mamy dane multizbiory

$$A = \{a_1, \ldots, a_n\} \subseteq \mathbb{R}^+,$$

$$B = \{b_1, \ldots, b_m\} \subseteq \mathbb{R}^+,$$

$$A \wedge B = \{c_1, \ldots, c_k\} \subseteq \mathbb{R}^+.$$

Przyjmujemy, że $A \wedge B$ jest uporządkowany niemalejąco: $c_1 \leq c_2 \leq \ldots \leq c_k$. Jako zbiór stanów przyjmujemy przystkie pary $(\sigma, \mu) \in S_n \times S_m$. Dla permutacji σ, μ mamy zdefiniowany multizbiór $C_{\sigma,\mu} = \{s_1, \ldots, s_k\}$ (też uporządkowany niemalejąco: $s_1 \leq \ldots \leq s_k$).

Otoczenia definiujemy następująco:

$$N(\sigma, \mu) = \{(\tau, \mu) \in S_n \times S_m \mid \tau \in N_1(\sigma)\} \cup \{(\sigma, \rho) \in S_n \times S_m \mid \rho \in N_2(\mu)\}.$$

Otoczenia N_1 i N_2 są zdefiniowane tak jak dla problemu komiwojażera w Sekcji 6.2.2 (odpowiednio, dla S_n i S_m). Wszystkie niezbędne warunki są oczywiście spełnione.

Jako funkcję kosztu przyjmujemy

$$f(\sigma, \mu) = \sum_{i=1}^{k} \frac{(s_i - c_i)^2}{c_i}.$$

Minimum tej funkcji to $f(\sigma, \mu) = 0$.