Wykład 3

Heurystyki lokalnych ulepszeń w optymalizacji parametrycznej

Kazimierz Grygiel

Optymalizacja parametryczna

- Dziedzina: podzbiór zwarty (domknięty i ograniczony) przestrzeni euklidesowej $D \subset \mathcal{R}^m$, zazwyczaj określony przez układ *więzów* postaci $g_i(x) \geqslant 0$
- Funkcja oceny (celu) $f : D \to \mathcal{R}$, co najmniej ciągła (najczęściej różniczkowalna)
- W ogólnym przypadku (zadanie *programowania nieliniowego*) nie są znane uniwersalne metody wyznaczania optimum globalnego
- Stosując metody *numeryczne*, zadowalamy się rozwiązaniami przybliżonymi
- Jedno z możliwych podejść: dyskretyzacja zadania

  Polega na zastąpieniu dziedziny D ε-siecią D_ε pokrywającą D, tj. skończonym podzbiorem $D_\varepsilon \subset D$ o tej własności, że dla każdego punktu $x \in D$ istnieje węzeł sieci $x' \in D_\varepsilon$ taki, że $\| x - x' \| < \varepsilon$)
- Tak zredukowane zadanie można rozwiązać przy użyciu algorytmu przeszukiwania (ogólnie bez gwarancji dobrej aproksymacji)

Przykłady

- Funkcja Ackleya

$$f(x) = -a \exp\left(-b\sqrt{\frac{1}{m}\sum_{i=1}^{m} x_i^2}\right) - \exp\left(\frac{1}{m}\sum_{i=1}^{m}\cos(2\pi x_i)\right) + a + \exp(1)$$

- Funkcja Griewangka

$$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\frac{x_i}{\sqrt{i}}$$

- Funkcja Rastrigina

$$f(x) = \sum_{i=1}^{n} (x_i^2 - a\cos 2\pi x_i) + a\ n$$

- Funkcja Rosenbrocka

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$$

Home Page
Title Page
Page 3 of 22
Go Back
Full Screen
Close
Quit

Reprezentacja rozwiązań

- Rozwiązanie: wektor m-wymiarowy o współrzędnych rzeczywistych
- Problem reprezentacji liczby rzeczywistej:
  - stałopozycyjna (skala równomierna, błąd bezwględny)
  - zmiennopozycyjna (skala nierównomierna, błąd względny)
- Binarna reprezentacja stałopozycyjna liczby naturalnej z zakresu $0..2^L - 1$

$$n(b) = \sum_{i=0}^{L-1} b_i 2^i$$

L — długość reprezentacji

- Binarna reprezentacja stałopozycyjna liczby rzeczywistej z przedziału $[0, 1]$ (funkcje dekodujące):

(a)

$$r_A(b) = \sum_{i=0}^{L-1} b_i 2^{i-L} + 2^{-(L+1)} = \frac{n(b) + 0.5}{2^L}$$

Dokładność reprezentacji: $\varepsilon_A = 0.5/2^L$

(b)

$$r_B(b) = \frac{1}{2^L - 1} \sum_{i=0}^{L-1} b_i 2^i = \frac{n(b)}{2^L - 1}$$

Dokładność reprezentacji: $\varepsilon_B = 0.5/(2^L - 1)$

Ograniczenia „kostkowe”

- Reprezentacja L-bitowa liczby r z przedziału $[u, v]$ (funkcja dekodująca):

$$h(b) = u + (v - u) \cdot r(b)$$

  Dokładność reprezentacji: $\epsilon = (v - u)\varepsilon$

- Kostka m-wymiarowa:

$$\{x = (x_1, x_2, \ldots, x_m) : u_j \leqslant x_j \leqslant v_j,\ \ j = 1, \ldots, m\}$$

- Reprezentacje stałopozycyjne dla ograniczeń kostkowych
  - „jednochromosomowa”: $b = b^{(1)} \circ b^{(2)} \circ \ldots \circ b^{(m)}$ (konkatencja wektorów zerojedynkowych; każdej zmiennej może odpowiadać segment innej długości, funkcja dekodująca najpierw „rozpakowuje” wektor na poszczególne segmenty)
  - „wielochromosomowa”: $\vec{b} = (b^{(1)}, b^{(2)} \ldots, b^{(m)})$ (wektor wektorów zerojedynkowych; każdej zmiennej może odpowiadać wektor innej długości)

Zasięg operatora mutacji

- *Zasięg bezpośredni*:
  — jak daleko można „przeskoczyć” w przestrzeni poszukiwań przy *jednokrotnym* zastosowaniu operatora mutacji?
- *Zasięg pośredni*:
  — jak daleko można „zawędrować” w przestrzeni poszukiwań przy *wielokrotnym* zastosowaniu operatora mutacji?
- Precyzyjniej:
  — jakie jest prawdopodobieństwo, że przeskoczymy/zawędrujemy do punktu należącego do zadanego podzbioru przestrzeni poszukiwań?
- Na te pytania odpowiemy sobie w sposób poglądowy...

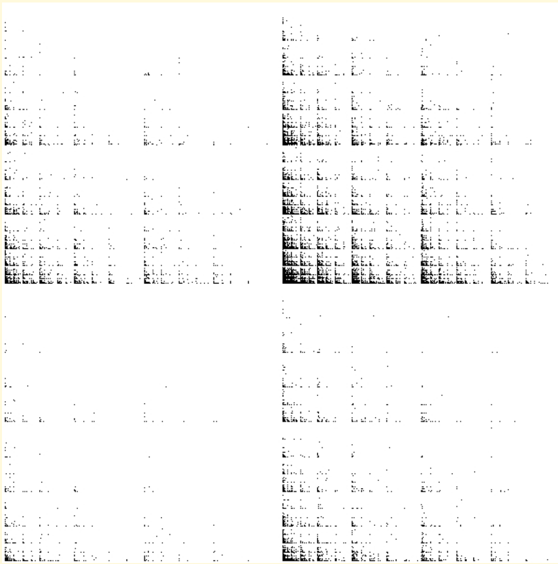
„Klify Hamminga”

- Niezgodność metryki Hamminga i metryki euklidesowej dla kodowania pozycyjnego: punkty bliskie na prostej mogą być odległe w B_m
- Przykład:

$$\begin{array}{rcl} n(01\ldots1) & = & 2^{L-1}-1 \\ n(10\ldots0) & = & 2^{L-1} \end{array}$$

  - odległość euklidesowa: 1
  - odległość Hamminga: L
- Prawdopodobieństwo przejścia przy mutacji Bernoulliego: p^L (a więc np. dla $p = 1/L$ jest ono równe $1/L^L$)
- Zjawisko nazwano „klifami Hamminga” przez analogię do trudności, na jakie natrafiłby wędrowiec próbując pokonać przeszkodę
- Wniosek: *Kodowanie pozycyjne nie jest odpowiednie przy stosowaniu operatora mutacji Bernoulliego*

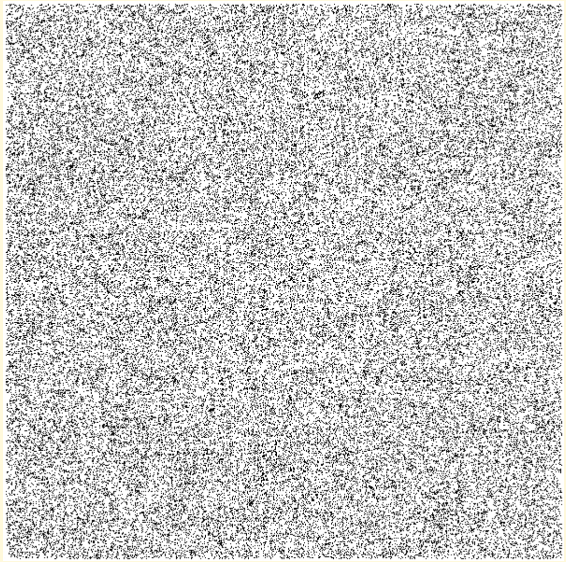
Mutacja Bernoulliego: zasięg bezpośredni w metryce euklidesowej



Zaczerniony obszar odpowiada punktom z kwadratu $[0,1] \times [0,1]$, otrzymanym przez zastosowanie operatora mutacji Bernoulliego do reprezentacji pozycyjnej punktu $(X_0, Y_0) = (0.5, 0.5)$.

100 tys. powtórzeń, $p = 0.2$

Mutacja Bernoulliego: zasięg pośred-ni w metryce euklidesowej



Zaczerniony obszar odpowiada punktom z kwadratu $[0,1] \times [0,1]$, otrzymanym przez zastosowanie operatora mutacji Bernoulliego do reprezentacji pozycyjnych kolejnych punktów (X_i, Y_i) przy $(X_0, Y_0) = (0.5, 0.5)$. 100 tys. iteracji, $p = 0.2$

Co widać z tych rysunków?

- Operator mutacji Bernoulliego w metryce euklidesowej ma charakter stochastycznie lokalno-globalny
- Operator mutacji Bernoulliego penetruje całą przestrzeń rozwiązań (gdy nie napotka przeszkód!)
- Skąd biorą się przeszkody?
- Wiele interesujących funkcji rzeczywistych charakteryzuje się regularnością (np. tendencją do wzrostu) w metryce euklidesowej, nie w metryce Hamminga dla reprezentacji pozycyjnej
- „Klify Hamminga” zmieniają „krajobraz”, rozrywając „naturalne drogi pod górkę” i wprowadzają sztuczne optima lokalne
- Ale działa to też w drugą stronę: punkty odległe w metryce euklidesowej mogą być bliskie w metryce Hamminga, co daje możliwość szybkiej penetracji „odległych regionów”
- Czy można zatem znaleźć kompromis?

Lustrzany kod Graya

- Idea: zmodyfikować kodowanie tak, żeby punkty bliskie w metryce euklidesowej miały kody bliskie w metryce Hamminga
- Taką własność mają *kody Graya*
- Lustrzany kod Graya tworzymy następująco:

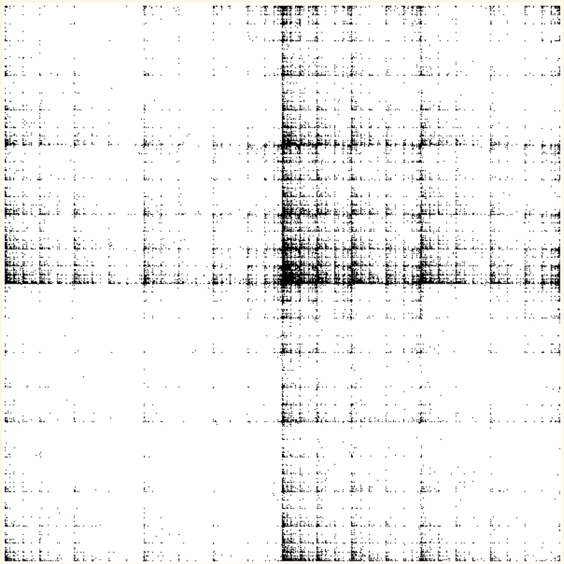
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0

- Z metody konstrukcji wynika, że ma on żądaną własność

Algorytm dekodowania dla kodu Graya

- Aby zastosować kodowanie Graya, musimy umieć „przetłumaczyć” go na kod pozycyjny
- Niech
$$g = (g_{L-1}, \ldots, g_1, g_0)$$
reprezentacja liczby n w kodzie lustrzanym Graya,
$$b = (b_{L-1}, \ldots, b_1, b_0)$$
reprezentacja liczby n w kodzie pozycyjnym
- Wtedy
$$b_{L-i} = \bigoplus_{j=1}^{i} g_{L-j} \text{ dla } i = 1, \ldots L$$
- Inaczej:
$$\begin{aligned} b_{L-1} &= g_{L-1} \\ b_j &= b_{j+1} \oplus g_j \end{aligned}$$
dla $j = L-2, \ldots, 0$

Mutacja Bernoulliego: zasięg bezpośredni (kod Graya)



Zaczerniony obszar odpowiada punktom z kwadratu $[0, 1] \times [0, 1]$, otrzymanym przez zastosowanie operatora mutacji Bernoulliego do reprezentacji w kodzie Graya punktu $(X_0, Y_0) = (0.5, 0.5)$.

100 tys. powtórzeń, $p = 0.2$

Arytmetyczna mutacja Bernoulliego

K. Grygiel, Algorytmy ewolucyjne z AB-mutacją, IV KKAEiOG, 2000

- Istota pomysłu: dodawanie zamiast „nakładania” maski
- Efekt jest następujący:

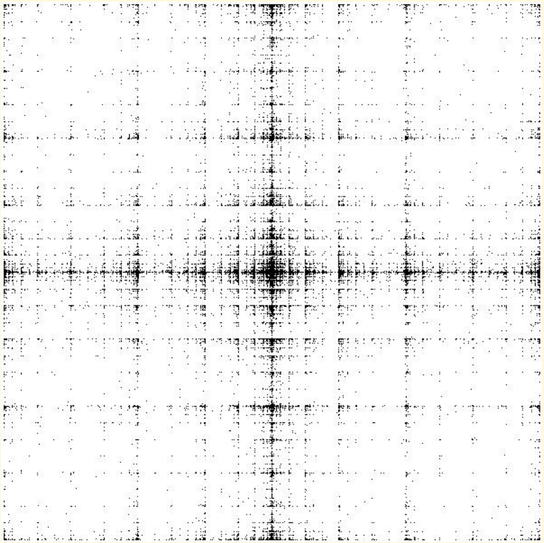
$$n(\phi_\xi(b)) = n(b) \pm n(\xi) \pmod{2^L}$$

  gdzie maska mutacyjna ξ jest tworzona jak w przypadku mutacji Bernoulliego, przy czym znak ($+$ lub $-$) jest losowany niezależnie od ξ z prawdopodobieństwem $1/2$, a operacja modulo jest wykonywana, aby uniknąć przekroczenia zakresu reprezentacji (zatem dziedzinę traktujemy tu jako wielowymiarowy torus)

- W przypadku funkcji wielu zmiennych należy użyć reprezentacji „wielochromosomowej” (każdy argument mutowany niezależnie)
- Operator jest prosty i efektywny w realizacji, jeśli stosujemy implementację całkowitoliczbową reprezentacji binarnej (zob. dalej)

Home Page
Title Page
◂◂ ▸▸
◂ ▸
Page 14 of 22
Go Back
Full Screen
Close
Quit

AB-mutacja: zasięg bezpośredni w metryce euklidesowej



Zaczerniony obszar odpowiada punktom z kwadratu $[0,1] \times [0,1]$, otrzymanym przez zastosowanie operatora arytmetycznej mutacji Bernoulliego do reprezentacji pozycyjnej punktu $(X_0, Y_0) = (0.5, 0.5)$. 100 tys. powtórzeń, $p = 0.2$

Implementacje

- Dekodowanie do przedziału $[0, 1]$ (funkcja r_A):

```
// N = 2^L
function decode(b: Chromosom): Real;
begin
   decode := (b+0.5) / N
end;
```

- Funkcja dekodująca Graya:

```
function invGray (j: Integer): Integer;
var i: Integer;
begin
   i := j;
   while j > 1 do begin
      j := j shr 1; i := i xor j
   end;
   invGray := i
end;
```

Implementacje, cd

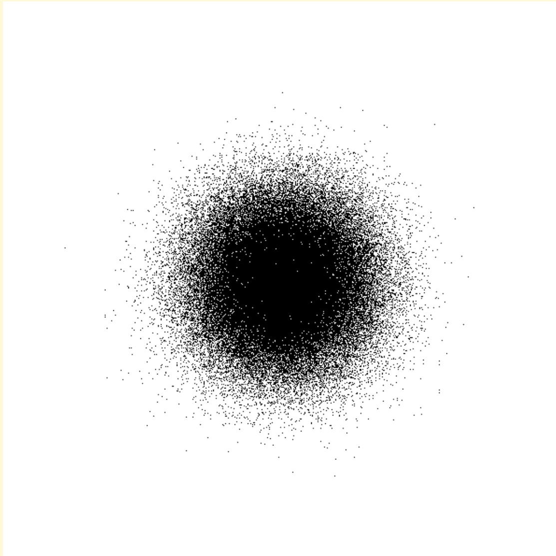
- Zegar mutacyjny:

  Dodając przesunięcie wygenerowane wg rozkładu geometrycznego podczas tworzenia maski (masek), wygodnie jest potraktować reprezentację rozwiązania jako pojedynczą sekwencję, nawet jeśli faktycznie obejmuje ona z wiele składowych (tj. ponumerować jednolicie wszystkie bity reprezentacji). Oba możliwe podejścia są statystycznie równoważne ze względu na „bezpamięciowość” rozkładu geometrycznego.

- AB-mutacja:

```
// dodawanie maski do chromosomu (b):
// N = 2^L
if mask <> 0 then
    if Random < 0.5 then
        b := (N-mask+b) mod N
    else
        b := (mask+b) mod N;
```

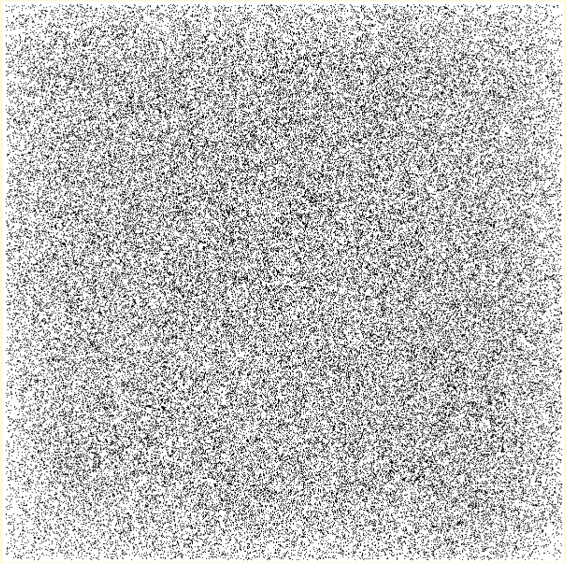
Mutacja gaussowska: zasięg bezpośredni



Zaczerniony obszar odpowiada punktom z kwadratu $[0,1] \times [0,1]$, otrzymanym przez zastosowanie operatora mutacji gaussowskiej do reprezentacji rzeczywistoliczbowej punktu $(X_0, Y_0) = (0.5, 0.5)$.

100 tys. powtórzeń, $\sigma = 0.125$

Mutacja gaussowska: zasięg pośredni

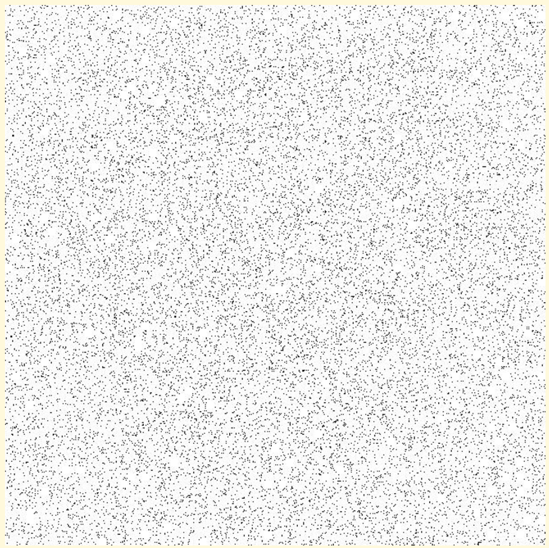


Zaczerniony obszar odpowiada punktom z kwadratu $[0,1] \times [0,1]$, otrzymanym przez zastosowanie operatora mutacji gaussowskiej do reprezentacji rzeczywistoliczbowych kolejnych punktów (X_i, Y_i) przy $(X_0, Y_0) = (0.5, 0.5)$.

100 tys. iteracji, $\sigma = 0.125$

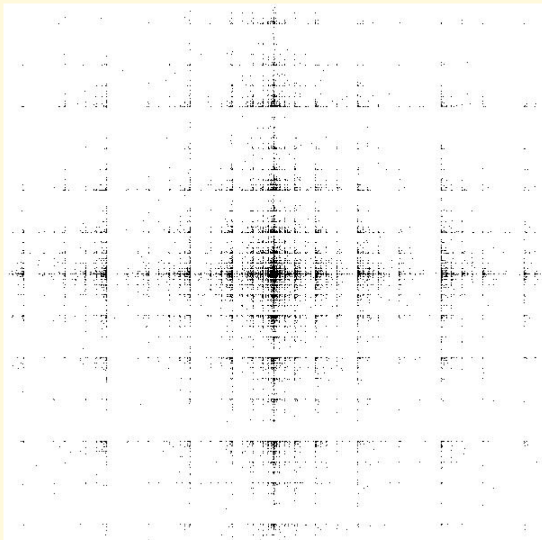
Home Page
Title Page
Page 19 of 22
Go Back
Full Screen
Close
Quit

Jak gęste jest “sito” mutacji gaussowskiej?



Zasięg bezpośredni mutacji gaussowskiej w powiększeniu 10-krotnym (w otoczeniu punktu centralnego)

A jak to wygląda dla arytmetycznej mutacji Bernoulliego?



Zasięg bezpośredni arytmetycznej mutacji Bernoulliego w powiększeniu 10-krotnym (w otoczeniu punktu centralnego)

Wnioski

- Operator mutacji Gaussa w metryce euklidesowej ma charakter stochastycznie lokalny
- Operator mutacji Gaussa penetruje całą przestrzeń rozwiązań
- Operator mutacji Gaussa „degeneruje się” do błądzenia przypadkowego w skali odległości rzędu σ
- Czy można to poprawić?
- Pomysł: schemat mutacji adaptacyjnej (automatyczna zmiana wielkości σ w zależności od stopnia skuteczności trafień) — zajmiemy się tym później