

Modyfikacje Symulowanego Wyżarzania

Wojciech Grabias

March 2023

Problem optymalizacyjny

Przedstawiony problem optymalizacyjny polegać będzie na znalezieniu takiego $x \in \mathcal{X}$, dla którego funkcja celu f przyjmuje wartość minimalną.

T – Temperatura

$f : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ – Funkcja celu, $\mathcal{X} \subset \mathbb{R}^d$

Rozkład Boltzmann

$$\pi_T(x) = \frac{1}{Z_T} \exp\left(-\frac{f(x)}{T}\right)$$

Intuicja:

Dla małych T , z dużym prawdopodobieństwem x przyjmuje tylko takie stany, że wartość $f(x)$ jest mała.

Dla funkcji celu f :

$$S_* = \{x \in \mathcal{X} : f(x) = 0\}$$

$$S_\epsilon = \{x \in \mathcal{X} : f(x) \leq \epsilon\}, \epsilon > 0$$

$\mathcal{M}(\mathcal{X})$ – zbiór prawdopodobieństw na $(\mathcal{X}, \mathfrak{B}(\mathcal{X}))$

$G(x, dy)$ – Będziemy je traktowali jako rozkład normalny o średniej x

Algorithm 1: SA

Initialization with $x_0 \sim \mu_0$, $\mu_0 \in \mathcal{M}(\mathcal{X})$

for $k = 0, \dots$ **do**

 Generate a candidate $y_k \sim G(x_k, dy)$

 Compute the acceptance probability

$$p_k = \exp \left(- \left(\frac{f(y_k) - f(x_k)}{T_k} \right)_+ \right)$$

 Set $x_{k+1} = \begin{cases} y_k & \text{with probability } p_k \\ x_k & \text{with probability } 1 - p_k \end{cases}$

end

Algorytm 1: SA, [2]

Metropolis-Hastings (MH) kernel

$$P_k(x, dy) = p_k(y, x)G(x, dy) + (1 - r(x))\delta_x(dy), \text{ gdzie}$$

$$r(x) = \int_{\mathcal{X}} p_k(y, x)G(x, dy)$$

Wówczas zdefiniować możemy:

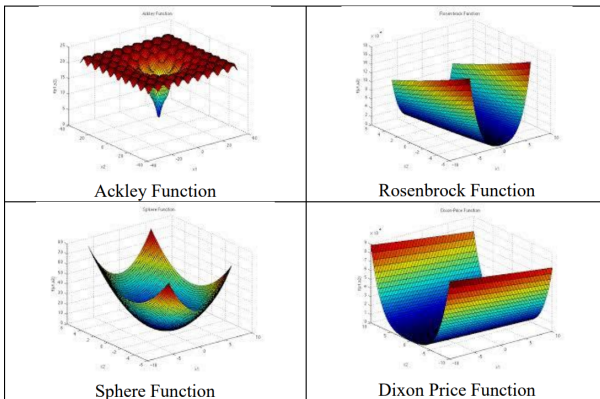
$$\mu_k(dx) = \mathbb{P}(x_k \in dx) = \mu_{k-1}P_k, \quad k > 1$$

Twierdzenie

Jeżeli T maleje logarytmicznie z dokładnością do pewnej stałej, to przy odpowiednio dużym T_0 :

$$\lim_{k \rightarrow \infty} \mathbb{P}(x_k \in S_\epsilon) = 1$$

Efektywność modyfikacji - przykład I-MSAA



Obrazek 1: Optymalizowane funkcje, [3]

Efektywność modyfikacji - przykład I-MSAA

| Function | D | Optimal | SA | | | I-MSAA | | |
|-------------|----|---------|-------------------|-------------------|-------------------|--------|--------|----------------------|
| | | | Best | Worse | SD | Best | Worse | SD |
| Dixon Price | 10 | 0.0 | 100.3 | 891.4 | 1.9×10^2 | 0.0 | 0.0 | 8.9×10^{-8} |
| Ackley | 50 | 0.0 | 17.9 | 19.0 | 0.2 | 0.0 | 0.0 | 2.3×10^{-5} |
| Neumaier | 10 | -210.0 | 764.8 | 3.6×10^3 | 6.4×10^2 | -210.0 | -210.0 | 3.2×10^{-9} |
| Rosenbrock | 50 | 0.0 | 5.7×10^5 | 9.4×10^5 | 8.3×10^4 | 0.0 | 4.0 | 1.2 |

Tabela 1: Porównanie SA i I-MSAA,[3]

FSA- Fast Simulated Annealing

FSA - zamysł

FSA bierze swoją nazwę od przyspieszonego schematu chłodzenia - chcemy szybciej osiągać niskie temperatury.

Aby kontrolować zachowanie algorytmu stosować będziemy funkcje malejące wolniej od dotychczasowo przyjętej e^{-x}

Generalizacja SA o dowolność funkcji q określającej prawdopodobieństwo przyjęcia nowego punktu y_k (ang. acceptance function):

$$p_k = q(\rho_k), \text{ gdzie} \\ \rho_k = \left(\frac{f(y_k) - f(x_k)}{T_k} \right)_+$$

Dozwolone są jednak funkcje malejące wolniej od e^{-x} .

Zwykle przyjmuje się, że $\rho \mapsto q(\rho) = \frac{1}{1+\rho}$

Tak zmodyfikowane jądra przejścia oznaczać będziemy $P_k^{(F)}$

Twierdzenie*

Jeżeli dla pewnego $\gamma \in (0, 1]$:

$$T_k = \frac{1}{(k+1)^\gamma \log((k+1)^\gamma)}$$

to istnieje $C_\epsilon > 0$ spełniające:

$$\mathbb{P}(x_k \in S_\epsilon) \geq 1 - \frac{C_\epsilon}{(k+1)^\gamma}, \forall k \in \mathbb{N}$$

SMC-SA: Sequential Monte Carlo-Simulated Annealing

SMC-SA - zamysł

Zamiast szukać minimum pojedynczym punktem, szukajmy go kilkoma punktami jednocześnie, moderując ich zachowanie w każdej iteracji.

SMC-SA: Sequential Monte Carlo-Simulated Annealing

Aktualny zbiór punktów

Do dyspozycji mamy n punktów x_1, \dots, x_n

Przypisanie wag każdemu z punktów

Każdy z punktów otrzymuje swoją wagę w oparciu o stosunek rozkładów Boltzmann'a

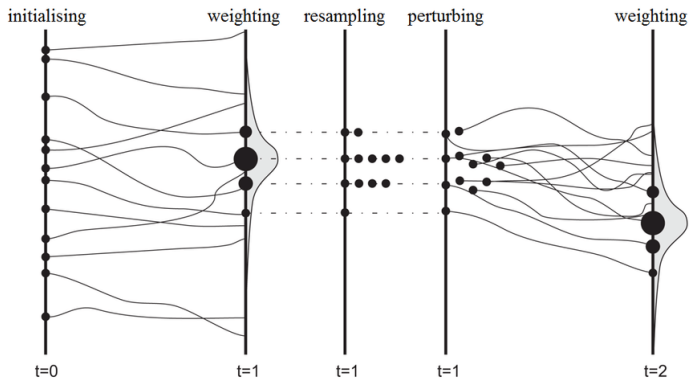
Resampling

Na podstawie wag z odpowiednim prawdopodobieństwem wybieramy odpowiednie punkty (mogą się powtarzać), nowy zbiór punktów może być większy od poprzedniego

Zmiana położenia punktów

Każdy z nowych punktów przepuszczamy przez kernel MH - P_k

SMC-SA: Schemat działania



Obrazek 2: Wizualizacja resamplingu na podstawie rozkładu normalnego, [1]

Algorithm 2: SMC-SA

Initialize the algorithm $x_k^{(n)} \sim \mu_0$ for $1 \leq n \leq N_0$;

for $k = 1, \dots$ **do**

 Compute the self-normalized weights

$$w_k^{(n)} \propto \frac{\pi_k}{\pi_{k-1}}(x_{k-1}^{(n)})$$

 Resample $\{\tilde{x}_k^{(n)}\}_{n=1}^{N_k}$ from $\{x_{k-1}^{(n)}, w_k^{(n)}\}_{n=1}^{N_{k-1}}$

 Generate $\{x_k^{(n)}\}_{n=1}^{N_k}$ propagating the points

$\{\tilde{x}_k^{(n)}\}_{n=1}^{N_k}$ with the MH kernel $P_k(x, dy)$

end

Algorytm 2: SMC-SA, [2]

Podsumowanie 3 modyfikacji SA

SA

Podstawowa wersja: p_k ściśle związane z π_k , wolne tempo chłodzenia

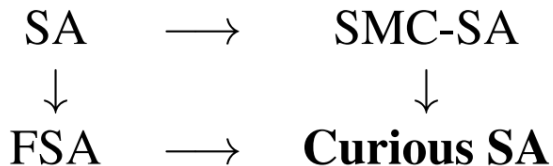
FSA

Szybsze tempo chłodzenia, wolniej malejąca funkcja wyznaczająca prawdopodobieństwa zmiany stanu

SMC-SA

Rozszerzenie SA o wielopunktowość, kontrola punktów poprzez narzucenie rozkładu

Curious Simulated Annealing



Algorithm 3: CSA

Initialize the algorithm $x_k^{(n)} \sim \mu_0$ for $1 \leq n \leq N$;

for $k = 1, \dots$ **do**

 Compute the self-normalized weights

$$w_k^{(n)} \propto \frac{\pi_k}{\pi_{k-1}}(x_{k-1}^{(n)})$$

 Resample $\{\tilde{x}_k^{(n)}\}_{n=1}^N$ from $\{x_{k-1}^{(n)}, w_k^{(n)}\}_{n=1}^N$

 Generate $\{x_k^{(n)}\}_{n=1}^N$ propagating the points

$\{\tilde{x}_k^{(n)}\}_{n=1}^N$ with the MH kernel $P_k^{(F)}(x, dy)$

end

Algorytm 3: CSA, [2]

Problem 1

Minimalizacja funkcji Rosenbrock'a w \mathbb{R}^{10} :

$$f(x) = \sum_{i=1}^9 5(x_{i+1} - x_i^2)^2 + (1 - x_i)^2, \quad \forall x \in \mathbb{R}^{10}$$

| | | SA | FSA | SMC-SA | CSA |
|---------|-----------------------------|-------|--------------|--------|--------------|
| (P_1) | $\langle f_{50}^* \rangle$ | 6.31 | 6.49 | 6.41 | 4.05 |
| | σ_{50}^* | 0.829 | 0.732 | 1.15 | 1.17 |
| | $\langle f_{500}^* \rangle$ | 3.64 | 3.72 | 5.06 | 2.19 |
| | σ_{500}^* | 0.761 | 0.778 | 1.26 | 0.447 |

Tabela 2: Porównanie testów numerycznych, [2]

Problem 2

Minimalizacja funkcji Rastrigin'a z równomiernie rozłożonymi minimami:

$$f(x) = \sum_{i=1}^{10} x_i^2 - \cos(2\pi x_i), \forall x \in \mathbb{R}^{10}$$

| | | SA | FSA | SMC-SA | CSA |
|---------|-----------------------------|--------------|--------------|--------|-------------|
| (P_2) | $\langle f_{50}^* \rangle$ | 3.29 | 3.36 | 3.26 | 3.23 |
| | σ_{50}^* | 0.425 | 0.453 | 0.521 | 0.484 |
| | $\langle f_{500}^* \rangle$ | 2.52 | 2.64 | 2.62 | 2.47 |
| | σ_{500}^* | 0.320 | 0.304 | 0.413 | 0.502 |

Tabela 3: Porównanie testów numerycznych, [2]

Dziękuję za uwagę :)



Danilo Alvares, Carmen Armero, Anabel Forte, and Nicolas Chopin.

Sequential monte carlo methods in bayesian joint models for longitudinal and time-to-event data.

Statistical Modelling, 21(1-2):161–181, 2021.



Emilie Chouzenoux, Víctor Elvira, and Thomas Guilmeau.

Simulated annealing: a review and a new scheme.

PGMO DAYS 2021, page 33, 2021.



Jesús Suarez, Carlos Millan, and Euriel Millan.

Improved modified simulated annealing algorithm for global optimization.

Contemporary Engineering Sciences, 11(96):4789–4795, 2018.