

ZMOG: Przestrzenie przeszukiwania i metody jednopunktowe

Michał Okulewicz

Wydział Matematyki i Nauk Informatycznych
Politechnika Warszawska

Plan wykładu

- 1 Algorytmy optymalizacyjne
 - Problemy optymalizacyjne
 - Klasyfikacja algorytmów
 - Święty Graal optymalizacji
- 2 Przestrzenie i operatory przeszukiwania
 - Przestrzenie ciągłe
 - Przestrzenie dyskretne i kombinatoryczne
- 3 Metody jednopunktowe
 - Monte Carlo
 - Losowe błądzenie
 - Hill climbing
 - Symulowane wyżarzanie
 - Variable Neighbourhood Search

Definicja (Problem optymalizacyjny)

Problem optymalizacyjny $\mathcal{P} = (\Omega, f, \succ)$ jest zdefiniowany przez przestrzeń rozwiązań Ω , funkcję jakości^a $f : \Omega \rightarrow \mathbb{R}$ oraz relację $\succ \in \{<, >\}$. Rozwiązaniem problemu jest znalezienie zbioru elementów (elementu) $\mathcal{X} \subseteq \Omega$ spełniającego następujące warunki:

$$\mathcal{X} = \{x \in \Omega : \forall x' \in \Omega f(x) \succeq f(x')\}$$

^aW zależności od dziedziny funkcja jakości może być też określana jako funkcja celu lub funkcja przystosowania.

Definicja (Podzbiór dopuszczalny)

Podzbiorem dopuszczalnym zbioru Ω jest zbiór \mathcal{D} spełniający następujące warunki:

$$\mathcal{D} = \{x \in \Omega : f(x) \text{ ma sens matematyczny i/lub biznesowy}\}$$

Algorytmy optymalizacyjne I

Poszukiwanie rozwiązania:

- Konstrukcyjne
- Iteracyjnego poprawiania

Gwarancja jakości rozwiązania:

- Heurystyczne
- Aproksymacyjne
- Probabilistyczne
- Dokładne

Zakres przeszukiwania zbioru rozwiązań:

- Lokalne

Algorytmy optymalizacyjne II

- Globalne

Sposób wykorzystania funkcji jakości:

- Gradientowe
- Próbkujące

Wiedza o funkcji jakości:

- Jednopunktowe
- Populacyjne

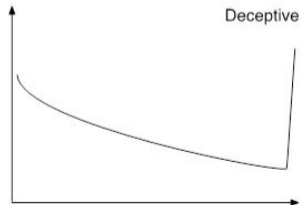
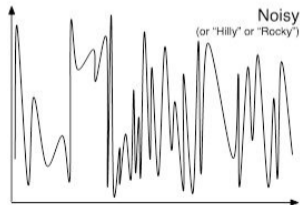
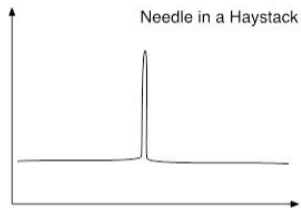
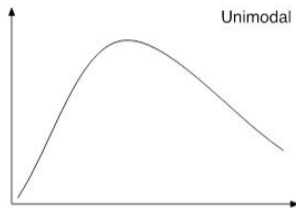
Definicja (Metaheurystyki)

Za podręcznikiem i wykładami prof. Arabasa [1, 2], **metaheurystyka** będzie rozumiana jako pewien ramowy (niezależny od rozwiązywanego problemu) sposób przeszukiwania przestrzeni rozwiązań. W sposobie tym należy określić (zależne już od problemu) kodowanie rozwiązania, operatory przeszukiwania tej przestrzeni oraz rozmiar pamięci algorytmu.





<http://slideplayer.com/slide/4579308/>



Przestrzenie przeszukiwania I



Przestrzenie przeszukiwania II

Definicja

Przestrzenią przeszukiwań \mathcal{G} problemu \mathcal{P} jest zbiór, na którym zdefiniowana jest operacja $dec : \mathcal{G} \rightarrow \Omega$ transformująca element przestrzeni przeszukiwań $g \in \mathcal{G}$ w element przestrzeni rozwiązań $x \in \Omega$.

Przestrzenie przeszukiwania III

Definicja

Algorytm optymalizacyjny \mathcal{A} to zbiór operatorów przeszukiwania \mathcal{O} pracujących na elementach przestrzeni przeszukiwań \mathcal{G} oraz wykorzystujący złożenie $f \circ \text{dec} : \mathcal{G} \rightarrow \mathbb{R}$ do ewaluacji elementów tej przestrzeni i zwracający zbiór \mathcal{X}^* , będący aproksymacją poszukiwanego zbioru \mathcal{X} o następujących cechach:

$$\mathcal{X}^* = \{x \in \Omega_{\mathcal{A}} : \forall_{x' \in \Omega_{\mathcal{A}}} f(x) \succeq f(x')\}$$

$\Omega_{\mathcal{A}}$ jest podzbiorem przestrzeni rozwiązań przeliczonych przez algorytm \mathcal{A} w trakcie jego działania. Operator $O \in \mathcal{O}$ jest postaci $O : \mathcal{P} \times \mathcal{G}^{n_0} \rightarrow \mathcal{G}^{n_1}$, gdzie $n_0 \in \mathbb{N}_0$, $n_1 \in \mathbb{N}_+$.

Przykład

Przykładem operatora (ozn. I) korzystającego wyłącznie z danych problemu i generującego element przestrzeni przeszukiwań $I : P \rightarrow \mathcal{G}$ są wszelkiego rodzaju inicjalizacje rozwiązań oraz algorytmy jednokrokowe.

Przykład

Przykładem operatora (ozn. M) korzystającego z pojedynczego elementu przestrzeni przeszukiwań i zwracającego jeden taki element $M : \mathcal{G} \rightarrow \mathcal{G}$ jest mutacja w algorytmie genetycznym.

Przykład

Przykładami operatorów wykorzystujących wiele elementów przestrzeni przeszukiwań i zwracających jeden lub wiele elementów jest krzyżowanie wymieniające (ozn. C) w algorytmie genetycznym $C : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G} \times \mathcal{G}$, czy zmiana prędkości (ozn. Δ_V) w optymalizacji rojem cząstek $\Delta_V : \mathcal{G} \times \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$.

Przestrzenie ciągłe

Definicja

Problem ciągły charakteryzuje się tym, że jego rozwiązanie jest reprezentowane w formie wektora liczb rzeczywistych ($\Omega = \mathbb{R}^n$).

Przestrzenie ciągłe

Zbiór rozwiązań:

- \mathbb{R}^n

Typowe zastosowania:

- Procesy sterowania

Typowe operatory:

- Gaussowskie zaburzenie
- Uśrednianie rozwiązań

Przestrzenie dyskretne i kombinatoryczne

Definicja

Problem dyskretny charakteryzuje się tym, że jego rozwiązanie jest reprezentowane w formie wektora liczb całkowitych ($\Omega = \mathbb{Z}^n$).

Definicja

Problem kombinatoryczny charakteryzuje się tym, że jego rozwiązanie jest reprezentowane w formie wektora binarnego ($\Omega = \mathbb{Z}_2^n$).

Przestrzenie dyskretne i kombinatoryczne

Zbiór rozwiązań:

- $\{0, 1\}^n$
 - Grafy (w tym np. sieci neuronowe)
- \mathbb{Z}^n
 - Słowa

Typowe zastosowania:

- Problemy logistyczne, produkcyjne i transportowe

Typowe operatory:

- Przełączanie bitów
- Permutacje
- Wymiany

Metody jednopunktowe

- Wykorzystują lokalne operatory
- Mogą być uruchamiane równoległe

Monte Carlo

- Najprostsza metoda w implementacji
- Losowanie kolejnych rozwiązań z przestrzeni problemu / przestrzeni przeszukiwania
- Przydatne minimalne założenia dotyczące lokalizacji zbioru dopuszczalnego

Monte Carlo

- Najprostsza metoda w implementacji
- Losowanie kolejnych rozwiązań z przestrzeni problemu / przestrzeni przeszukiwania
- Przydatne minimalne założenia dotyczące lokalizacji zbioru dopuszczalnego

Monte Carlo

- Najprostsza metoda w implementacji
- Losowanie kolejnych rozwiązań z przestrzeni problemu / przestrzeni przeszukiwania
- Przydatne minimalne założenia dotyczące lokalizacji zbioru dopuszczalnego

Monte Carlo

- Najprostsza metoda w implementacji
- Losowanie kolejnych rozwiązań z przestrzeni problemu / przestrzeni przeszukiwania
- Przydatne minimalne założenia dotyczące lokalizacji zbioru dopuszczalnego

Losowe błądzenie

- Pozornie bezsensowna modyfikacja metody Monte Carlo
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie
- Jeżeli znamy przybliżony rozkład wartości funkcji możemy poprawić jakość działania algorytmu
- WPP równoważny metodzie Monte Carlo

Losowe błędzenie

- Pozornie bezsensowna modyfikacja metody Monte Carlo
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie
- Jeżeli znamy przybliżony rozkład wartości funkcji możemy poprawić jakość działania algorytmu
- WPP równoważny metodzie Monte Carlo

Losowe błędzenie

- Pozornie bezsensowna modyfikacja metody Monte Carlo
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie
- Jeżeli znamy przybliżony rozkład wartości funkcji możemy poprawić jakość działania algorytmu
- WPP równoważny metodzie Monte Carlo

Losowe błędzenie

- Pozornie bezsensowna modyfikacja metody Monte Carlo
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie
- Jeżeli znamy przybliżony rozkład wartości funkcji możemy poprawić jakość działania algorytmu
- WPP równoważny metodzie Monte Carlo

Losowe błędzenie

- Pozornie bezsensowna modyfikacja metody Monte Carlo
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie
- Jeżeli znamy przybliżony rozkład wartości funkcji możemy poprawić jakość działania algorytmu
- WPP równoważny metodzie Monte Carlo

Hill climbing

- Naturalne rozszerzenie metody losowego błędzenia
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie, które poprawiło wynik algorytmu
- Algorytm lokalnego przeszukiwania

Hill climbing

- Naturalne rozszerzenie metody losowego błędzenia
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie, które poprawiło wynik algorytmu
- Algorytm lokalnego przeszukiwania

Hill climbing

- Naturalne rozszerzenie metody losowego błędzenia
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie, które poprawiło wynik algorytmu
- Algorytm lokalnego przeszukiwania

Hill climbing

- Naturalne rozszerzenie metody losowego błędzenia
- Losowanie kolejnych rozwiązań w oparciu o poprzednio testowane rozwiązanie, które poprawiło wynik algorytmu
- Algorytm lokalnego przeszukiwania

Simulated Annealing

- Połączenie losowego błądzenia z *hill climbing*
- Nowe rozwiązania są akceptowane zawsze, jeżeli poprawiają poprzedni wynik
- Nowe rozwiązania są akceptowane z pewnym prawdopodobieństwem, nawet jeżeli pogarszają poprzedni wynik
- Zawężanie zakresu działania algorytmu

Simulated Annealing

- Połączenie losowego błądzenia z *hill climbing*
- Nowe rozwiązania są akceptowane zawsze, jeżeli poprawiają poprzedni wynik
- Nowe rozwiązania są akceptowane z pewnym prawdopodobieństwem, nawet jeżeli pogarszają poprzedni wynik
- Zawężanie zakresu działania algorytmu

Simulated Annealing

- Połączenie losowego błądzenia z *hill climbing*
- Nowe rozwiązania są akceptowane zawsze, jeżeli poprawiają poprzedni wynik
- Nowe rozwiązania są akceptowane z pewnym prawdopodobieństwem, nawet jeżeli pogarszają poprzedni wynik
- Zawężanie zakresu działania algorytmu

Simulated Annealing

- Połączenie losowego błądzenia z *hill climbing*
- Nowe rozwiązania są akceptowane zawsze, jeżeli poprawiają poprzedni wynik
- Nowe rozwiązania są akceptowane z pewnym prawdopodobieństwem, nawet jeżeli pogarszają poprzedni wynik
- Zawężanie zakresu działania algorytmu

Simulated Annealing

- Połączenie losowego błądzenia z *hill climbing*
- Nowe rozwiązania są akceptowane zawsze, jeżeli poprawiają poprzedni wynik
- Nowe rozwiązania są akceptowane z pewnym prawdopodobieństwem, nawet jeżeli pogarszają poprzedni wynik
- Zawężanie zakresu działania algorytmu

Variable Neighbourhood Search

- Alternatywny sposób przełamania lokalności algorytmu *hill climbing*
- Poszerzanie zakresu działania operatora modyfikującego rozwiązanie

Variable Neighbourhood Search

- Alternatywny sposób przełamania lokalności algorytmu *hill climbing*
- Poszerzanie zakresu działania operatora modyfikującego rozwiązanie





Variable Neighbourhood Search

- Alternatywny sposób przełamania lokalności algorytmu *hill climbing*
- Poszerzanie zakresu działania operatora modyfikującego rozwiązanie

Ćwiczenia praktyczne / praca domowa

- Zbadać szybkość zbieżności metody Monte Carlo, losowego błędzenia, *hill climbing* oraz SA lub VNS na funkcjach Rastrigina i Rosenbrocka
- Narysować wykres zbieżności
- Narysować przykładowy przebieg algorytmu dla funkcji 2D
- Sztuczne dane
- Tutorial

Bibliografia

-  Jarosław Arabas.
Wykłady z algorytmów ewolucyjnych.
Wydawnictwa Naukowo-Techniczne, 2004.
-  Jarosław Arabas.
Metaheuristics – a modern approach.
jun 2014.
-  Trung Thanh Nguyen, Shengxiang Yang, and Jürgen Branke.
Evolutionary dynamic optimization: A survey of the state of the art.
Swarm and Evolutionary Computation, 6:1–24, 08 2012.
-  Karsten Weicker.
Evolutionary Algorithms and Dynamic Optimization Problems.
PhD thesis, Universität Stuttgart, 2003.