**Politechnika Wrocławska, Automatyka i Robotyka manuscript No.**
(will be inserted by the editor)

# Optymalizacja globalna

**Piotr Wojna**
**238949**

## 1 Wstęp

Najważniejszym celem optymalizacji globalnej, zarówno w algorytmach multistartu jak i również w algorytmach bezpośrednio dedykowanych do poszukiwań optimum globalnego, jest zyskanie maksymalnej ilości informacji o rozważanym problemie. Co więcej, jednym z najważniejszych wskaźników jakości różnych metod optymalizacji globalnej, jest liczba obliczeń funkcji celu. Tak też, najbardziej efektywna strategia powinna pozyskiwać jak najwięcej informacji o nieznanej funkcji celu, za pomocą minimalnej ilości punktów testowych.

## 2 Stochastyczne algorytmy globalnej optymizacji

Metody przeszukiwań czysto losowych należą do tzw. stochastycznych, nieadaptacyjnych metod optymalizacji globalnej. Oznacza to, że kolejno generowane punkty testowe nie wykorzystują informacji o punktach wygenerowanych wcześniej. W przypadku braku jakiejkolwiek informacji o funkcji celu, kolejne punkty testowe generowane są za pomocą rozkładu równomiernego na zbiorze.
Często wybieramy algorytmy stochastyczne

- by uprościć wyznaczanie nowego kierunku poszukiwań ( w przypadku np. gradientu trzeba na każde oszacowanie gradientu obliczenia $2 * k, k$- wymiar zadania wartości funkcji. )
- uniknąć wyznaczania w każdym kroku iteracyjnym $N(\varepsilon)$ punktów $\varepsilon$ sieci, których liczba $N(\varepsilon)$ rośnie potęgowo z wymiarem zadania.

Jednym słowem często algorytmy stochastyczne są prostsze, ale z drugiej strony wymagają innego niż typowe podejścia probabilistycznego. W tym losowego wybierania punktów bądź kierunków a także stosowania probabilistycznego podejścia do zbieżności.

W praktyce idea konstrukcji algorytmów stochastycznych okazała się bardzo owocna. Znalazła ona liczne rozwinięcia w postaci nieraz bardzo wyszukanych algorytmów. Udało się skutecznie zastosować

Piotr Wojna
238949

takie algorytmy do rozwiązania licznych zadań programowania nieliniowego, przede wszystkim zadań inżynierskich.

## 3 Algorytmy genetyczne

W algorytmach genetycznych stosuje się pojęcia zapożyczone z genetyki naturalnej. Istnieje populacja rozwiązań. Każde rozwiązanie jest nazywane osobnikiem. Na populacji dokonujemy selekcji (słabe osobniki giną), krzyżowania (czyli tworzenia nowych osobników) i mutacji (dzika karta). W terminologii algorytmów genetycznych używa się następujących pojęć:

- **Populacja** – zbiór osobników o określonej liczebności.
- **Osobniki populacji** w algorytmach genetycznych to zakodowane w postaci chromosomów zbiory parametrów zadania, czyli rozwiązania, określane też jako punkty przestrzeni poszukiwań.
- **Chromosomy** – inaczej łańcuchy lub ciągi kodowe – to uporządkowane ciągi genów.
- **Gen** – nazywany też cecha, znakiem, detektorem – jest to pojedynczy element genotypu, w szczególności chromosomu.
- **Genotyp, czyli struktura** - to zespół chromosomów danego osobnika. Osobnikami populacji mogą być genotypy albo pojedyncze chromosomy.
- **Fenotyp** – zestaw wartości odpowiadający danemu genotypowi, czyli zdekodowana struktura. Jest to zbiór parametrów zadania (rozwiązanie, punkt przestrzeni poszukiwań).
- **Allel** to wartość danego genu (określana też jako wartość cechy lub wariant cechy).
- **Locus** to położenia danego genu w łańcuchu czyli w chromosomie.
- **Funkcja przystosowania** – nazywana też funkcją dopasowania lub funkcją oceny. Stanowi ona miarę przystosowania (dopasowania) danego osobnika w populacji. Funkcja przystosowania pozwala ocenić stopień przystosowania poszczególnych osobników w populacji. Na tej podstawie wybiera się osobniki najlepiej przystosowane (o największej wartości funkcji przystosowania). Jest to zgodne z ewolucyjną zasadą przetrwania „najsilniejszych" (najlepiej przystosowanych). Ma ona duży wpływ na działanie algorytmów genetycznych i musi być odpowiednio zdefiniowana.
- **Generacja** – to kolejna iteracja w algorytmie genetycznym.
- **Pokolenie** (nowe pokolenie lub pokolenie potomków) – to nowoutworzona populacja osobników.

W algorytmie genetycznym, w każdej jego iteracji, jest oceniane przystosowanie każdego osobnika danej populacji za pomocą funkcji przystosowania. Na tej podstawie tworzona jest nowa populacja osobników, którzy stanowią zbiór potencjalnych rozwiązań problemu, np. zadania optymalizacji.

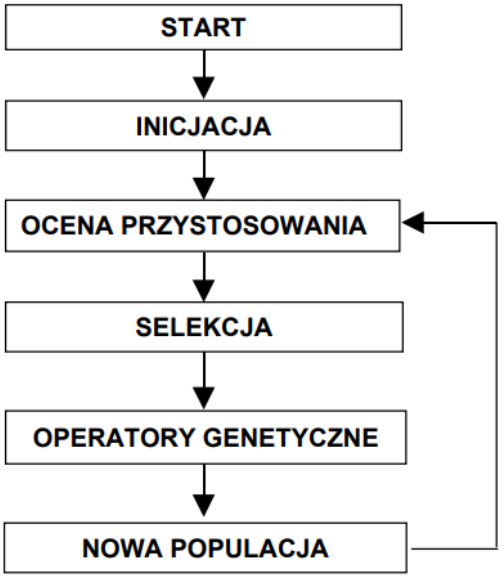
### 3.1 Klasyczny algorytm genetyczny

Mechanizm działania klasycznego (elementarnego, prostego) algorytmu genetycznego jest prosty. Polega na kopiowaniu ciągów i wymianie podciągów. Na rysunku 1 pokazano schemat działania algorytmu genetycznego, które składa się z następujących kroków [7] :

- **Inicjacja** – utworzenie populacji początkowej, poprzez losowy wybór ustalonej liczby chromosomów.
- **Ocena przystosowania** – obliczenie wartości funkcji przystosowania dla każdego chromosomu.
- **Selekcja chromosomów** – wybór chromosomów, które biorą udział w tworzeniu nowej populacji.

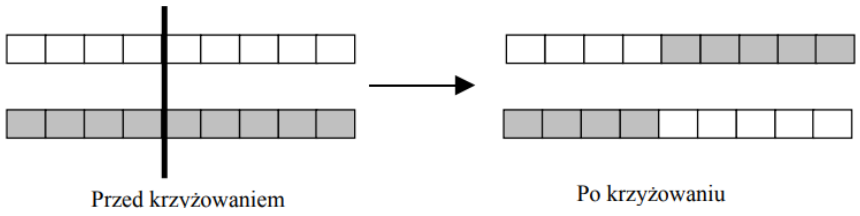
- **Zastosowanie operatorów genetycznych** – na grupie chromosomów, wybranej drogą selekcji działają operatory genetyczne. W klasycznym algorytmie genetycznym są operatory krzyżowania i mutacji. Przy czym, krzyżowanie powinno zachodzić znacznie częściej niż mutacja.
- **Utworzenie nowej populacji** – Chromosomy otrzymane jako rezultat działania operatorów genetycznych na chromosomy tymczasowej populacji wchodzą w skład nowej populacji. W klasycznym algorytmie genetycznym cała poprzednia populacja chromosomów jest zastępowana przez tak samo liczną nowa populacje potomków.
- **Wyprowadzenie „najlepszego" chromosomu** –najlepszym rozwiązaniem jest chromosom o największej wartości funkcji przystosowania.

Nową populację tworzą chromosomy powstałe w wyniku selekcji i działania operatorów genetycznych. Nowa populacja zastępuje w całości starą, i staje się bieżącą w kolejnej generacji (iteracji) algorytmu genetycznego.




Rysunek 1: Schemat algorytmu genetycznego

Charakterystyczny dla klasycznego algorytmu genetycznego jest sposób kodowania. Chromosomy są zakodowanie binarnie - geny mogą przyjmować tylko wartości 0 i 1. Selekcji dokonuje się z wykorzystaniem metody ruletki. Krzyżowanie jest jednopunktowe (rysunek 2)




Rysunek 2: Schemat krzyżowania jednopunktowego

### 3.2 Metody kodowania

**Kodowanie binarne** – przyjęto w klasycznym algorytmie genetycznym. Oznacza to, że allele wszystkich genów w chromosomie są równe 0 lub 1. Reprezentacja rozwiązań w postaci łańcuchów binarnych zdominowała badania nad algorytmami genetycznymi. Kodowanie to ułatwia także analizę teoretyczną i umożliwia wprowadzenie operatorów genetycznych.

**Kodowanie logarytmiczne** – jest kolejnym przykładem modyfikacji kodowania binarnego. Ten sposób kodowania jest stosowany celem zmniejszenia długości chromosomów w algorytmie genetycznym. W kodowaniu logarytmicznym pierwszy bit ciągu kodowego oznacza znak funkcji wykładniczej, drugi bit oznacza znak wykładnika funkcji, a pozostałe bity są reprezentacją wykładnika funkcji.

**Wprowadzenie zapisu zmiennopozycyjnego** można uznać za kolejną modyfikację metod kodowania. Każdy chromosom jest zapisany jako wektor liczb zmiennopozycyjnych. Dokładność takiego podejścia jest zwykle znacznie wyższa niż przy kodowaniu binarnym. Reprezentacja zmiennopozycyjna może objąć całkiem duże lub nawet nieznane dziedziny. W przypadku kodowania zmiennopozycyjnego znacznie łatwiej jest zaprojektować specjalistyczne narzędzia ułatwiające postępowanie w przypadku nie trywialnych ograniczeń.

Jedną z podstawowych zasad przy wyborze kodu jest zasada minimalnego alfabetu. Mówi ona, że należy wybrać najmniejszy alfabet, w którym rozpatrywane zadanie wyraża się w sposób naturalny. Wynika stąd, że można przyjąć struktury inne niż wektory tj. listy, drzewa, macierze, permutacje. Należy pamiętać, że przy wyborze metody kodowania ważne jest, aby zakodowane osobniki były możliwie podobne do rzeczywistych rozwiązań oraz, że do wybranej metody należy zaprojektować odpowiednio specjalistyczne operatory.

### 3.3 Metody selekcji

W algorytmach genetycznych wyróżniamy etap selekcji, który z bieżącej populacji osobników wybiera te o największej wartości funkcji przystosowania do populacji rodzicielskiej.

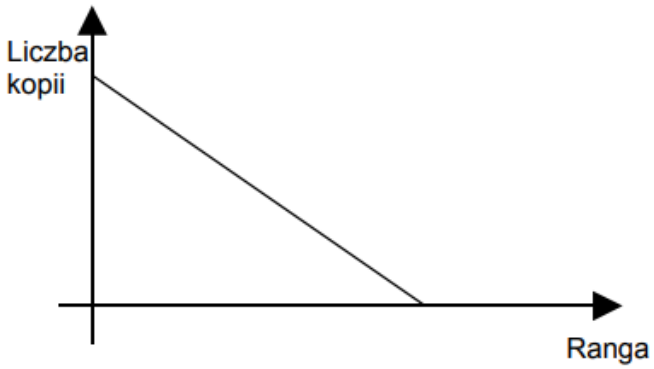
**Selekcję metodą koła ruletki** omówiono przy okazji klasycznego algorytmu genetycznego. Jest to metoda prosta i daje w miarę zadowalające wyniki. Ma jednak wady.

- Jedną z nich jest możliwość stosowania metody ruletki tylko do jednej klasy zadań, tzn. tylko do maksymalizacji (lub tylko do minimalizacji). Takie ograniczenie jest niewątpliwie jej wadą.
- Bardziej istotną wadą metody ruletki jest zbyt wczesne eliminowanie z populacji osobników o bardzo małej wartości funkcji przystosowania. Może to prowadzić do zbyt wczesnej zbieżności algorytmu.

Dlatego stosuje się inne metody selekcji takie jak: selekcja turniejowa, rankingowa i strategia elitarna.

**Selekcja turniejowa** polega na podzieleniu populacji na podgrupy k-elementowe (k to rozmiar turnieju – zwykle 2 lub 3) i wyborze z każdej podgrupy osobnika o najlepszym przystosowaniu. Można to zrobić poprzez wybór losowy lub wybór deterministyczny. Wtedy wyboru dokonuje się z prawdopodobieństwem równym 1. Metoda turniejowa nadaje się zarówno do problemów maksymalizacji jak i minimalizacji.

**W selekcji rankingowej** osobniki populacji są ustawiane kolejno w zależności od funkcji przystosowania – tzn. od najlepiej do najgorzej przystosowanego. Każdemu osobnikowi przyporządkowana jest liczba zwana rangą i oznaczająca jego pozycję na liście. Liczba kopii danego osobnika wprowadzonych do nowej populacji jest ustalana na podstawie wcześniej zdefiniowanej funkcji, zależnej od rangi (Rysunek 3).




Rysunek 3: Przykład funkcji określającej zależność liczby osobników w nowej populacji od wartości rangi

**W strategii elitarnej** nacisk położony jest na zachowanie w kolejnych iteracjach najlepiej przystosowanych osobników. W klasycznym algorytmie genetycznym możliwa jest sytuacja, w której do nowej populacji nie wejdą najlepsze osobniki. Model elitarny ma za zadanie do tego nie dopuścić.

Wymienione powyżej metody selekcji są pewnymi podstawowymi kanonami. Metody selekcji można podzielić na statyczne i dynamiczne. Statyczna selekcja oznacza, że prawdopodobieństwa wyboru są stałe dla wszystkich pokoleń, w selekcji dynamicznej natomiast takiego wymagania nie ma.

**Skalowanie funkcji celu** stosuje się, aby zapobiec przedwczesnej zbieżności algorytmu genetycznego. Przedwczesna zbieżność algorytmu polega na tym, że najlepsze ale jeszcze nie optymalne chromosomy dominują w populacji.

Ponadto w końcowej fazie algorytmu, w przypadku, gdy populacja zachowuje znaczną różnorodność, średnia wartość przystosowania niewiele odbiega od maksymalnej. Skalowanie funkcji przystosowania może wówczas zapobiec takiej sytuacji, że osobniki przeciętne i najlepsze otrzymują prawie taką samą liczbę potomków w następnych generacjach, co jest zjawiskiem niepożądanym. Skalowanie funkcji przystosowania chroni populację przed dominacją nieoptymalnego chromosomu. Skalowanie polega na odpowiednim przekształceniu funkcji przystosowania. Podstawowe metody skalowania to: skalowanie liniowe, skalowanie $\sigma$-obcinające typu sigma i skalowanie potęgą.

- **Skalowanie liniowe** polega na przekształceniu funkcji przystosowania $f$ do postaci $f'$ poprzez przekształcenie liniowe według wzoru:

$$f' = af + b \tag{1}$$

  Współczynniki a i b powinny: zachować niezmienioną wartość średniego przystosowania oraz ustalić maksymalną wartość wyskalowanego przystosowania na poziomie określonej krotności średniego przystosowania. Te dwa warunki łącznie zapewniają, że przeciętny osobnik populacji otrzymuje przeciętnie jednego potomka, a najlepszy tylu – ile wynosi wspomniana krotność. Skalowanie liniowe działa dobrze, za wyjątkiem, gdy funkcja $f'$ przyjmuje wartości ujemne.

- **Skalowanie $\sigma$-obcinające** –typu sigma. Przekształcenie funkcji przystosowania $f$ do postaci $f'$ dokonuje się według następującej zależność:

$$f' = f + ((\bar{f}) - c * \sigma) \tag{2}$$

  gdzie: $f$ jest średnią wartością funkcji przystosowania w populacji, jest małą liczbą naturalną (zwykle 1 do 5), $\sigma$ jest odchyleniem standardowym populacji. Wartości ujemne $f'$ przyjmuje się jako równe zeru.

- **Skalowanie potęgą**, polega na podniesieniu pierwotnej funkcji przystosowania $f$ do pewnej ustalonej potęgi $k$:

$$f' = f^k \tag{3}$$

### 3.4 Operatory genetycznei

Następnym po etapie selekcji, jest etap nazywany też ewolucją. Polega on stosowaniu operatorów genetycznych, tzn. krzyżowania i mutacji, które dokonują rekombinacji genów w chromosomach.

#### Operator krzyżowania

Operacja krzyżowania polega na wymianie fragmentów łańcuchów dwóch chromosomów rodzicielskich. Krzyżowanie jest kluczowym operatorem w algorytmach genetycznych, stanowiącym o ich sile i efektywności. Mutacja odgrywa znacznie mniejsza rolę.

Ideą operatorów krzyżowania jest wymiana kodu genetycznego pomiędzy osobnikami, tak jak to się dzieje w naturze. Stworzono wiele teorii i rodzajów krzyżowań, które stosowane są do różnych rodzajów zadań i są zależne od sposobu kodowania.

- **Krzyżowanie arytmetyczne**, jest to dwuargumentowy operator, zdefiniowany jako liniowa kombinacja dwóch wektorów. Jeżeli do krzyżowania zostały wybrane wektory (chromosomy) $x_1$ i $x_2$, to potomkowie są wyznaczani następująco:

$$x_1' = ax_1 + (1-a)x_2 \tag{4}$$

$$x_2 = ax_2 + (1-a)x_1. \tag{5}$$

  gdzie: $a$ jest wartością przedziału [0,1], co gwarantuje, że potomkami są rozwiązania dopuszczalne.

- **Krzyżowanie heurystyczne**, jest to operator do kierunku poszukiwań używa wartości funkcji celu. Tworzy tylko jednego potomka i może w ogóle nie utworzyć potomka. Nowy osobnik jest określany następująco:

$$ch_3 = r(ch_2 - ch_1) + ch_2 \tag{6}$$

  gdzie: $r$ jest wartością losową z przedziału [0,1] i $F(ch_2) \leqslant F(ch_1)$

**Operator mutacji**

Mutacja polega na wprowadzeniu do istniejących zakodowanych chromosomów, pewnych losowych zmian. Mutacja tworzy nowego osobnika na bazie jednego i tylko jednego rodzica. Jest wiele metod tworzenia nowych osobników u operatora mutacji. Podstawową formę mutacji można zapisać następująco:

$$x' = m(x) \tag{7}$$

gdzie: $x$ jest chromosomem rodzica, $m$ funkcją mutacji, $x'$ chromosomem potomka.

W przypadku chromosomów zakodowanych binarnie (klasyczny algorytm genetyczny) nie ma problemu ze stosowaniem mutacji (po prostu zamieniamy wartość genu na przeciwny).

W przypadku wartości zmiennopozycyjnych nie jest to już takie oczywiste. Poniżej przedstawiono kilka sposobów implementacji operatora mutacji.

**Mutacja równomierna.** Operator ten działa na jednym rodzicu $x$ tworzy potomka $x'$. Operator wybiera losowo pozycję genu, który zostanie mutowany $k \in (1,..,q)$. Mutowany wektor $x = (x_1,..,x_k,...,x_q)$ przekształca się w nowy $x' = (x_1,..,x_k',...,x_q)$, gdzie $x_k'$ przyjmuję wartość losową z dozwolonego przedziału dla tej wartości. Operator ten odgrywa szczególnie ważną rolę we wczesnych fazach procesu ewolucyjnego.

Mutacja brzegowa. Działa na podobnych zasadach, co mutacja równomierna. Natomiast $x_k'$ może z jednakowym prawdopodobieństwem przyjąć wartości brzegów dozwolonego przedziału. Operator ten jest

przydatny dla zadań optymalizacji, gdy poszukiwane rozwiązanie leży blisko brzegu dopuszczalnej przestrzeni poszukiwań.

**Operator inwersji**

Operator inwersji, jest to operator, którego nie można zakwalifikować ani jako operator krzyżowania ani jako operator mutacji. Operuje on na istniejącym chromosomie i wszystkie geny pochodzą z istniejącego chromosomu – nie ma żadnej „dzikiej karty".

Operacja inwersji jest głównym mechanizmem odpowiedzialnym za rekonfigurację kodu. Podczas inwersji chromosom ulega przecięciu w dwóch wybranych punktach, a następnie środkowy jego odcinek ulega odwróceniu i połączeniu z dwoma pozostałymi. Operator inwersji działa przeciwko wstrzymującemu przeszukiwanie brakowi różnorodności.

## 4 Algorytm symulowanego wyżarzania

Algorytm symulowanego wyżarzania jest algorytmem wyszukującym przybliżone rozwiązanie konkretnej instancji jakiegoś problemu NP-trudnego. Jak wie każdy mniej więcej ogarnięty, przestrzeń możliwych rozwiązań problemów NP-trudnych jest z reguły na tyle duża, że odpada jej dokładne przebadanie, tj. sprawdzanie każdego możliwego rozwiązania po kolei. Dla problemu komiwojażera na przykład mamy $\frac{(n-1)!}{2}$ rozwiązań, co już dla 20 miast daje liczbę na tyle dużą, że wykonanie takiego algorytmu zajmie niewiarygodnie dużo czasu. Wykorzystując algorytmy przybliżone jesteśmy w stanie znaleźć rozwiązanie, które może i nie będzie najlepszym (optymalnym), jednak będzie „wystarczająco dobrym", a do tego zostanie ono znalezione w czasie w miarę rozsądnym.

Modyfikując algorytmy metaheurystyczne zezwalając na zmianę rozwiązania z pewnym prawdopodobieństwem, nawet wtedy, gdy jest ono gorsze – bo możliwe, że w sąsiedztwie gorszego będą lepsze niż te, które mamy w sąsiedztwie rozwiązania aktualnie rozpatrywanego. Dzięki temu symulowane wyżarzanie potrafi „unikać" problemów minimum lokalnego.

Algorytm cechuje się następującymi parametrami :

- początkową temperaturą $T$,
- końcową temperaturą $T_min$,
- funkcją prawdopodobieństwa $P$,
- funkcją obniżania temperatury $G$.

Funkcja prawdopodobieństwa określa, z jakim prawdopodobieństwem będziemy zmieniać rozwiązanie na gorsze. W „oryginalnej" odmianie symulowanego wyżarzania funkcja ta była zależna od oceny rozwiązania nowego, starego i aktualnej temperatury. Bierze się to z analogii tego algorytmu do procesu metalurgicznego – póki temperatura jest wysoka, akceptujemy zmiany na rozwiązania początkowo gorsze, ale potencjalnie lepsze. Zakładamy, że w miarę postępowania algorytmu i obniżania temperatury dojdziemy do rozwiązania „w miarę dobrego", tak więc „ryzykowne" zmiany na rozwiązania początkowo gorsze nie będą już potrzebne. Z tego wynika też funkcja obniżania temperatury – wykonywana jest ona po każdej iteracji algorytmu i w oryginalnej odmianie była zależna od aktualnej temperatury i aktualnego numeru iteracji.

## Literatura

1. Stachurski A.: Wprowadzenie do optymalizacji, Ofic. Wyd. PW, Warszawa, 2009.
2. Stadnicki J.: Teoria i praktyka rozwiązywania zadań optymalizacji z przkładami zastosowań technicznych, WNT, Warszawa, 2006.
3. Cegielski A.: Programowanie matematyczne, Ofic. Wyd. Uniw. Zielona Góra, Zielona Góra, 2002.
4. Kusiak J., Danielewska-Tułecka A.: Oprycha P., Optymalizacja. Wybrane metody z przykładami zastosowań, PWN, Warszawa, 2009.
5. Ostanin A.: Optymalizacja liniowa i nieliniowa, Wyd. Pol. Biał., Białystok, 2005.
6. Arabas J.: Wykłady z algorytmów ewolucyjnych, WNT, Warszawa, 2001.
7. Osowski S.: Sieci neuronowe w ujęciu algorytmicznym, WNT, Warszawa 1997
8. Białoszewski T.: Wielokryterialna optymalizacja parametryczna układów z zastosowaniem algorytmów ewolucyjnych, PWNT, Gdańsk, 2007.