



POLITECHNIKA WARSZAWSKA

WYDZIAŁ MATEMATYKI
I NAUK INFORMACYJNYCH



PRACA DYPLOMOWA MAGISTERSKA

**ANALIZA MOŻLIWOŚCI WYKORZYSTANIA
W ALGORYTMIE CMA-ES WIEDZY
O OGRANICZENIACH KOSZTOWYCH**

AUTOR:

INŻ. ROBERT JAKUBOWSKI

PROMOTOR:

DR HAB. INŻ. JAROSŁAW ARABAS
PROF. NZW. PW

WARSZAWA MAJ 2016

.....

podpis promotora

.....

podpis autora

Spis treści

1	Streszczenie	4
2	Wstęp	5
2.1	Cel pracy	5
3	Techniki uwzględniania ograniczeń	6
3.1	Transformacje rozwiązań	6
3.1.1	Metoda klasyczna	6
3.1.2	Rzutowanie	7
3.1.3	Reinicjacja	7
3.1.4	Odbicie	7
3.1.5	Próbkowanie	7
3.1.6	Zawijanie	8
3.2	Błądzenie przypadkowe	8
3.3	Metoda przeprowadzania testów	8
3.3.1	Wartość oczekiwana generowanego punktu	9
3.3.2	Rozkład prawdopodobieństwa generowanych punktów	10
3.4	Wyniki testów	10
3.5	Test przesunięć	21
3.6	Wnioski	21
4	Metodyka testowania metod optymalizacji globalnej	22
4.1	Funkcje benchmarkowe	22
4.2	Techniki porównywania wyników	22
5	Wpływ technik na efektywność CMA-ES	23
5.1	Algorytm CMA-ES	23
5.1.1	Opis algorytmu	23
5.2	CEC 2013	25
6	Podsumowanie	26
6.1	Wyniki	26
6.2	Możliwości rozwoju	26

1. Streszczenie

2. Wstęp

2.1. Cel pracy

3. Techniki uwzględniania ograniczeń

Niektóre problemy optymalizacyjne posiadają ograniczenia. Szukając rozwiązania należy zapewnić, że będzie ono dopuszczalne. Bazując na —przypis— techniki uwzględniania ograniczeń można podzielić w następujący sposób:

- definicja przestrzeni przeszukiwań - zapewnienie, że podczas krzyżowań, mutacji i innych zmian punktów, żaden z punktów nie wypadnie poza przestrzeń przeszukiwań,
- modyfikacja funkcji celu - zmienienie funkcji celu tak, aby funkcja celu dla punktów spoza ograniczeń zwracały gorsze wyniki,
- transformacja rozwiązań - punkty, które są poza ograniczeniami zostają zamieniane na punkty, które znajdują się w ograniczeniach.

W tej pracy skupiono się na transformacji rozwiązań.

3.1. Transformacje rozwiązań

Istnieje wiele technik transformacji rozwiązań spoza ograniczeń na dopuszczalne. W kolejnych podrozdziałach znajdują się metody transformacji rozwiązań, które zostały zbierane. Każda z technik jest opisana słownie oraz pseudokodem. Opis słowny zawiera wyjaśnienie, co się dzieje z punktem, który znalazł się poza ograniczeniem. W pseudokodzie zastosowano następujące oznaczenia:

- x - punkt, który poddajemy naprawie
- x' - rodzic punktu x , czyli z punktu x' z zadany rozkładem został wygenerowany punkt x
- $x(i)$ - wartość i -tej współrzędnej punktu x
- lb - ograniczenie dolne
- ub - ograniczenie górne

3.1.1. Metoda klasyczna

Nowy punkt zostaje odrzucony i wraca do poprzedniej pozycji.

$$\mathbf{x} = \mathbf{x}'$$

3.1.2. Rzutowanie

Punkt jest transformowany do najbliższego punktu, który spełnia ograniczenie. Oznacza to, że dla każdej współrzędnej sprawdzany jest warunek zawierania się w ograniczeniach. Dla współrzędnych, dla których nie jest on spełniony, wartość jest zamieniana na wartość ograniczenia (dolnego lub górnego), które jest najbliższe.

```
dla każdej współrzędnej i
    jeżeli  $lb(i) > x(i)$ 
         $x(i) = lb(i)$ 
    jeżeli  $ub(i) < x(i)$ 
         $x(i) = ub(i)$ 
```

3.1.3. Reinicjacja

Punkt jest przenoszony do pozycji początkowej. W tej pracy był to jednocześnie środek układu współrzędnych oraz środek symetrii ograniczeń.

```
 $x = x_0$ 
```

3.1.4. Odbicie

Dla każdej współrzędnej sprawdzane są warunki na ograniczenie. W przypadku współrzędnych, na których punkt jest poza ograniczeniem, wartość punktu tej współrzędnej jest symetrycznie odbita względem ograniczenia, którego warunek został złamany.

```
dla każdej współrzędnej i
    jeżeli  $lb(i) > x(i)$ 
         $x(i) = x(i) + 2 * (lb(i) - x(i))$ 
    jeżeli  $ub(i) < x(i)$ 
         $x(i) = x(i) - 2 * (x(i) - ub(i))$ 
```

3.1.5. Próbkowanie

Punkt jest powtórnie generowany do momentu, aż spełni ograniczenie kostkowe.

```
dopóki  $!w\_ograniczeniach(x)$ 
     $x = losuj(x')$ 
```

3.1.6. Zawijanie

Dla każdej współrzędnej sprawdzane są warunki na ograniczenie. W przypadku współrzędnych, na których punkt jest poza ograniczeniem, różnica, pomiędzy ograniczeniem a wartością współrzędnej punktu, jest zapamiętywana. Tę różnicę odkładamy na przeciwległym ograniczeniu po stronie, która jest wewnątrz ograniczenia. W tym miejscu znajduje się nowa wartość współrzędnej punktu. W intuicyjny sposób można to wyjaśnić tak, że dla punktów nie ma ograniczeń, a przestrzeń przeszukiwań po każdym wymiarze jest jakby "zawinięta".

```
dla każdej współrzędnej i
    jeżeli lb(i) > x(i)
        x(i) = ub(i) - (lb(i) - x(i))
    jeżeli ub(i) < x(i)
        x(i) = lb(i) + (x(i) - ub(i))
```

3.2. Błądzenie przypadkowe

Można się spodziewać, że algorytm CMA-ES dla funkcji stałej będzie zachowywał się analogicznie do błądzenia przypadkowego. Takie założenie skłoniło autora, żeby zbadać błądzenie przypadkowe z ograniczeniami. Błądzenie przypadkowe jest algorytmem dużo prostszym, niż CMA-ES, więc umożliwia szybsze testowanie i wyciąganie wniosków.

Niech X_1, X_2, \dots będą niezależnymi n -wymiarowymi zmiennymi losowymi o wartości oczekiwanej równej $[0]^n$. Błądzeniem przypadkowym nazywamy sekwencję punktów S_1, S_2, \dots , takich że:

$$S_1 = X_1, S_i = S_{i-1} + X_i \quad (1)$$

Zmienne losowe mogą być realizowane w różny sposób. Mogą to być wektory o rozkładzie normalnym, jednostajnym lub innym.

3.3. Metoda przeprowadzania testów

Celem testów jest zaobserwowanie, jaki wpływ na symulację ma metoda uwzględniania ograniczeń. Wybrano 2 cechy, które miały być przetestowane:

1. Wartość oczekiwana generowanego punktu (razem z poprawą).
2. Rozkład prawdopodobieństwa generowanych punktów.

Sposoby testowania obu cech są opisane w kolejnych podrozdziałach.

3.3.1. Wartość oczekiwana generowanego punktu

Błądzenie przypadkowe jest dobrze określone. W tym algorytmie wartością oczekiwaną losowanej liczby X_i jest 0. Implikuje to fakt, że wartością oczekiwaną S_i jest S_{i-1} . Dodanie ograniczeń kostkowych powoduje, że nadal losujemy liczbę X_i , której wartość oczekiwana jest równa 0. Dodajemy jednak metody naprawy. W związku z tym nie zawsze wartością oczekiwaną S_i jest S_{i-1} .

Dla każdego punktu p przestrzeni przeszukiwań można przypisać wektor \vec{d} . Początek \vec{d} znajduje się w p . Załóżmy teraz, że p jest punktem S_i pewnego błędzenia przypadkowego z ograniczeniami. Koniec wektora \vec{d} znajduje się w wartości oczekiwanej punktu S_{i+1} .

Wykres, który będzie zawierał wektory \vec{d} , pozwoli na określenie charakterystyki przemieszczania się punktów w poszczególnych częściach przestrzeni przeszukiwań.

Implementacja

Do wizualizacji wybrano symulację, w której punkty posiadają 2 wymiary. Pozwala to na przejrzystą wizualizację wektorów dla każdego z punktów. Pokazuje też zachowanie się wektorów w rogach - punktach, które są blisko ograniczeń.

Dla obu wymiarów wybrano kilkanaście, równoodległych punktów. W skład tych punktów wchodziły też punkty znajdujące się na ograniczeniach. Następnie stworzono 2 pętle po tych punktach (jedna zagnieżdżona w drugiej). W ten sposób uzyskano regularnie rozłożone punkty w dwóch wymiarach.

Dla każdego z tych punktów przeprowadzono następujące obliczenia. 1000 razy zasyulowano krok algorytmu błędzenia przypadkowego (z rozkładem normalnym) z odpowiednią naprawą. Otrzymano tysiąc punktów, z których następnie obliczono średnią

arytmetyczną. Tak uzyskany punkt potraktowano jako wartość oczekiwaną i wyrysowano wektor \vec{d} .

3.3.2. Rozkład prawdopodobieństwa generowanych punktów

Punkty w błędzeniu przypadkowym poruszają się chaotycznie. Ta losowość jest częściowo uporządkowana na ograniczeniach. W zależności od metody naprawy, punkty zachowują się inaczej, a rozkład prawdopodobieństwa jest zniekształcony. Autor postanowił przyjrzeć się jak wygląda rozkład prawdopodobieństwa wystąpienia punktu z perspektywy całej symulacji.

Dla każdej metody naprawy opisanej w 3.1 jednokrotnie uruchomiono algorytm błędzenia przypadkowego jednego punktu. Symulowano oddzielnie również błędzenie z rozkładem normalnym oraz jenostajnym na przedziale $[-0.5; 0.5]$ (przedział co najmniej kilkukrotnie krótszy od ograniczeń kostkowych). Po każdej iteracji nowo wygenerowany punkt zapisywano w tablicy. Z tak przygotowanej tablicy generowano histogram.

Implementacja

Dla każdej z metod naprawy przygotowano oddzielny skrypt. Pseudokod skryptu zamieszczono poniżej.

```
x - błądzący punkt
punkty - tablica wszystkich położzeń punktu x
iteracje - liczba iteracji podana jako parametr
i = 0
dopóki i < iteracje
    d = wektor wylosowany z zadany rozkładem
    x = x + d
    jeżeli x jest poza ograniczeniem
        popraw x
    dodaj x do tablicy punkty
    i = i + 1
```

3.4. Wyniki testów

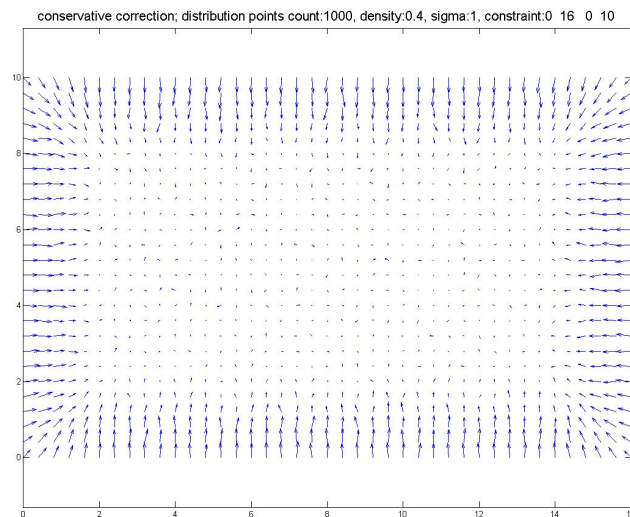
Wykresy zamieszczone w tym podrozdziale są histogramami wystąpień punktu. Symulacje skryptów były uruchamiane z liczbą iteracji wynoszącą 2-100 milionów. Szerokość

przedziałów na wykresach jest różna i była dobierana tak, aby jak najlepiej przedstawić interesujące fakty.

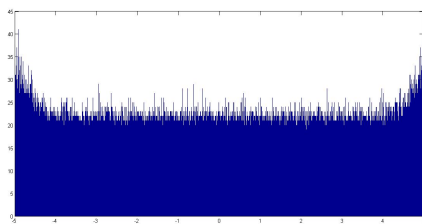
Wszystkie wykresy, które pokazują wyniki błędzenia przypadkowego w dwóch wymiarach zostały przetworzone w następujący sposób: otrzymane wyniki symulacji powielono poprzez symetryczne odbicie ich względem osi X, osi Y oraz początku układu współrzędnych. Wykonano ten zabieg, aby zwiększyć dokładność wyników. Nie wpływają one na badania, ponieważ punkty 0 są środkami ograniczeń obu osi, a z perspektywy algorytmu nie ma znaczenia, w której ćwiartce znajduje się aktualnie punkt.

Metoda klasyczna

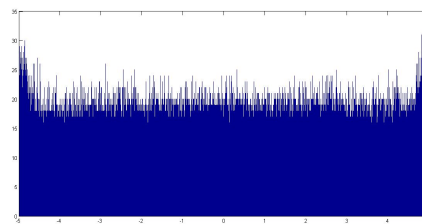
Histogramy pokazują, że punkt z większym prawdopodobieństwem występuje przy ograniczeniach. Takie zachowanie można uznać za intuicyjne, ponieważ potomkowie punktów blisko granicy mogą "wypaść" poza ograniczenie. Po naprawie dziecko będzie tym samym punktem. Symulacja dwuwymiarowa nie pokazuje tego tak dokładnie, natomiast również można dostrzec ten efekt.



Rysunek 1: Wartość oczekiwana punktu

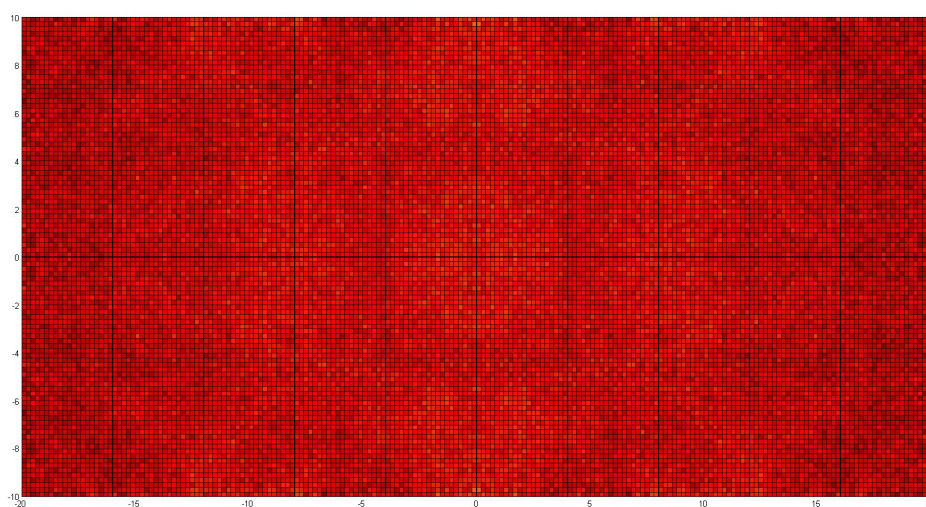


(a) Generowanie z rozkładem normalnym



(b) Generowanie z rozkładem jednostajnym

Rysunek 2: Rozkład prawdopodobieństwa punktów; 1 wymiar

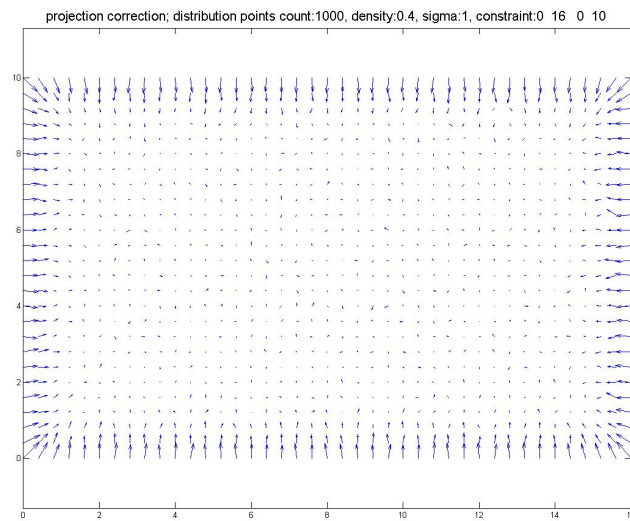


Rysunek 3: Rozkład prawdopodobieństwa punktów; 2 wymiary; generowanie z rozkładem normalnym

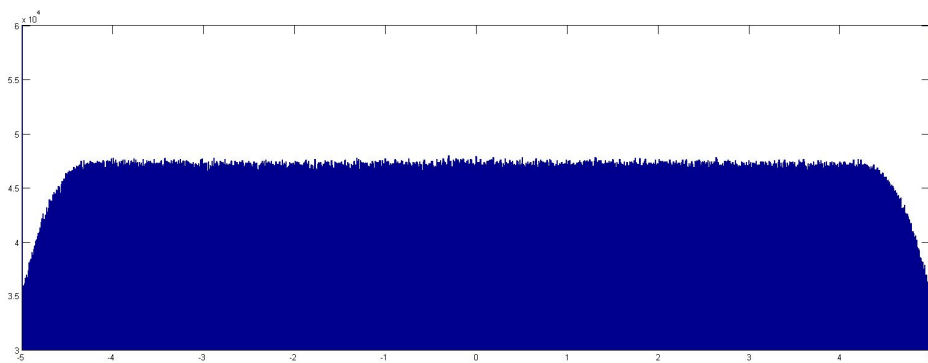
Rzutowanie

Zgodnie z przewidywaniami największe prawdopodobieństwo wystąpienia punktu jest na ograniczeniu, co bardzo dobrze obrazuje rysunek 7. Na rysunkach 5 i 6 zakres osi Y został zmniejszony do odpowiednio $[30000; 60000]$ oraz $[70000; 140000]$, aby uwypuklić interesujące efekty. W związku z tym, zostały "ucięte" wartości przedziałów brzegowych. Warto zwrócić uwagę na niespodziewane zjawisko - wartości przedziałów blisko ograniczeń są mniejsze, niż pozostałe. Oznacza to, że punkty w tych przedziałach występują z mniejszym prawdopodobieństwem, niż pozostałe. Wydawać by się mogło, że powinno być inaczej, ponieważ podczas symulacji punkty często są rzutowane na ograniczenie. Dodatkowo, w rozkładzie jednostajnym widać, że istnieją przedziały,

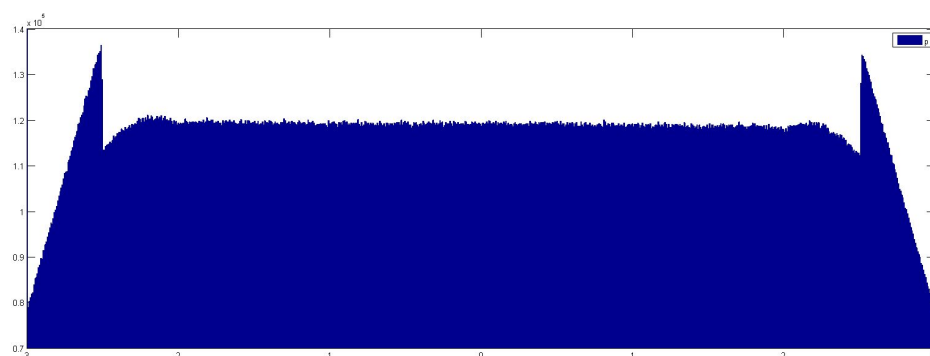
w których punkty występują z większym prawdopodobieństwem. Autorowi nie udało się formalnie uzasadnić tego zjawiska. Intuicja prowadzi do hipotezy, że punkty, które znalazły się poza ograniczeniem, bez naprawy po kilku iteracjach wypełniałyby "dołek". Naprawa natomiast sprawia, że cała dalsza symulacja zostaje niejako przesunięta. W ten sposób powstaje "górką" w rozkładzie jednostajnym. Z charakterystyki rozkładu normalnego może brać się fakt, iż owa górką w nim nie występuje.



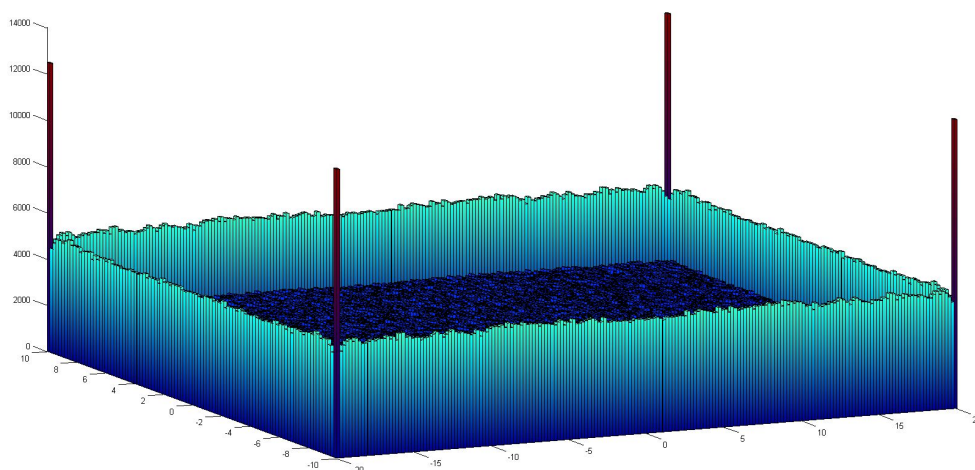
Rysunek 4: Wartość oczekiwana punktu



Rysunek 5: Rozkład prawdopodobieństwa punktów; 1 wymiar; generowanie z rozkładem normalnym



Rysunek 6: Rozkład prawdopodobieństwa punktów; 1 wymiar; generowanie z rozkładem jednostajnym



Rysunek 7: Rozkład prawdopodobieństwa punktów; 2 wymiary; generowanie z rozkładem normalnym

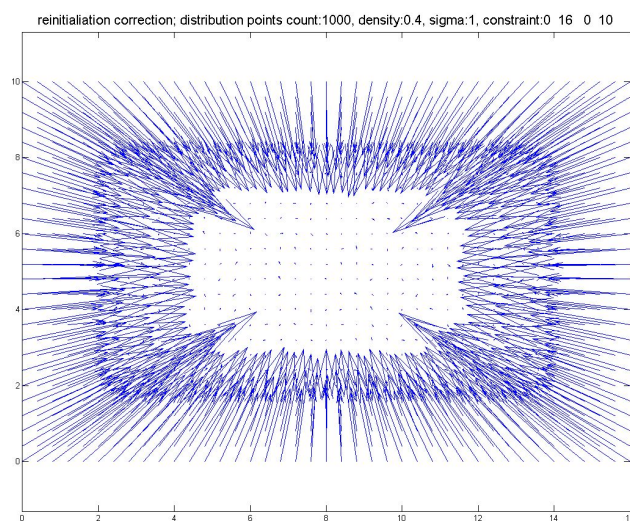
Reinicjacja

Na pierwszy rzut oka wykresy nie ukazują nic nadzwyczajnego, ponieważ łatwo zauważyć pik związany z reinicjacją oraz wartości histogramu malejące wraz ze zbliżaniem się do ograniczeń. Warto jednak zwrócić uwagę na 2 fakty.

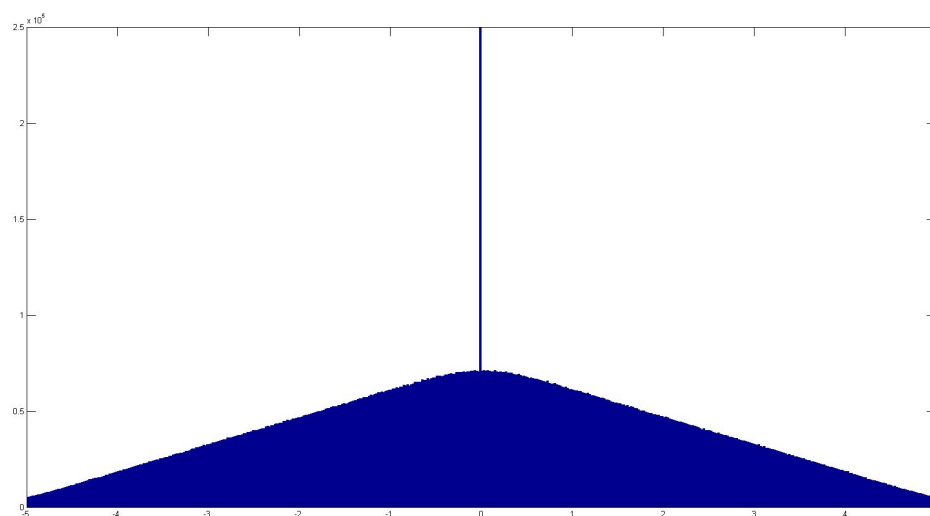
Pierwsze spostrzeżenie, to kształt histogramu. Zarówno dla rozkładu jednostajnego, jak i normalnego w jednym wymiarze jest on stożkowy. Łuk można zauważyć tylko blisko punktu środkowego, w pozostałej części spadek jest liniowy. Brakuje charakterystycznego, gaussowskiego przegięcia. Sytuacja jest ciekawsza, gdy występuje więcej

wymiarów. Widać wówczas przegięcie. Dokładniejsze badania pokazały, że przegięcie nie występuje tylko na jednym wymiarze - tym, który jest relatywnie najkrótszy. Celowo jest użyte słowo relatywnie, ponieważ z perspektywy błędzenia przypadkowego i rozkładu normalnego trzeba brać pod uwagę parametr σ . Wymiary o małym σ będą relatywnie dłuższe od tych z dużym σ , ponieważ błędzenie będzie wykonywało mniejsze kroki.

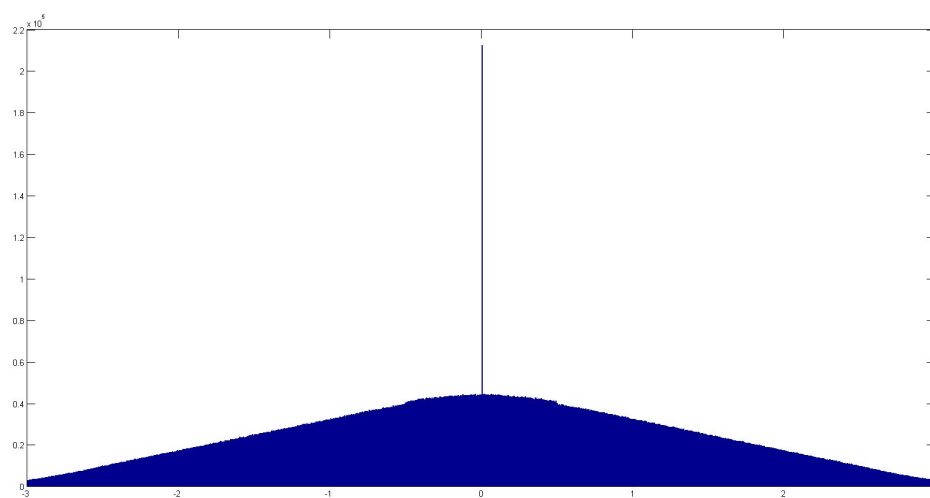
Na rysunku ?? można dostrzec też dwa uskoki, które związane są z pikiem w punkcie 0 oraz charakterystyką rozkładu.



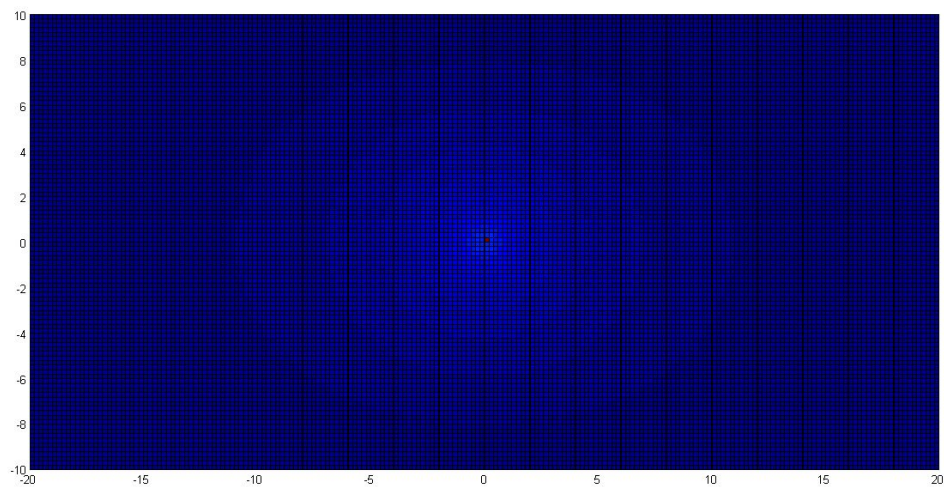
Rysunek 8: Wartość oczekiwana punktu



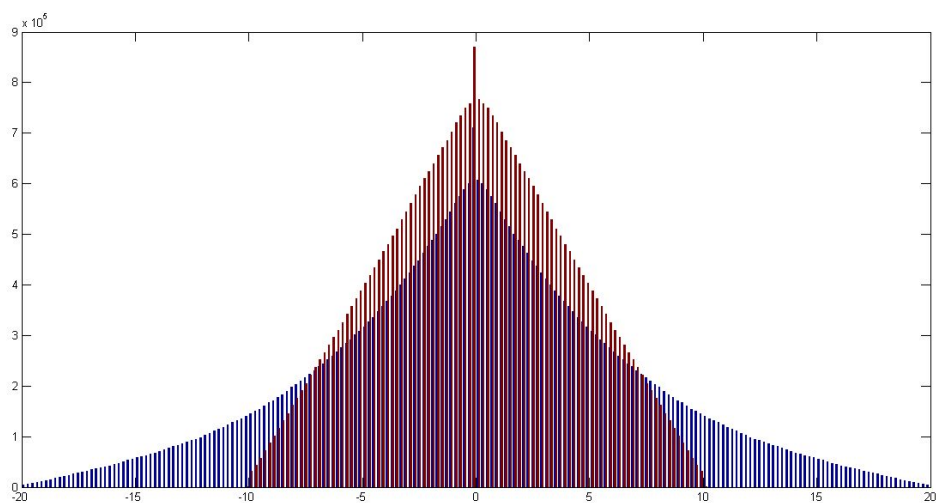
Rysunek 9: Rozkład prawdopodobieństwa punktów; 1 wymiar; generowanie z rozkładem normalnym



Rysunek 10: Rozkład prawdopodobieństwa punktów; 1 wymiar; generowanie z rozkładem jednostajnym



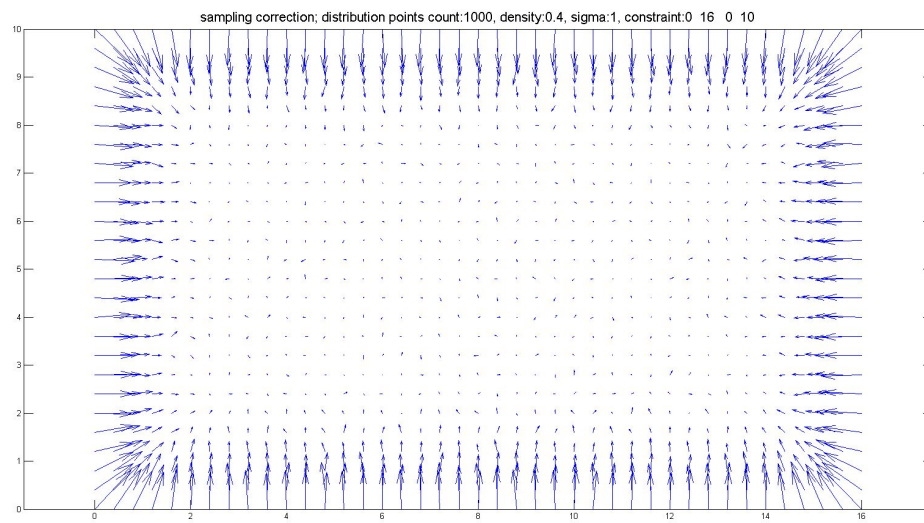
Rysunek 11: Rozkład prawdopodobieństwa punktów; 2 wymiary; generowanie z rozkładem normalnym



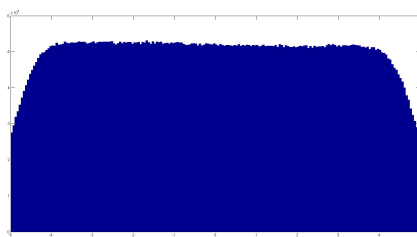
Rysunek 12: Rozkład prawdopodobieństwa punktów; 2 wymiary; generowanie z rozkładem normalnym; oddzielne histogramy dla obu wymiarów

Próbkowanie

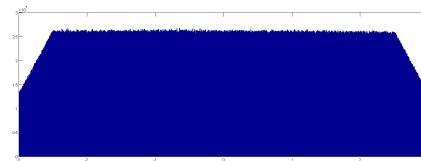
Ten rozkład charakteryzuje się spadkiem wartości prawdopodobieństwa wraz ze zbliżaniem się do ograniczenia. Punkty, które wypadłyby poza ograniczenia oraz ich potomkowie są przesuwane w kierunku środka przedziału.



Rysunek 13: Wartość oczekiwana punktu

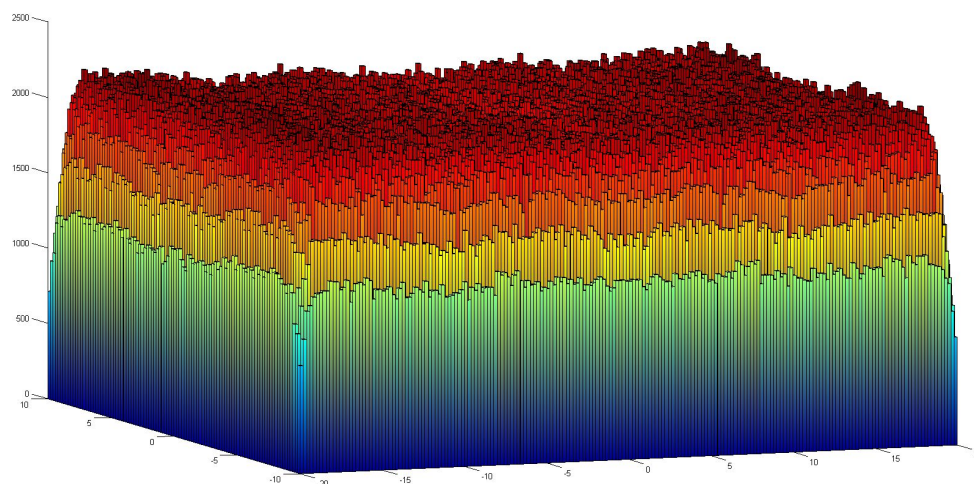


(a) Generowanie z rozkładem normalnym



(b) Generowanie z rozkładem jednostajnym

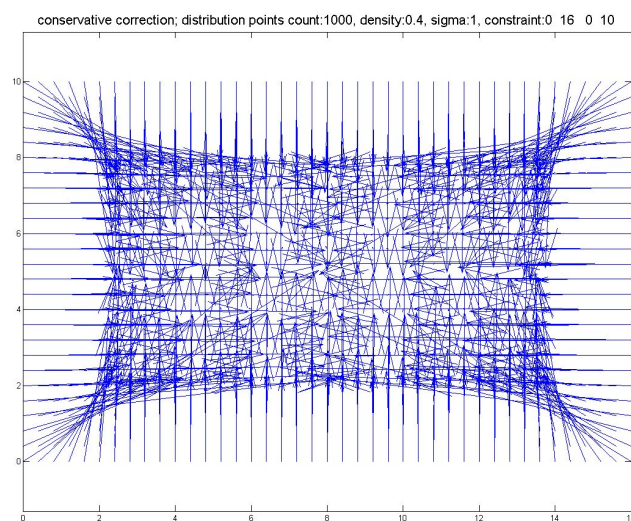
Rysunek 14: Rozkład prawdopodobieństwa punktów; 1 wymiar



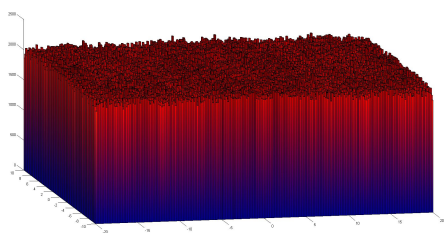
Rysunek 15: Rozkład prawdopodobieństwa punktów; 2 wymiary; generowanie z rozkładem normalnym

Zawijanie

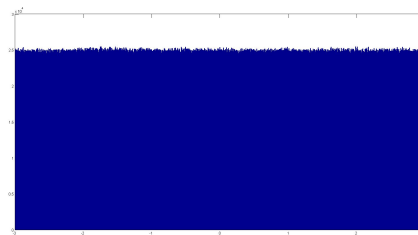
Wyobraźmy sobie, że naszą symulację przeprowadzamy bez ograniczeń, następnie wykres z wynikami tniami w równych odstępach wzdłuż każdej osi. Na koniec tak pocięte części łączymy w jeden wykres. Takie zachowanie symuluje zawijanie. W związku z tym nie dziwi fakt, że poniższe histogramy przedstawiają rozkład jednostajny z szumem.



Rysunek 16: Wartość oczekiwana punktu



(a) 1 wymiar; generowanie z rozkładem normalnym

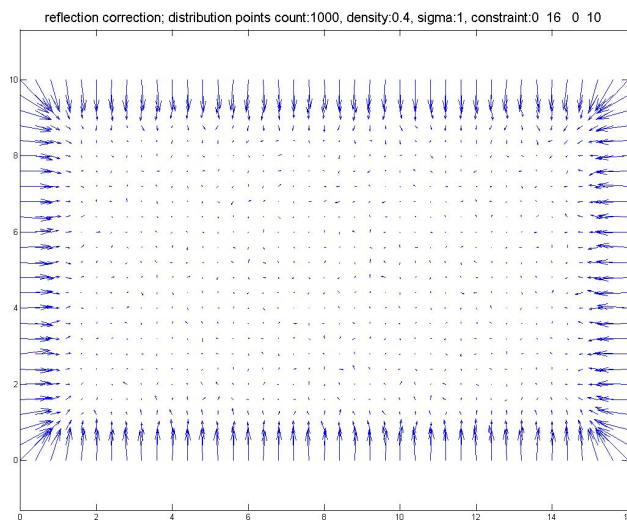


(b) 2 wymiary; generowanie z rozkładem jednostajnym

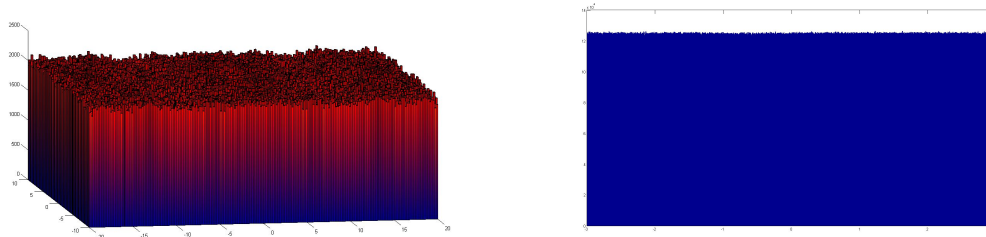
Rysunek 17: Rozkład prawdopodobieństwa punktów;

Odbicie

Podobnie jak zawijanie, odbicie zwraca histogram rozkładu normalnego z szumem. W tej sytuacji nieco trudniej o analogię, lecz po chwili zastanowienia nie dziwi kształt poniższych wykresów.



Rysunek 18: Wartość oczekiwana punktu



(a) 1 wymiar; generowanie z rozkładem normalnym

(b) 2 wymiary; generowanie z rozkładem jednostajnym

Rysunek 19: Rozkład prawdopodobieństwa punktów;

3.5. Test przesunięć

Dodatkowo został wykonany test, w którym badano wartość oczekiwaną przesunięcia punktu w kolejnej iteracji (razem z naprawą). Dla każdej z metod wyglądał on podobnie:

Jedynie dla zawijania oraz reinicjacji długości wektorów były znacząco większe. W pozostałych zmieniał się kierunek oraz długość wektora. Były to jednak zmiany wynikające pośrednio z pokazanych wcześniej histogramów.

3.6. Wnioski

Techniki uwzględniania ograniczeń mają swoje charakterystyczne cechy. W większości metod zachowanie pojedynczej cząsteczki przekłada się na globalne zachowanie całego algorytmu. Testy pokazały cechy, które nie były oczywiste podczas prostej analizy algorytmów.

Z perspektywy algorytmu CMA-ES warto zwrócić uwagę przede wszystkim na metody zawijania i reinicjacji. W tych metodach wartości oczekiwane punktów znajdują się daleko od rodzica. Oznacza to, że w algorytmie CMA-ES macierz kowariancji może znacząco się zniekształcać.

4. Metodyka testowania metod optymalizacji globalnej

Funkcje posiadają charakterystyczne cechy, m.in. monotoniczność, liczba ekstremów, ciągłość. Metody optymalizacji globalnej również posiadają swoje charakterystyczne własności. W zależności od przypadku, wymagania stawiane algorytmowi mogą być różne: szybkości działania, przeszukanie całej przestrzeni, czy też dokładne określenie ekstremum. Połączenie tego wszystkiego powoduje, że trudno jednoznacznie określić, która metoda optymalizacji jest najlepsza. Warto jednak posiadać wiedzę w których przypadkach metoda przynosi lepsze rezultaty.

Zapewne zgodziliby się z tym stwierdzeniem twórcy konkursów takich jak CEC, czy BBOB. Tworzą oni szereg funkcji benchmarkowych, które mają na celu empirycznie pokazać silne i słabe strony algorytmów w konkretnych sytuacjach.

4.1. Funkcje benchmarkowe

Funkcje benchmarkowe, to złożone funkcje. Skomplikowanie polega między innymi na wielu ekstremach funkcji lub specyficznym umieszczeniu minimum globalnego. Dzięki nim można porównać algorytmy.

4.2. Techniki porównywania wyników

5. Wpływ technik na efektywność CMA-ES

5.1. Algorytm CMA-ES

Klasyczne algorytmy ewolucyjne nie dostosowują się do charakterystyki optymalizowanej funkcji. W większości z nich, podczas generowania punktów, rozkład prawdopodobieństwa jest stały. Z tego faktu wynika problem ustalenia parametrów przeszukiwania. Na przeciw temu wychodzi algorytm CMA-ES, który w swej idei ma dopasowywać się do badanej funkcji.

Rozwinięcie akronimu CME-ES podpowiada, w jaki sposób jest to realizowane: Covariance Matrix Adaptation - Evolution Strategy (adaptacja macierzy kowariancji - strategia ewolucyjna). Jest to alorytm oparty na tehchnikach ewolucyjnych, natomiast punkty losowane są na podstawie macierzy kowariancji, która jest w każdej iteracji dostosowywana do aktualnej sytuacji przeszukiwań.

5.1.1. Opis algorytmu

Zgodnie z tym, co zostało ustalone na początku tego rozdziału, algorytm CMA-ES jest podobny do klasycznych algorytmów ewolucyjnych. Inny jest sposób generowania punktów. Można to zauważyć w poniższym, ogólnym pseudokodzie opisującym algorytm CMA-ES:

```
inicjalizuj zmienne
dopóki nie są spełnione warunki stopu:
    generuj punkty
    oblicz wartości punktów
    przeprowadź selekcję punktów
    oblicz wartość oczekiwaną rozkładu
    oblicz długość kroku
    oblicz macierz kowariancji
```

Ten pseudokod zaiwera kilka pojęć, które nie zostały jeszcze omówione. Ich opis znajduje się w kolejnych podrozdziałach. Nie jest on wyczerpujący. Aby dokładnie zgłębić algorytm CMA-ES oraz zapoznać się z formalnym opisem poszczególnych fragmentów warto przeczytać —przypis—

Macierz kowariancji

Pierwszą zmienną, która będzie opisana jest macierz kowariancji. To ona jest trzonem, a wszystkie kolejne z niej wypływają. Tak jak napisano we wstępie do tego rozdziału, algorytm CMA-ES posiada inny sposób generowania punktów, niż klasyczne algorytmy ewolucyjne. Wszystkie punkty są generowane poprzez losowanie na podstawie macierzy kowariancji. Oznacza to, że nie istnieje bezpośredni związek pomiędzy punktami - w klasycznych algorytmach ewolucyjnych zazwyczaj można było mówić o rodzicach nowo wygenerowanego punktu.

Rola dotychczas wygenerowanych punktów jest jednak istotna w algorytmie CMA-ES. Mają one wpływ na adaptację macierzy kowariancji. Odbyna się to w następujący sposób: z populacji punktów wybiera się μ punktów (zazwyczaj połowę), która zwraca lepsze wartości funkcji celu. Następnie przypisuje się im wagi - w zależności od tego, jak dobry jest punkt. Na podstawie tych danych wyznaczana jest empiryczna macierz kowariancji.

Wartość oczekiwana rozkładu

Punkty generowane są na podstawie rozkładu określonego macierzą kowariancji wokół wartości oczekiwanej. Sama nie mówi jednak nic o wartości oczekiwanej. Wymaga to oddzielnego wyznaczenia tej wartości.

Podobnie, jak w macierzy kowariancji wybiera się μ punktów. Dla tych punktów wybiera się średnią ważoną. Ta średnia jest nową wartością oczekiwaną.

Długość kroku

Generowanie punktów z macierzy kowariancji pozwala na skalowanie tego rozkładu przez dodanie jednego parametru. Ten parametr jest nazywany długością kroku. W algorytmie CMA-ES jest na podstawie kierunku przemieszczania się wartości oczekiwanej rozkładu. Długość kroku zwiększa się, gdy wartość oczekiwana przemieszcza się w tym samym kierunku. Skrac natomiast, gdy kierunek jest zmienny.

Generowanie punktów

Punkty są generowane poprzez wylosowanie wektora liczb na podstawie macierzy kowariancji. Wylosowany wektor jest skalowany przez długość kroku, a na koniec jest przesuwany o wektor równy wartości oczekiwanej rozkładu.

Selekcja punktów

Selekcja punktów, to wybór μ punktów spośród całej populacji. Wybierane są te punkty, które mają lepszą wartość funkcji. Punktom przypisywane są również wagi wykorzystywane do wyznaczania wartości oczekiwanej.

5.2. CEC 2013

- wybrane kilka funkcji
- kilkukrotne powtórzenie testów
- porównanie dokładności wyników i liczby iteracji

Zgodnie z założeniami poczynionymi w rozdziale 3.2 testy algorytmu CMA-ES powinny przynieść rezultaty zbliżone do testów błędzenia przypadkowego. Do przeprowadzania testów została użyta biblioteka przygotowana przez Nikolausa Hansena, współautora algorytmu CMA-ES. Podobnie, jak w przypadku błędzenia przypadkowego, wykorzystano implementację w języku MATLAB —przypis—.

6. Podsumowanie

6.1. Wyniki

6.2. Możliwości rozwoju

Literatura

Warszawa, dnia 23 maja 2016

Oświadczenie

Oświadczam, że pracę magisterską pod tytułem „Analiza możliwości wykorzystania w algorytmie CMA-ES wiedzy o ograniczeniach kosztowych”, której promotorem jest dr hab. inż. Jarosław Arabas prof. nzw. PW, wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....