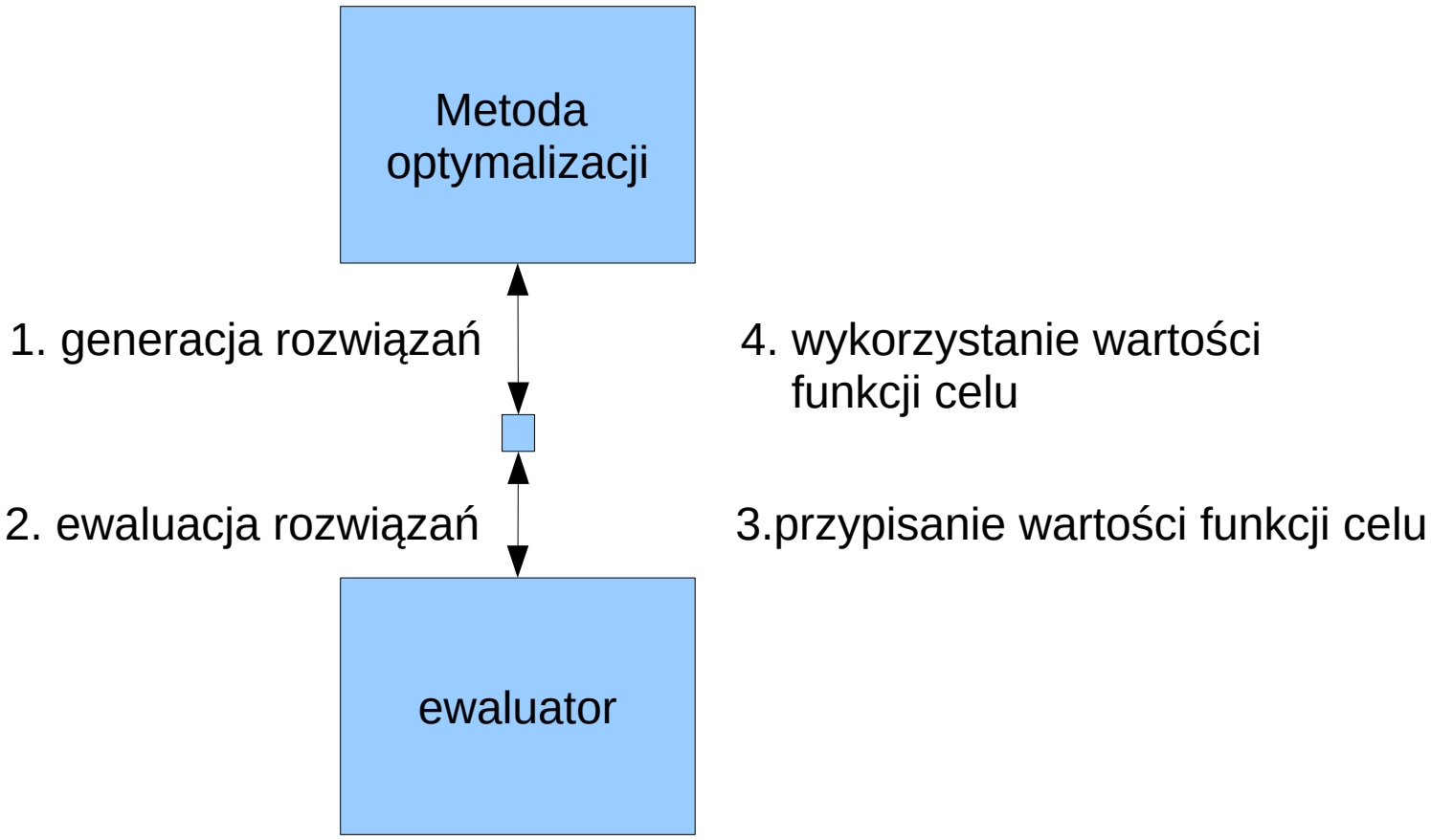
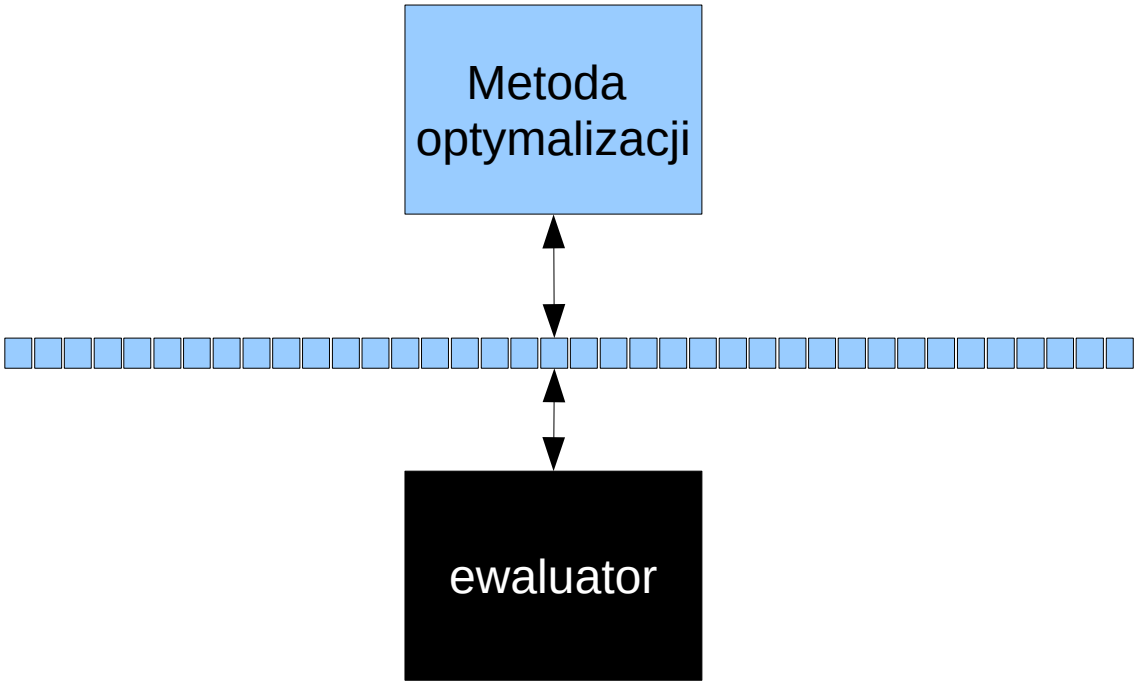
Testowanie metod optymalizacji i hipotez statystycznych

Opracowanie: Łukasz Lepak, 277324

Po wyborze i wykorzystaniu pewnej metody optymalizacji nadchodzi czas na ocenę wytworzonych przez nią punktów. Potrzebny do tego jest ewaluator, który jest w stanie wygenerować wartość funkcji celu dla danego punktu w przestrzeni. Ewaluator taki jest wyróżniony jako osobny blok dokonujący ocen na podstawie punktów dostępnych w logu generowanym przez metodę optymalizacyjną. Możemy zatem powiedzieć, że metoda optymalizacyjna pełni rolę generatora punktów, które następnie są oceniane przez ewaluator. Jednak istnieją metody, które wykorzystują wartość funkcji celu danego punktu do podejmowania dalszych decyzji, np. symulowane wyżarzanie czy stochastyczny algorytm wspinaczkowy. W tym celu ewaluator przypisuje do zadanego punktu w logu wartość funkcji celu, która następnie może zostać wykorzystana przez metodę optymalizacyjną celem podjęcia pewnych decyzji. Powyższy opis prezentuje sposób komunikacji między metodą optymalizacyjną, a ewaluatorem. Schemat tej komunikacji dla pewnego punktu z logu jest przedstawiony na rysunku poniżej.

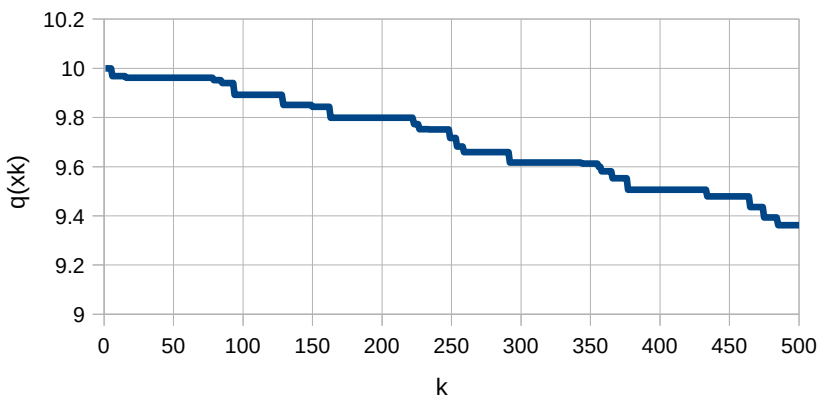


Schemat ten może być wykorzystany do wprowadzenia koncepcji optymalizacji typu „czarna skrzynka” (ang. black-box optimization). Zakłada ona, że metoda optymalizacji nie wie nic o optymalizowanej funkcji celu, jedyne, czego może się o niej dowiedzieć, to wartość tej funkcji dla wygenerowanej przez nią punktów. Czyni to poprzez odpytywanie ewaluatora, który jest traktowany jak czarna skrzynka podająca – w jakiś, nieznany metodzie sposób – wartości funkcji celu.



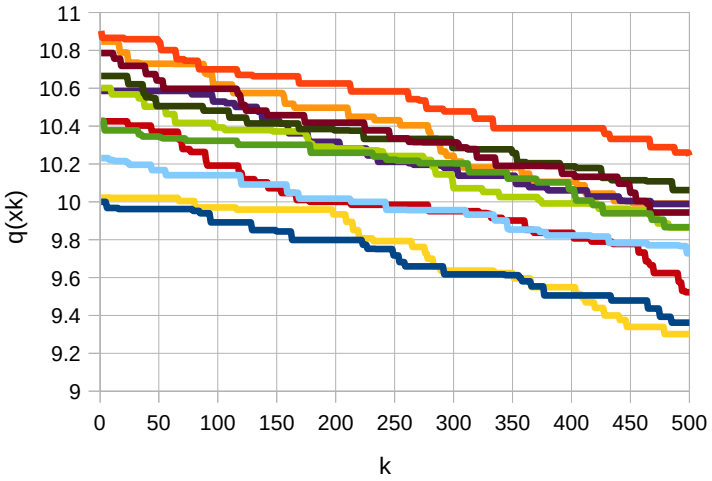
Jednym ze sposobów na przetestowanie metody optymalizacji są tzw. zadania benchmarkowe, wśród których można wyróżnić m.in. CEC (2005…2017) i BBOB (2009). Celem takich zadań jest minimalizacja wartości bardzo skomplikowanej funkcji posiadającej najczęściej wiele minimów lokalnych, której forma analityczna nie jest oczywiście znana. Na takich benchmarkach są najczęściej testowane nowe algorytmy optymalizacyjne, które chcą pokazać, że są w stanie konkurować z obecnie znanymi rozwiązaniami.

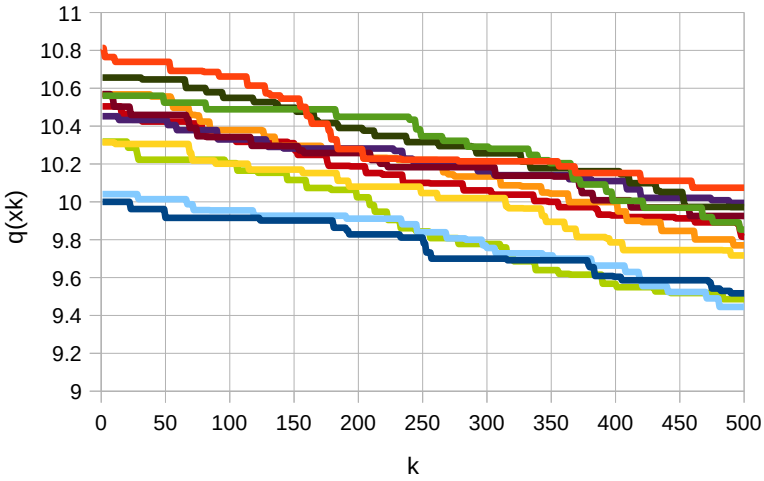
Wyniki osiągnięte z wykorzystaniem metody optymalizacyjnej można zaprezentować za pomocą tzw. krzywej zbieżności. Jest to najczęściej nierosnąca krzywa prezentująca wartości funkcji celu wraz z pojawianiem się kolejnych punktów w logu. W większości zastosowań prezentuje ona najlepszy dotychczas uzyskany wynik do pewnego punktu w logu, a zatem zmiany jej wartości następują w punkcie, którego wartość funkcji celu okazała się lepsza niż do tej pory obliczona. Przykładowa krzywa zbieżności dla problemu minimalizacji jest przedstawiona poniżej.



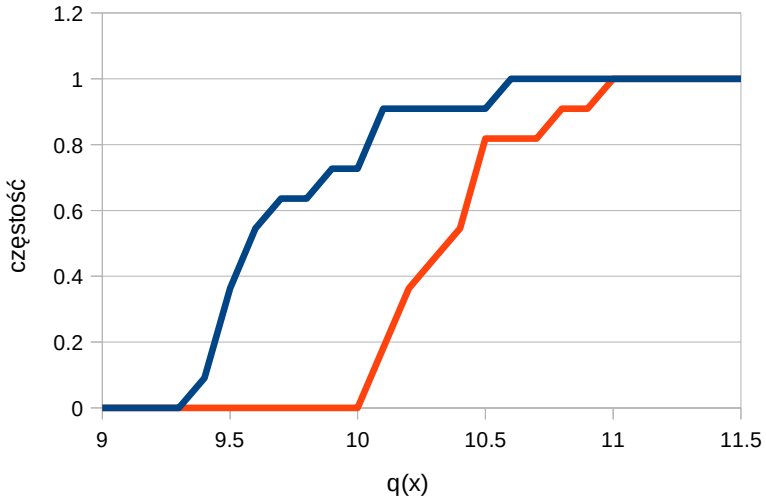
Można zadać sobie pytanie, która z dostępnych metod optymalizacyjnych jest najlepsza zawsze. Nie ma jednak prawidłowej odpowiedzi na tak postawione pytanie, co potwierdza No Free Lunch Theorem (NFL). NFL mówi, że jeżeli nie znamy z góry problemu optymalizacyjnego, który przyjdzie nam rozwiązać, to nie możemy dobrać efektywnej metody jego rozwiązania. Można to zobrazować w następujący sposób – mamy przed sobą dwa zegary. Pierwszy z nich stoi w miejscu i cały czas pokazuje tę samą godzinę, drugi natomiast chodzi dwie minuty spóźniony. Jak określić, który z nich jest lepszy? To zależy od tego, jak zostanie zdefiniowana lepszość. Jeśli przez lepszy zegar rozumiemy taki, który częściej pokazuje prawidłowy czas, to lepszy będzie pierwszy zegar. Jednak jeżeli zdefiniujemy lepszość jako np. średni błąd popełniany przez wskazania zegara w ciągu doby, to lepszy będzie drugi zegar. Tak samo jest z metodami optymalizacyjnymi – nie wiedząc, jaki problem przyjdzie nam rozpatrywać, nie możemy jednoznacznie stwierdzić, która z metod jest lepsza. Przy zdefiniowanym problemie istnieją rozwiązanią umożliwiające porównanie dwóch metod.

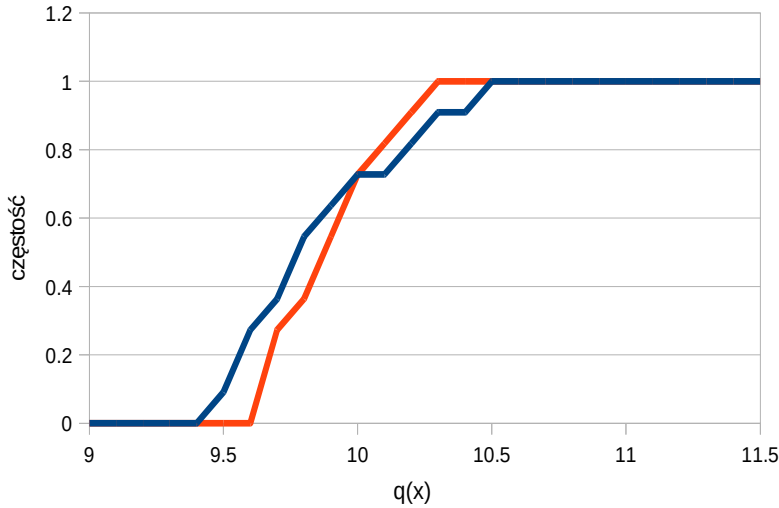
Do porównania dwóch losowych metod optymalizacyjnych można wykorzystać krzywe zbieżności wytworzone z logów wygenerowanych przez dane metody. Należy zauważyć, że krzywe zbieżnośći w obrębie jednej metody mogą się od siebie różnić, co może wynikać m.in. ze sposobu inicjalizacji punktu początkowego czy elementów losowości występujących w niektórych algorytmach. Przykładowe krzywe zbieżności dla dwóch losowych metod znajdują się poniżej.





Jak na podstawie takich dwóch zestawów krzywych określić, która z metod optymalizacji jest lepsza? Jednym ze sposobów jest dystrybuanta empiryczna. Określa ona – na podstawie powyższych krzywych – jak często jesteśmy w stanie osiągnąć zadany wynik po pewnym, określonym czasie. Im większa wartość dystrybuanty w danym miejscu, tym częściej dany wynik był osiagany. Interpretacja wartości dystrybuanty empirycznej jest podobna do dystrybuanty znanej z probabilistyki, czyli można ją uznać za prawdopodobieństwo osiągnięcia wartości funkcji celu nie gorszej niż sprawdzana w danym punkcie.





Powyżej znajdują się dwie pary dystrybuant empirycznych. Załóżmy, że kolorem niebieskim jest reprezentowany algorytm A, a czerwonym – algorytm B. Na podstawie lewego wykresu możemy stwierdzić, że algorytm A jest lepszy niż algorytm B. Potwierdza to położenie krzywych dystrybuant empirycznych – dystrybuanta empiryczna algorytmu A znajduje się nad dystrybuantą empiryczną algorytmu B, co oznacza, że algorytm A – po zadanym czasie – częściej lub z taką samą częstością osiąga pewien zadany wynik niż algorytm B. Prawy wykres nie pozwala nam stwierdzić lepszośći ani gorszości żadnej z metod. Objawia się to przecięciem w wykresach dystrybuant empirycznych, co nie pozwala wskazać metody, która jest „zawsze” lepsza. Metoda porównywania dystrybuant empirycznych jest przykładem testu nieparametrycznego.

Innym rodzajem są testy parametryczne, którego przykładem może być np. test t-Studenta. Zakłada on, że wartości funkcji celu wygenerowane przez dwie sprawdzane metody pochodzą z pewnego rozkładu normalnego. Test ten sprawdza, czy rozkłady te pochodzą z rozkładu o tej samej wartości średniej, a więc czy nie występuje między nimi statystyczna istotność między wartościami średnimi.

Testy parametryczne i nieparametryczne są rodzajami testów hipotez statystycznych. Sposób działania testów statystycznych jest następujący - na początku jest stawiana tzw. hipoteza zerowa (null hypothesis). Jest to hipoteza, którą najczęściej chcemy obalić. Następnie test – na podstawie przekazanych mu danych – oblicza tzw. p-wartość (p-value), która określa prawdopodobieństwo zaobserwowania danych takich, jakie zostały podane do testu przy założeniu, że hipoteza zerowa jest prawdziwa. Jeśli p-wartość jest mniejsza od pewnej, założonej wartości, to hipoteza zerowa jest odrzucana. Przykładowo, dla testu t-Studenta hipoteza zerowa mówi, że różnica między wartościami średnimi rozkładów, z których pochodzą przekazane dane, jest równa zero. W testach metod optymalizacyjnych potwierdzenie hipotezy zerowej oznacza, że nie jesteśmy w stanie określić, która z metod jest lepsza lub gorsza.

Przykładowymi testami statystycznymi są wyżej wymieniony test t-Studenta, nieparametryczny test Wilcoxona (hipoteza zerowa – mediany przekazanych do testu danych są równe), a także test chi kwadrat (hipoteza zerowa – dane przekazane do testu pochodzą z rozkładu chi kwadrat). Testy te są zaimplementowane w języku R, są wywoływane przez funkcje *t.test* [t-Studenta], *wilcox.test* [test Wilcoxona] oraz *chisq.test* [test chi kwadrat]. Dokładniejszy opis działania testów statystycznych znajduje się w książce „Statystyka dla studentów kierunków technicznych i przyrodniczych” Jacka Koronackiego i Jana Mielniczuka.

Po przeprowadzeniu testów statystycznych między sprawdzanymi metodami optymalizacji należy w pewien sposób zagregować otrzymane wyniki. Przy porównywaniu wielu algorytmów dla jednego zadania można posłużyć się tabelką, której *i*-ty rząd i kolumna reprezentują *i*-tą metodę optymalizacyjną. Komórka *i,j* takiej tabeli określa, czy algorytm *i* był lepszy niż algorytm *j* przy porówaniu w parach. Następnie w rzędach tabeli sumowane są wyniki porównań, gdzie wynik „lepszy” ma wagę 1, „gorszy” ma wagę -1, a „nie wiadomo” - 0. Tworzy to pewien bilans punktów dla każdego algorytmu, dzięki czemu możemy uszeregować je od najlepszego do najgorszego dla zadanego problemu. Przykładowa tabelka, bilans i ranga znajdują się na rysunkach poniżej.

	A1	A2	A3	A4
A1	.	+	+	-
A2	-	.	-	.
A3	-	+	.	-
A4	+	.	+	.

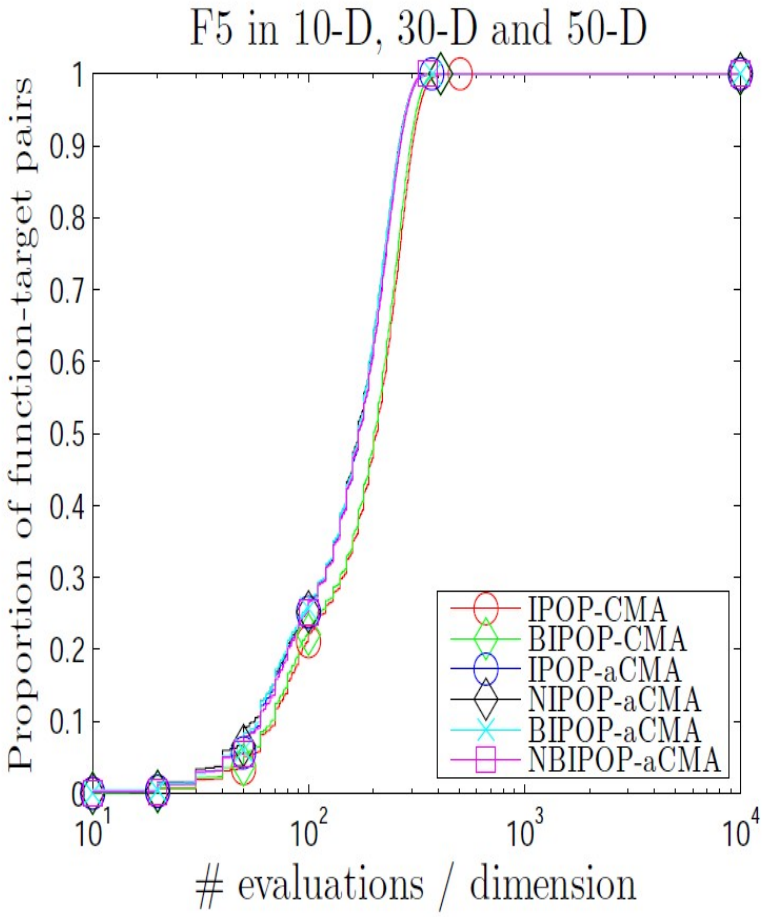
	bilans	ranga
A1	1	2
A2	-2	4
A3	-1	3
A4	2	1

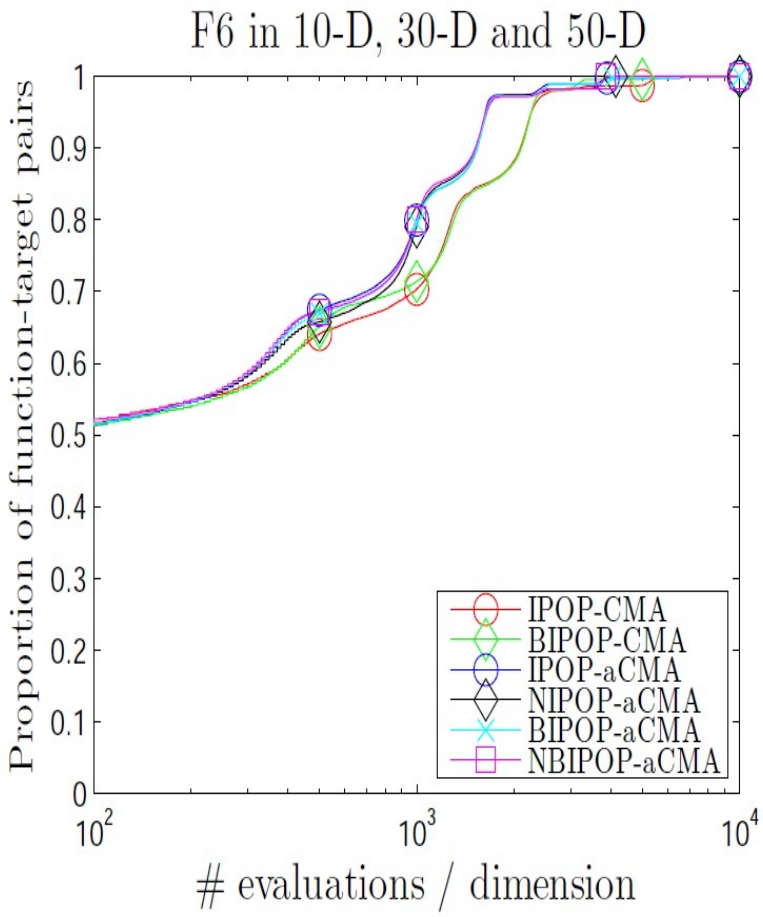
Porównanie algorytmów optymalizacji dla wielu zadań sprowadza się do przygotowania rang dla każdego zadań z osobna i obliczeniu średniej, co prezentuje poniższy rysunek.

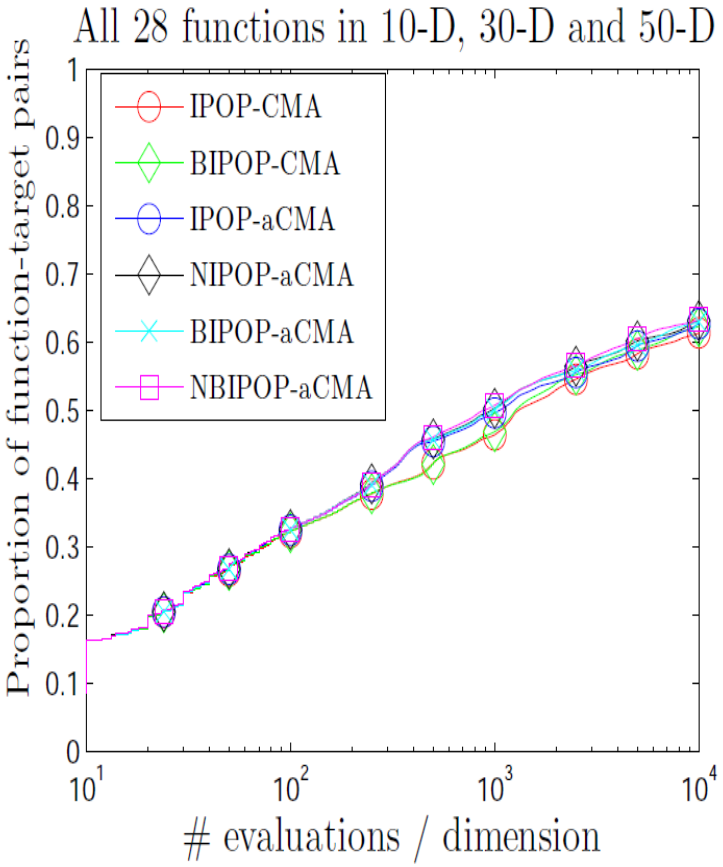
	Z1	Z2	Z3	Z4	Z5	średnia
A1	2	3	1	1	4	2.2
A2	4	2	1	2	1	2
A3	3	1	1	2	2	1.8
A4	1	1	1	3	3	1.8

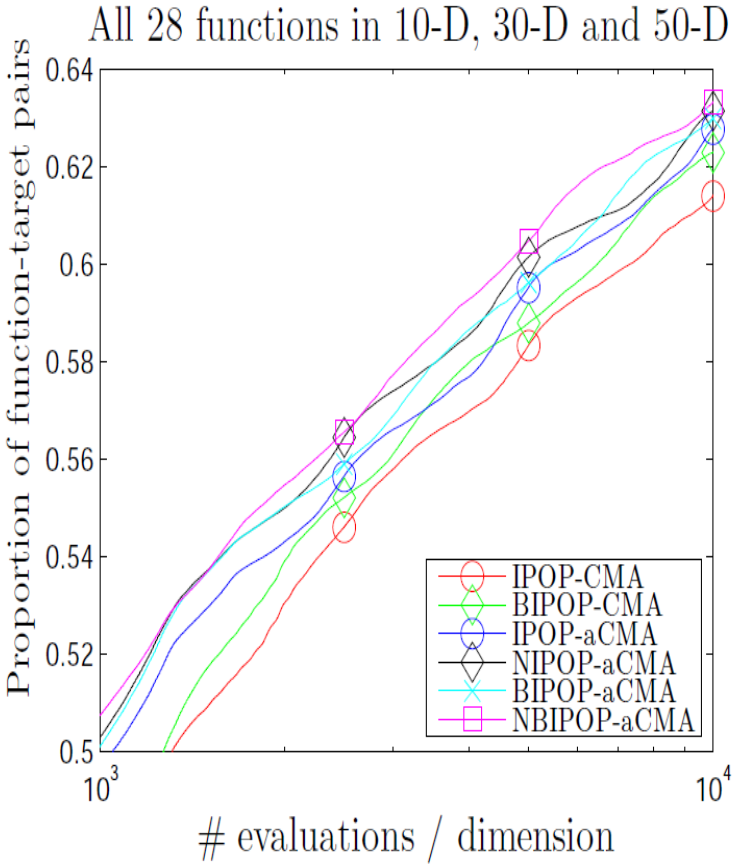
Widać na nim testy 4 różnych algorytmów dla 5 różnych problemów. Rangi dla każdego problemu zostały przygotowane zgodnie ze schematem postępowania przy porównywaniu wielu algorytmów dla jednego problemu. Średnia ranga jest średnią arytmetyczną osiągniętych rang. „Wygrywają” algorytmy, które maja najniższą średnią rangę, czyli w powyższym przykładzie – algorytmy A3 i A4.

Innym sposobem porównywania metod optymalizacyjnych są tzw. *function-target pairs*. Polega on na tym, że – na podstawie średniej krzywej zbieżności - tworzy się przedziały między pewną początkową a końcową wartością funkcji celu oddalone od siebie o jakiś odstęp (odstęp może być liniowy lub logarytmiczny). Dla wartości z osi X ustala się, ile z tych przedziałów jest zawartych między zadaną początkową wartością funkcji celu, a wartością funkcji celu dla wybranego punktu z osi X. Wartość ilorazu zawartych przedziałów przez liczbę wszystkich przedziałów jest wartością, która trafia na wykres dystrybuanty empirycznej dla danej metody. W ten sposób można łatwo stworzyć dystrybuanty empiryczne dla wielu problemów i porównać je. Sposób ten można także wykorzystać do porównania wielu algorytmów na wielu problemach – wystarczy po prostu uśrednić wyniki dla danej wartości z osi X. Warto wspomnieć, że w wielu rzeczywistych benchmarkach problemy są wielowymiarowe, a więc – w celu ujednolicenia skali – na osi X często podaje się liczbę wykonanych ewaluacji podzieloną przez liczbę wymiarów optymalizowanej funkcji (najczęściej w skali logarytmicznej).






All 28 functions in 10-D, 30-D and 50-D
Proportion of function-target pairs
evaluations / dimension
IPOP-CMA
BIPOP-CMA
IPOP-aCMA
NIPOP-aCMA
BIPOP-aCMA
NBIPOP-aCMA


All 28 functions in 10-D, 30-D and 50-D
Proportion of function-target pairs
evaluations / dimension
IPOP-CMA
BIPOP-CMA
IPOP-aCMA
NIPOP-aCMA
BIPOP-aCMA
NBIPOP-aCMA