# Parallel Multiobjective Optimization

KONSPEKT REFERATU

Na podstawie pracy autorstwa A. J. Nebro, F. Luna, E. -G. Talbi, E. Alba

## 1 Czym jest „multiobjective optimization”?

***MOP*** – Multiobjective Optimization Problem, czyli Wielocelowy Problem Optymalizacyjny.

- w ostatnich latach dużo uwagi poświęcono zagadnieniom optymalizacji problemów z więcej niż jedną funkcją celu, co jest spowodowane wielokryterialną naturą „życiowych” problemów,
- MOP'y mają wiele funkcji celu, które dążą do maksimum (lub minimum) i formują matematyczny opis kryterium wydajności i są na ogół ze sobą sprzeczne (ich „interesy” są sprzeczne),
- problemy życiowe mają wiele celów, przykład:

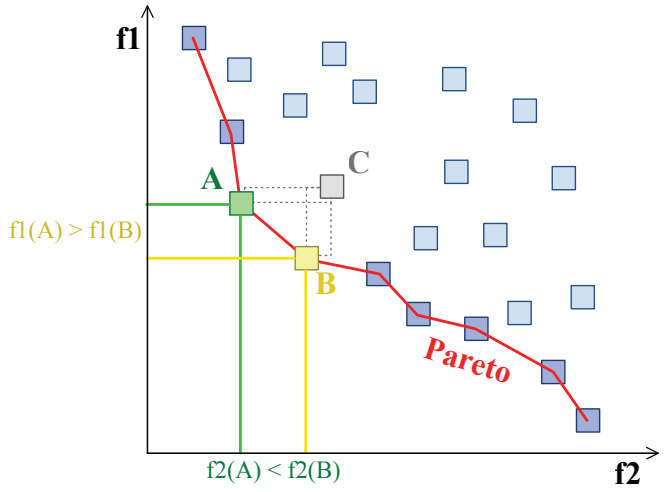
  Roważmy problem *trasowania z oknami czasowymi* [ang. *Vechicle Routing Problem with Time Windows*]. Problem ten możemy zdefiniować następująco: mamy centralę w której znajduje się magazyn oraz $n$ klientów, którym musimy dostarczyć zamówiony towar przy użyciu ciężarówek o ustalonej, identycznej pojemności. Wielkości zamówień są różne, tak samo jak ostateczne terminy realizacji zamówień. Dodatkowo znamy położenie centrali i klientów oraz znamy najkrótsze trasy między nimi.
  **Celem** jest znalezienie rozwiązania, które:

  – minimalizuje liczbę pojazdów (pierwsze kryterium),
  – minimalizuje całkowitą przebytą przez nie trasę (drugie kryterium).

  Łatwo zauważyć, że w rzeczywistych warunkach oba te kryteria są sprzeczne, gdyż zmniejszenie ilości ciężarówek pociągnie za sobą wydłużenie sumarycznej trasy przebytej przez pozostałe auta, które będą musiały pokonać niektóre trasy wielokrotnie.

- Formalnie, rozwiązanie MOP oznacza odnalezienie wektora $\vec{x}^* = [\vec{x}_1^*, \vec{x}_2^*, ..., \vec{x}_n^*]$, który spełnia $m$ nierówności $g_i(\vec{x}) \geq 0, i = 1, 2, ..., p$ i optymalizuje funkcję wektorową $\bar{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), ..., f_k(\vec{x})]^T$, gdzie $\bar{x} = [x_1, x_2, ..., x_n]^T$ jest wektorem zmiennych decyzyjnych.
- Ogólnie – wielokryterialna optymalizacja nie ogranicza się do odnalezienia unikalnego, pojedynczego rozwiązania, lecz raczej zbioru rozwiązań zwanych *rozwiązaniami niezdominowanymi* [ang. *nondominated solutions*]. Każde rozwiązanie z tego zbioru jest *optymalne w sensie Pareto.*

## 1.1 Pareto Front




Przykład *Pareto Front*. Kwadraty reprezentują możliwe wybory. Mniejsze wartości są preferowane.

**Definicja.** ***Optimum w sensie Pareto*** – termin opracowany przez włoskiego ekonomistę Vilfreda Pareta, oznacza taki podział dóbr, którego nie można już poprawić nie pogarszając jednocześnie sytuacji któregokolwiek z podmiotów.

- Załóżmy na przykład, że rozpatrujemy dwie osoby, Kowalskiego i Malinowskiego. Kowalski ma początkowo pewien zasób chleba, a Malinowski pewien zasób wody. Ponieważ obaj chcieliby mieć i jedno i drugie dobro, to zaczną wymieniać chleb na wodę.
- Oczywiście, jeżeli Kowalski ma tylko chleb, to pierwszy kubek wody będzie dla niego bardzo cenny i skłonny będzie do oddania dużej ilości chleba. Analogicznie, Malinowski będzie skłonny wymienić dużą ilość wody w zamian za kromkę chleba. W miarę kontynuowania wymiany ich skłonność do poświęcania jednego dobra w zamian za drugie będzie maleć. Ostatecznie osiągnięty zostanie taki punkt, w którym dalsza wymiana nie będzie już możliwa. Kowalski za kolejną kromkę chleba będzie sobie życzyć coraz więcej wody, a Malinowski za kolejny kubek wody będzie chciał coraz więcej chleba.
- W ten sposób osiągnięty został punkt optimum w sensie Pareto. Jeżeli bowiem chcielibyśmy Kowalskiemu dać kolejny kubek wody, to musielibyśmy zmusić Malinowskiego do wymiany, pogarszając tym samym jego sytuację (albowiem, gdyby jego sytuacja miała się poprawić, to do wymiany doszłoby dobrowolnie).
- Warto zauważyć, że wolna wymiana dóbr prowadzi do efektywnej alokacji w sensie Pareto. Co więcej, jeżeli alokacja nie jest efektywna w sensie Pareto, to można poprawić sytuację niektórych uczestników bez pogarszania sytuacji innych, co jest oczywiście bardzo pożądane. Dlatego alokacja nieefektywna w sensie Pareto zdarza się rzadko.
- Ponieważ efektywność w sensie Pareto odwołuje się do indywidualnych preferencji, to nie uwzględnia interesu społecznego, co jest największym ograniczeniem stosowania tego kryterium.
- Rozważmy na przykład budowę drogi. W interesie mieszkańców jest budowa drogi, jednak w tym celu należy wykupić kilka działek. Oczywiście działki te można kupić w drodze negocjacji z właścicielami, jednak okazuje się, że jeden z nich jest tak emocjonalnie przywiązany do danego terenu, że nie chce sprzedać działki za żadne pieniądze. Sytuacja, w której droga nie powstaje jest efektywna w sensie Pareto, albowiem nie udało się poprawić położenia mieszkańców bez pogarszania położenia jednego z właścicieli.

- Innym skrajnym przykładem alokacji efektywnej w sensie Pareto jest taki podział dóbr, w którym wszystkie dobra są u jednej osoby, a pozostałe nie mają nic. Taka alokacja również jest efektywna w sensie Pareto.

# 2 Pareto Front i MOP oraz zrównoleglanie

Techniki używane przy obliczaniu zbioru *rozwiązań niezdominowanych* można podzielić na *dokładne* [ang. *exact*] i *heurystyczne* [ang. *heuristic*].

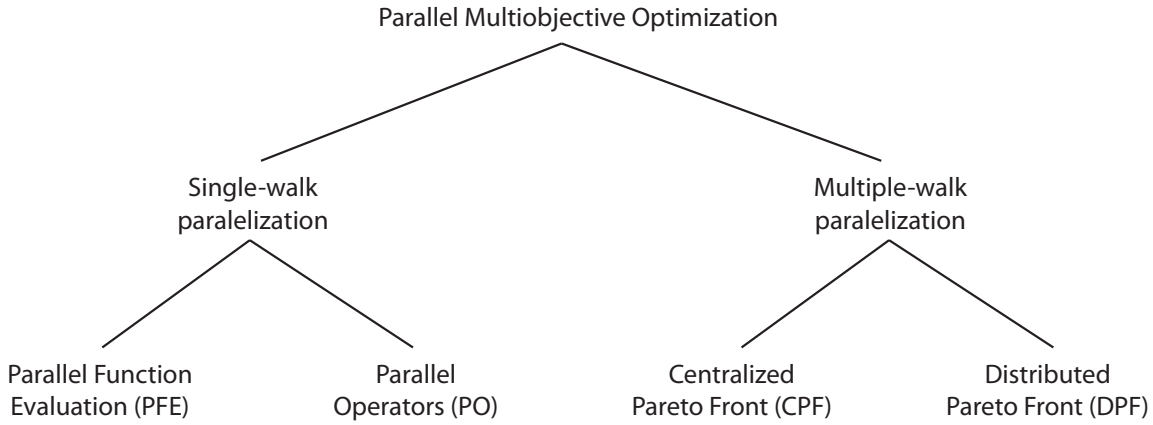
- Metody *dokładne* umożliwiają znalezienie zbioru rozwiązań optymalnych w sensie Pareto dla danego MOP, lecz ich czas działania jest wykładniczy w najgorszym przypadku – nie nadają się zatem do większości praktycznych zastosowań.
- Z drugiej zaś strony *heurystyki* nie gwarantują odnalezienia optymalnego rozwiązania, lecz dostarczają rozwiązania bliskie optymalnych przy wielu problemach optymalizacyjnych w znacząco skróconym (w stosunku do wykładniczego) czasie działania.
- W klasie heurystycznych metod optymalizacji *metaheurystyki* są podklasą która łączy w sobie podstawowe metody heurystyczne tworząc *framework* wyższego poziomu mający na celu wydajne i skuteczne badanie przestrzeni poszukiwań.

Algorytmy rozwiązywania MOP'ów zwykle używają skomplikowanych, nieliniowych funkcji celu. Nawet przy wykorzystaniu zaawansowanych heurystyk ich złożoność obliczeniowa jest bardzo wysoka, co czyni je bezużytecznymi w wielu realnych zastosowaniach. W tym kontekście zrównoleglenie algorytmów wydaje się właściwą metodą osiągnięcia rezultatów w sensownym czasie. W rzeczywistości równoległe algorytmy są szeroko używane na polu *jednocelowej optymalizacji* [ang. *mono-objective optimization*]. Dzięki nim udaje się rozwiązywać problemy szybciej lub (w sensownym czasie) rozwiązywać problemy o wyższym stopniu komplikacji.

Zrównoleglanie służy nie tylko rozwiązywaniu problemów szybciej, lecz przyczynia się także do powstawania wydajniejszych modeli poszukiwań: równoległy heurystyczny algorytm może być bardziej efektywny od sekwencyjnego nawet gdy jest wykonywany na pojedynczym procesorze.

## 2.1 Równoległe metaheurystyki dla MOP

Kilka taksonomii zostało zaproponowanych w celu sklasyfikowania równoległych implementacji metaheurystyk. Ta szerorko akceptowana głównie rozróżnia pomiędzy strategiami mającymi na celu przyspieszenie algorytmu sekwencyjnego (*Single-walk parallelization*) oraz strategiami które modyfikują zachowanie algorytmu sekwencyjnego w taki sposób by nie tylko przyspieszyć obliczenia ale również podnieść jakość rozwiązań (*Multiple-walk parallelization*).




Klasyfikacja równoległych metaheurystyk dla wielocelowych optymizacji.

Strategie mające na celu jedynie przyspieszanie obliczeń dzielimy na te, które zrównoleglają funkcję ewaluacji problemu optymalizacyjnego (PFE) oraz te, które zrównoleglają jeden lub więcej operatorów danej metody poszukiwań (PO). Z kolei w strategiach *Multiple-walk* wyróżniamy dwa podejścia:

- *Pareto Front* jest scentralizowanym elementem całego algorytmu (CPF) – mówimy wtedy o *global nondominated solutions*,
- *Pareto Front* jest rozproszony i zarządzany lokalnie przez każdy wątek przeszujujący przestrzeń w czasie obliczeń (DPF) –mówimy wtedy o *local nondominated solutions*.

Do tej pory nie istnieje żadna „czysta" implementacja algorytmu typu CPF, co jest podyktowane względami wydajnościowymi. Istniejące implementacje algorytmów CPF w pewnym stopniu wykorzystują fazy DPF w których lokalne, niedominujące rozwiązania są rozważane. Po każdej fazie DPF budowane jest pojedyncze optymalne rozwiązanie (w sensie Pareto), a następnie jest ono dystrybuowane do poszczególnych lokalnych węzłów obliczeń.

## 2.2 Kilka prac o równoległych algorytmach dla MOP

Teraz zaprezentjemy kilka prac w których zaproponowano rozwiązania pewnych rzeczywistych problemów przy użyciu zrównoleglonych algortmów optymalizacji wielocelowej.

- D.A. Linkens i H. Okola Nyongesa – *A Distributed Genetic Algorithm for Multivariable Fuzzy Control* (1993)

  Jest to algorytm *Multiple-Walk* nie polegający bezpośrednio na optymalności w sensie Pareto. Napisany w C dla topologii *full-connected*, gdzie niezdominowane rozwiązania są okresowo propagowane do wszystkich. Jest to *Diversity Control Genetic Algorithm* gdzie każdemu celowi przypisana jest jedna wyspa.

- D. Quagiarella, A. Vicini:
  - *Sub-Population Policies for Parallel Multiobjective Genetic Algorithm with Applicaton to Wing Design.* (1998),
  - *Multiobjective Approach to Transonic Wing Design by Means of Genetic Algorithms* (1999).

  Zaproponowany przez nich algorytm *Single-Walk* polegający na optymalności w sensie Pareto stosujący zrównoleglenie funkcji ewaluacji. Jest to równoległy algorytm genetyczny opierający się na topologii Master/Slave. Pomaga on w projektowaniu skrzydeł dla samolotów latających z prędkościami okołodźwiękowymi (w miarę zbliżania się do prędkości dźwięku pojawia się zjawisko gwałtownego bardzo silnego oporu [ang. *drag*] oddziaływującego na samolot).

- N. Jozefowicz, F. Semet i E. -G. Talbi – *Parallel and Hybrid Models for Multiobjective Optimization: Application to the Vehicle Routing Problem.* (2002)

  Do rozwiązywania problemu trasowania zaproponowano tu wieloprzebiegowy [ang. *Mutiple-Walk*] algorytm, również opierający się na optymalności w sensie Pareto, wykorzystujący rozproszone obliczanie *Pareto Front*. Algorytm jest hybrydą *algorytmu ewolucyjnego* i *tabu search*.

- L. S. Oliviera, R. Sabourin, F. Bortolozzi i C.Y. Suen – *A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition.*

  Zaproponowano tu algorytm rozpoznawania odręcznego pisma. Jest to wariant równoległego algorytmu NSGA [ang. *Non-dominated Sorting Genetic Algorithm*]. W tym jednoprzebiegowym algorytmie zrównoleglono funkcję ewaluacji.

# 3 Dwie równoległe wielocelowe metaheurystyki

## 3.1 Równoległy PAES – pPAES

***PAES*** – Pareto Archived Evolution Strategy jest dobrze znanym, wielocelowym, sekwencyjnym algorytmem.

**Definicja.** ***(1+1)-ES*** – Strategia ewolucji [ang. *Evolution Strategy*] jest prostą techniką optymalizacji bazującą na ideach adaptacji i ewolucji. Bazuje na populacji rozmiaru 2: bieżący punkt (rodzic) i rezultat jego mutacji. Mutant (potomek) staje się ojcem następnej generacji tylko w przypadku, gdy pewien zbiór jego właściwości jest „lepszy" niż odpowiedni zbiór właściwości jego ojca.

1. **(1+1)-PAES:**

   Reprezentuje najprostsze, nietrywialne podejście do tworzenia procedury wielocelowego lokalnego poszukiwania [ang. *multiobjective local search procedure*].

   Szkic tego algorytmu jest następujący:

```
1   generate initial random solution c and add it to the archive
2   while (termination criterion has not been reached)
3       mutate c to produce m and evaluate m
4       if (c dominates m)
5           discard m
6       else if (m dominates c)
7           replace c with m
8           add m to the archive
9       else if (m is dominated by any member of the archive)
10          discard m
11      else
12          apply test (c, m, archive) to determine
13          which becomes the new current solution
14          and whether to add m to the archive
15  end while
```

   - PAES bazuje na utrzymaniu jednego rozwiązania, które przy każdej iteracji jest mutowane by wygenerować nowego kandydata na rowiązanie (linia 3),
   - następnie algorytm sprawdza czy zaakceptować „wymutowane" rozwiązanie oraz czy zapamiętać je na liście niezdominowanych rozwiązań w kategoriach zadanego kryterium akceptacji,
   - naszym celem jest rozwiązywanie realnych problemów (których ewaluacja wymaga dużych zasobów obliczeniowych) zastosowano tu podstawowe udoskonalenie – jeśli operator mutacji nie modyfikuje bieżącego reprezentanta populacji $c$, wtedy ani dalsza ewaluacja ani archiwizacja tego osobnika nie ma miejsca. W takim przypadku kryterium zakończenia polega na osiągnięciu ustalonej wcześniej liczby wykonań funkcji testu (linia 12). To implikuje, że liczba iteracji jest mocno zależna od prawdopodobieństwa mutacji użytej w ES.

2. **pPAES:**

   Ten algorytm jest oparty o strategię zrównoleglania *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES. Jest to najprawdopodobniej pierwsza (udana) próba zrównoleglenia algorytmu PAES. Zasada działania algorytmu jest następująca:

   - Każdy proces wykonuje algorytm PAES przez zdefiniowaną wcześniej liczbę ewaluacji (wykonania) funkcji testu utrzymując przy tym własne, lokalne archiwum niezdominowanych rozwiązań (strategia DPF). To archiwum jest zbiorem rozwiązań optymalnych w sensie Pareto i ma taką samą maksymalną moc (powiedzmy $N$) we wszystkich wątkach wykonania.
   - Okresowo następuje wymiana rozwiązań pomiędzy wątkami w topologii pierścienia. Proces migracji rozwiązań polega na:

  - określeniu ile i które rozwiązania (reprezentanci poplulacji) powinny być przesłane. Algorytm pPAES migruje po jednym reprezentancie w jednej chwili, który jest losowo wybierany z lokalnego *Pareto Front* wątku. Następnie nowy osobnik jest dołączany jak gdyby był mutantem na drodze „zwyczajnej" ewolucji w lokalnym algorytmie PAES (linia 3),
  - określeniu częstotliwości z jaką ma następować wymiana rozwiązań. Jako że liczba iteracji w poszczególnych wątkach może być różna wymiana rozwiązań w algorytmie pPAES odbywa się co określoną, stałą liczbę kroków, zwaną *częstotliwością migracji* [ang. *migration frequency*], mierzoną w kategoriach ilości ewaluacji funkcji testu (dlatego każdy wątek migruje taką samą liczbę osobników, niezależnie od ilości wykonanych iteracji).

- Ostatnim krokiem jest skonstruowanie finalnego *Pareto Front*. Ten front będzie miał tą samą ilość niedominujących rozwiązań co pojedynczy wątek PAES. Zatem, jeśli pPAES używa $p$ równoległych wątków, to najwyżej $p \cdot N$ rozwiązań optymalnych w sensie Pareto może zostać obliczonych (może się zdarzyć, że nie wszystkie wątki dostarczą $N$ rozwiązań). Wszystkie wątki pPAES przesyłają ich niezdominowane rozwiązania do wyróżnionego procesu który łączy je w jeden *Pareto Front*.

## 3.2 Zrównoleglony, wielocelowy ssGA

Drugim algorytmem jest ssGA [ang. *Solid State Genetic Algorithm*. Służy on do rozwiązywania problemu pokrycia obszaru sygnałem radiowym przy użyciu jak najmniejszej liczby nadajników [ang. *Radio Network Design problem*]. Zarys algorytmu równoległego wzoruje się na zwykłym ssGA, gdzie występuje tylko jedna zamiana na pokolenie. Użyto klasycznych operacji krzyżowania i mutacji przystosowując je do wymagań problemu RND. Aby sprostać wielu celom ssGA używa **rang** by posortować populację zgodnie z definicją dominacji Pareto, **współdzielenia** by sprostać różnorodności niedominujących rozwiązań i **„elitarności"** [ang. *elitism*] by przyspieszyć proces zbieżności.

W prawdziwym problemie RND w przestrzeni poszukiwań jest bardzo wiele rozwiązań a koszt obliczeń funkcji celu jest bardzo wysoki. Występuje również duże zapotrzebowanie pamięciowe. Można więc zaproponować trzy modele algorytmu ssGA, uwzględniając różne cele:

- by poprawić jakość *Pareto front*,
- by przyspieszyć wyszukiwanie,
- aby umożliwić rozwiązywanie dużych problemów.

A oto te modele:

- model równoległy, polegający na wieloprzebiegowym rozproszonym obliczaniu *Pareto front*, bazujący na modelu migracyjnym algorytmów ewolucyjnych (zaadaptowanym dla MO),
- model równoległy asynchroniczny (strategia jednoprzebiegowa, polegająca na równoległym wyliczaniu funkcji celu),
- model równoległy synchroniczny, w którym wyliczenie pojedynczego rozwiązania odbywa się przez podzielenie obszaru geograficznego i przetwarzanie poszczególnych rejonów równolegle. Jednoprzebiegowy.

Jak widać pierwsze dwa modele nie są związane z RND, zaś trzeci model jest określony szczególnie dla niego.

# 4 Wydajność

Obydwa algorytmy zostały użyte do rozwiązania problemu *Cellular Radio Network Design*. Analiza czasów działania algorytmu pPAES dla jednego, dwóch, ośmiu i szesnastu procesorów pokazuje, że choć ogólny czas działania algorytmu skraca się w miarę zwiększania liczby procesorów, to zauważalnie rośnie czas poświęcany na komunikację przez co gwałtownie spada efektywność algorytmu. Dla czterech procesorów wynosi ona prawie 96%, dla ośmiu nieco ponad 75% zaś dla szesnastu już tylko 57%, dalsze zwiększanie liczby procesorów wiąże się z tak dużym spadkiem efektywności, że zysk czasowy tak osiągnięty jest nieopłacalny. Widać więc, że zrównoleglony PAES jest słabo skalowalny. Co do algorytmu ssGA, model opierający się na równoległym wyliczaniu funkcji celu dobrze się skaluje, gdyż całkowite wyliczenie pojedynczego rozwiązania jest czasochłonnym zadaniem w porównaniu z czasami komunikacji. Metoda podziału zadania na mniejsze jest bezpośrednio zależna od ilości procesorów (ilość obszarów) i stopniowo traci na efektywności dla więcej niż szesnastu procesorów.

# Spis treści