

Parallel Multiobjective Optimization

Michał Kraszewski Mateusz Styrzcula

Instytut Informatyki Uniwersytetu Wrocławskiego

18 listopada 2008

MOP – Multiobjective Optimization Problem.

MOP – Multiobjective Optimization Problem.

- optymalizacja problemów z więcej niż jedną funkcją celu,

MOP – Multiobjective Optimization Problem.

- optymalizacja problemów z więcej niż jedną funkcją celu,
- cele na ogół są sprzeczne,

Vehicle Routing Problem

Vehicle Routing Problem with Time Windows

- centrala z magazynem oraz n klientów,
- musimy dostarczyć towar,

Celem jest znalezienie rozwiązania, które:

- minimalizuje liczbę pojazdów,
- minimalizuje całkowitą przebytną przez nie trasę.

Rozwiązanie MOP oznacza odnalezienie wektora

$$\vec{x}^* = [\vec{x}_1^*, \vec{x}_2^*, \dots, \vec{x}_n^*]$$

który spełnia m nierówności

$$g_i(\vec{x}) \geq 0, \quad i = 1, 2, \dots, p$$

i optymalizuje funkcję wektorową

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T$$

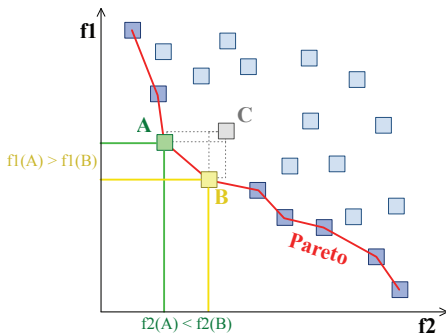
gdzie

$$\vec{x} = [x_1, x_2, \dots, x_n]^T$$

jest wektorem zmiennych decyzyjnych.

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.



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

Techniki używane przy obliczaniu zbioru *rozwiązań niezdominowanych* można podzielić na:

- *dokładne* [ang. *exact*],
- *heurystyczne* [ang. *heuristic*].

- Metody *dokładne* umożliwiają znalezienie zbioru rozwiązań optymalnych w sensie Pareto dla danego MOP, lecz nie nadają się do większości praktycznych zastosowań,

- Metody *dokładne* umożliwiają znalezienie zbioru rozwiązań optymalnych w sensie Pareto dla danego MOP, lecz nie nadają się do większości praktycznych zastosowań,
- *heurystyki* nie gwarantują odnalezienia optymalnego rozwiązania,

- Metody *dokładne* umożliwiają znalezienie zbioru rozwiązań optymalnych w sensie Pareto dla danego MOP, lecz nie nadają się do większości praktycznych zastosowań,
- *heurystyki* nie gwarantują odnalezienia optymalnego rozwiązania,
- *metaheurystyki* tworzą podklasę łączącą w sobie podstawowe metody heurystyczne,

Zrównoleglenie

- Zrównoleglenie algorytmów wydaje się właściwą metodą osiągnięcia rezultatów w sensownym czasie.

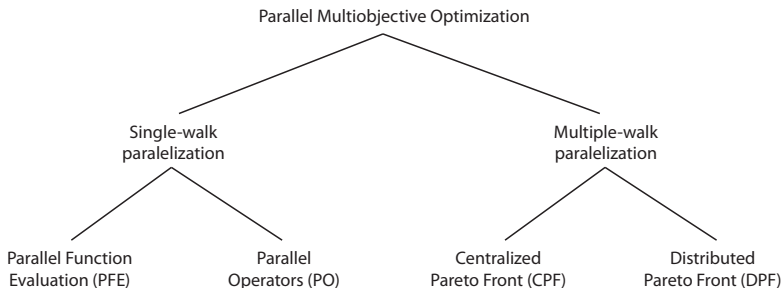
Zrównoleglenie

- Zrównoleglenie algorytmów wydaje się właściwą metodą osiągnięcia rezultatów w sensownym czasie.
- Równoległe algorytmy są szeroko używane na polu *jednocelowej optymalizacji* [ang. *mono-objective optimization*].

Zrównoleglenie

Zrównoleglenie 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.

Różne warianty zrównoleglenia „parallel multiobjective”



Klasyfikacja równoległych metaheurystyk dla wielocelowych optymalizacji.

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

- D.A. Linkens i H. Okola Nyongesa
 - *A Distributed Genetic Algorithm for Multivariable Fuzzy Control* (1993),
- D. Quagiarella, A. Vicini:
 - *Sub-Population Policies for Parallel Multiobjective Genetic Algorithm with Application to Wing Design* (1998),
 - *Multiobjective Approach to Transonic Wing Design by Means of Genetic Algorithms* (1999),

- D.A. Linkens i H. Okola Nyongesa
 - *A Distributed Genetic Algorithm for Multivariable Fuzzy Control* (1993),
- D. Quagiarella, A. Vicini:
 - *Sub-Population Policies for Parallel Multiobjective Genetic Algorithm with Application to Wing Design* (1998),
 - *Multiobjective Approach to Transonic Wing Design by Means of Genetic Algorithms* (1999),
- N. Jozefowicz, F. Semet i E. -G. Talbi
 - *Parallel and Hybrid Models for Multiobjective Optimization: Application to the Vehicle Routing Problem.* (2002)

- D.A. Linkens i H. Okola Nyongesa
 - *A Distributed Genetic Algorithm for Multivariable Fuzzy Control* (1993),
- D. Quagiarella, A. Vicini:
 - *Sub-Population Policies for Parallel Multiobjective Genetic Algorithm with Application to Wing Design* (1998),
 - *Multiobjective Approach to Transonic Wing Design by Means of Genetic Algorithms* (1999),
- N. Jozefowicz, F. Semet i E. -G. Talbi
 - *Parallel and Hybrid Models for Multiobjective Optimization: Application to the Vehicle Routing Problem.* (2002)
- 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.*

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)-PAES

```
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
```

Szkic algorytmu.

pPAES

- jest oparty o strategię zrównoleglenia *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES,

pPAES

- jest oparty o strategię zrównoleglenia *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES,
- Zasada działania:

pPAES

- jest oparty o strategię zrównoleglenia *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES,
- Zasada działania:
 - Każdy proces wykonuje algorytm PAES przez zdefiniowaną wcześniej liczbę wykonań funkcji testu utrzymując przy tym własne, lokalne archiwum niezdominowanych rozwiązań (strategia *DPF*).

pPAES

- jest oparty o strategię zrównoleglenia *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES,
- Zasada działania:
 - Każdy proces wykonuje algorytm PAES przez zdefiniowaną wcześniej liczbę wykonań funkcji testu utrzymując przy tym własne, lokalne archiwum niezdominowanych rozwiązań (strategia *DPF*).
 - Okresowo następuje wymiana rozwiązań pomiędzy wątkami:

pPAES

- jest oparty o strategię zrównoleglenia *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES,
- Zasada działania:
 - Każdy proces wykonuje algorytm PAES przez zdefiniowaną wcześniej liczbę wykonań funkcji testu utrzymując przy tym własne, lokalne archiwum niezdominowanych rozwiązań (strategia *DPF*).
 - Okresowo następuje wymiana rozwiązań pomiędzy wątkami:
 - określenie ile i które rozwiązania powinny być przesłane,

pPAES

- jest oparty o strategię zrównoleglenia *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES,
- Zasada działania:
 - Każdy proces wykonuje algorytm PAES przez zdefiniowaną wcześniej liczbę wykonań funkcji testu utrzymując przy tym własne, lokalne archiwum niezdominowanych rozwiązań (strategia *DPF*).
 - Okresowo następuje wymiana rozwiązań pomiędzy wątkami:
 - określenie ile i które rozwiązania powinny być przesłane,
 - określeniu częstotliwości z jaką ma następować wymiana rozwiązań,

pPAES

- jest oparty o strategię zrównoleglenia *Multiple-walk*, *Distributed Pareto Front* w której każdy wątek realizuje algorytm (1+1)-PAES,
- Zasada działania:
 - Każdy proces wykonuje algorytm PAES przez zdefiniowaną wcześniej liczbę wykonań funkcji testu utrzymując przy tym własne, lokalne archiwum niezdominowanych rozwiązań (strategia *DPF*).
 - Okresowo następuje wymiana rozwiązań pomiędzy wątkami:
 - określenie ile i które rozwiązania powinny być przesłane,
 - określeniu częstotliwości z jaką ma następować wymiana rozwiązań,
 - Konstrukcja finalnego *Pareto Front*.

Solid State Genetic Algorithm

Zastosowany do rozwiązywania problemu pokrycia obszaru sygnałem radiowym przy użyciu jak najmniejszej liczby nadajników [ang. *Radio Network Design problem*].

Solid State Genetic Algorithm

Zastosowany do rozwiązywania problemu pokrycia obszaru sygnałem radiowym przy użyciu jak najmniejszej liczby nadajników [ang. *Radio Network Design problem*].

Zrównoleglony algorytm bazuje na zwykłym **ssGA**.

Solid State Genetic Algorithm

Zastosowany do rozwiązywania problemu pokrycia obszaru sygnałem radiowym przy użyciu jak najmniejszej liczby nadajników [ang. *Radio Network Design problem*].

Zrównoleglony algorytm bazuje na zwykłym **ssGA**.

Aby sprostać wielu celom używa:

Solid State Genetic Algorithm

Zastosowany do rozwiązywania problemu pokrycia obszaru sygnałem radiowym przy użyciu jak najmniejszej liczby nadajników [ang. *Radio Network Design problem*].

Zrównoleglony algorytm bazuje na zwykłym **ssGA**.

Aby sprostać wielu celom używa:

- **rang**,

Solid State Genetic Algorithm

Zastosowany do rozwiązywania problemu pokrycia obszaru sygnałem radiowym przy użyciu jak najmniejszej liczby nadajników [ang. *Radio Network Design problem*].

Zrównoleglony algorytm bazuje na zwykłym **ssGA**.

Aby sprostać wielu celom używa:

- **rang,**
- **współdzielenia,**

Solid State Genetic Algorithm

Zastosowany do rozwiązywania problemu pokrycia obszaru sygnałem radiowym przy użyciu jak najmniejszej liczby nadajników [ang. *Radio Network Design problem*].

Zrównolegiony algorytm bazuje na zwykłym **ssGA**.

Aby sprostać wielu celom używa:

- **rang**,
- **współdzielenia**,
- „**elitarności**” [ang. *elitism*].

Solid State Genetic Algorithm

- 1 bardzo duża przestrzeń rozwiązań,

Solid State Genetic Algorithm

- 1 bardzo duża przestrzeń rozwiązań,
- 2 koszt obliczeń funkcji celu jest bardzo wysoki,

Solid State Genetic Algorithm

- 1 bardzo duża przestrzeń rozwiązań,
- 2 koszt obliczeń funkcji celu jest bardzo wysoki,
- 3 występuje duże zapotrzebowanie pamięciowe,

Solid State Genetic Algorithm

- 1 bardzo duża przestrzeń rozwiązań,
- 2 koszt obliczeń funkcji celu jest bardzo wysoki,
- 3 występuje duże zapotrzebowanie pamięciowe,

Zaproponowano trzy modele mające na celu:

Solid State Genetic Algorithm

- 1 bardzo duża przestrzeń rozwiązań,
- 2 koszt obliczeń funkcji celu jest bardzo wysoki,
- 3 występuje duże zapotrzebowanie pamięciowe,

Zaproponowano trzy modele mające na celu:

- poprawę jakości *Pareto Front*,

Solid State Genetic Algorithm

- 1 bardzo duża przestrzeń rozwiązań,
- 2 koszt obliczeń funkcji celu jest bardzo wysoki,
- 3 występuje duże zapotrzebowanie pamięciowe,

Zaproponowano trzy modele mające na celu:

- poprawę jakości *Pareto Front*,
- przyspieszenie przeszukiwania,

Solid State Genetic Algorithm

- 1 bardzo duża przestrzeń rozwiązań,
- 2 koszt obliczeń funkcji celu jest bardzo wysoki,
- 3 występuje duże zapotrzebowanie pamięciowe,

Zaproponowano trzy modele mające na celu:

- poprawę jakości *Pareto Front*,
- przyspieszenie przeszukiwania,
- umożliwienie rozwiązywania dużych problemów.

Obydwa algorytmy zostały użyte do rozwiązania problemu
Cellular Radio Network Design.

Obydwa algorytmy zostały użyte do rozwiązania problemu
Cellular Radio Network Design.

- pPAES,

Obydwa algorytmy zostały użyte do rozwiązania problemu
Cellular Radio Network Design.

- pPAES,
- ssGA,

KONIEC