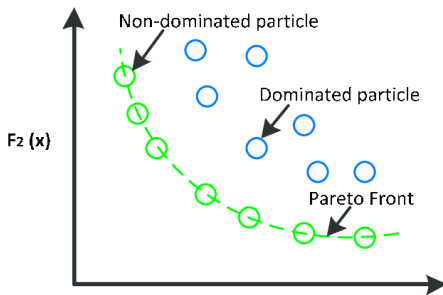
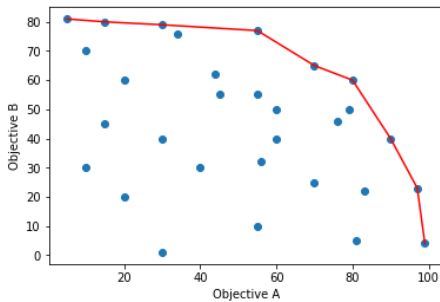


Algorytmy optymalizacji wielokryterialnej SPEA i SPEA2

Jakub Kuciński

31 stycznia 2021

Front Pareto



SPEA(N, \bar{N}):

$P_0 = \text{Initial-Population}()$

$\bar{P}_0 = \emptyset, t = 0$

while not Termination-Condition():

$\bar{P}_{t+1} = \text{Update-archive}(P_t, \bar{P}_t)$

 Fitness-assignment(P_t, \bar{P}_t)

$P_{t+1} = \text{Selection}(P_t, \bar{P}_t)$

 Recombination(P_{t+1})

 Mutation(P_{t+1})

 Increment(t)

return \bar{P}_t

Oznaczenia:

N, \bar{N} – rozmiar populacji, maksymalny rozmiar archiwum

P_t – populacja w t -tej iteracji

\bar{P}_t – archiwum w t -tej iteracji

Uaktualnianie archiwum:

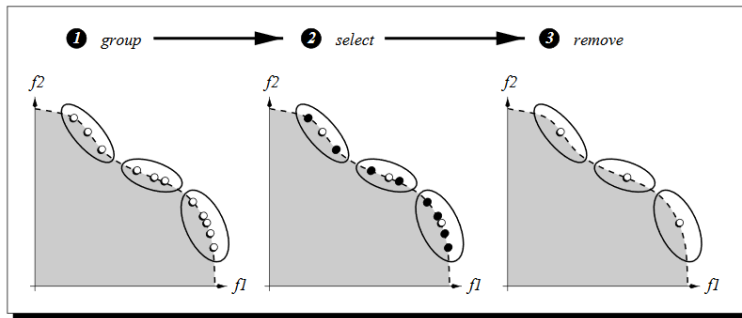
- Skopiuj archiwum \overline{P}_t do \overline{P}' .
- Skopiuj osobniki niezdominowane w P_t do \overline{P}' .
- Usuń z \overline{P}' osobniki zdominowane.
- Jeśli liczba osobników w zbiorze \overline{P}' przekracza \overline{N} to zmniejsz ich liczbę przez Clustering.
- Przypisz \overline{P}' do \overline{P}_{t+1} .

- Zainicjalizuj $\mathbf{C} = \bigcup_{i \in \overline{\mathbf{P}}'} \{\{i\}\}$.
- Jeśli $|\mathbf{C}| \leq \overline{N}$ to przeskocz do ostatniego punktu.
- Policz odległość (w przestrzeni funkcji celu) między wszystkimi klastrami $c_1, c_2 \in \mathbf{C}$:

$$d_c = \frac{1}{|c_1||c_2|} \cdot \sum_{\substack{i_1 \in c_1 \\ i_2 \in c_2}} d(i_1, i_2)$$

- Połącz dwa klastry znajdujące się najbliżej siebie. Powrót do drugiego punktu.
- Każdy klaster zastąp jego centroidem. Zwróć zbiór tych centroidów.

Ilustracja Clusteringu



Rysunek: Eckart Zitzler 1999. Diss. ETH No. 13398. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications

Obliczanie przystosowania $F(i)$ osobników:

- Najpierw każdemu osobnikowi $i \in \overline{\mathbf{P}}_t$ przypisywana jest wartość $S(i) \in [0, 1)$ proporcjonalna do liczby dominowanych przez nią osobników z \mathbf{P}_t . Formalnie:

$$S(i) = \frac{|\{j \mid j \in \mathbf{P}_t \wedge i \succ j\}|}{N + 1}$$

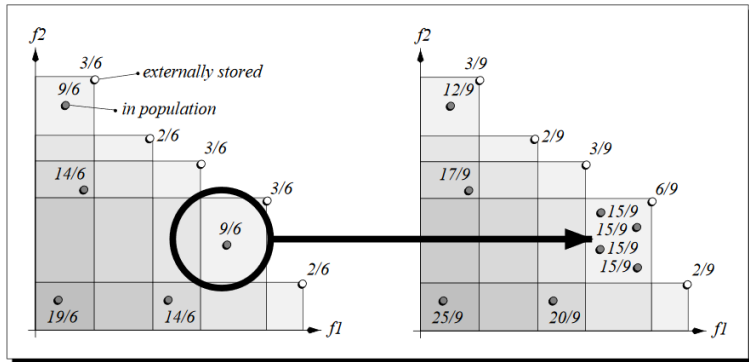
Przystosowanie i jest równe $S(i)$.

- Przystosowanie osobnika $j \in \mathbf{P}_t$ wylicza się przez sumowanie wartości S wszystkich osobników z $\overline{\mathbf{P}}_t$, które dominują j :

$$F(j) = 1 + \sum_{\substack{i \in \overline{\mathbf{P}}_t \\ i \succ j}} S(i)$$

Dodanie jedynki zapewnia, że osobniki z $\overline{\mathbf{P}}_t$ będą miały lepsze przystosowanie od osobników z \mathbf{P}_t .

Komentarz do przystosowania



Rysunek: Eckart Zitzler 1999. Diss. ETH No. 13398. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications

W SPEA jako algorytm selekcji został zaproponowany binarny turniej na sumie zbiorów P_t i \bar{P}_t .

- Osobniki z populacji, które są dominowane przez te same osobniki z archiwum mają taką samą wartość przystosowania (w szczególności jeśli w archiwum jest jeden osobnik to cała populacja ma to samo przystosowanie).
- Jeśli dużo osobników w populacji jest nieporównywalnych, to mamy bardzo mało informacji z dominowania. SPEA nie wykorzystuje informacji o gęstości osobników w populacji.
- Clustering co prawda pozwala zachować charakterystykę frontu w archiwum, ale obcina brzegowe osobniki.

SPEA2(N, \bar{N}):

$P_0 = \text{Initial-Population}()$

$\bar{P}_0 = \emptyset, t = 0$

while not Termination-Condition():

 Fitness-assignment(P_t, \bar{P}_t)

$\bar{P}_{t+1} = \text{Update-archive}(P_t, \bar{P}_t)$

$P_{t+1} = \text{Selection}(\bar{P}_{t+1})$

 Recombination(P_{t+1})

 Mutation(P_{t+1})

 Increment(t)

return \bar{P}_t

- Każdemu osobnikowi i z P_t oraz \bar{P}_t przypisujemy wartość $S(i) = |\{j \mid j \in \bar{P}_t \cup P_t \wedge i \succ j\}|$ odpowiadającą liczbie osobników, które dominuje.
- Dla każdego osobnika obliczane jest surowe przystosowanie:

$$R(i) = \sum_{\substack{j \in \bar{P}_t \cup P_t \\ j \succ i}} S(j)$$

- Dla każdego osobnika i obliczana jest gęstość σ_i^k równa odległości do k -tego najbliższego osobnika (w przestrzeni funkcji celu). Zazwyczaj przyjmuje się $k = \sqrt{N + \bar{N}}$. Gęstość określa wzór:

$$D(i) = \frac{1}{\sigma_i^k + 2}$$

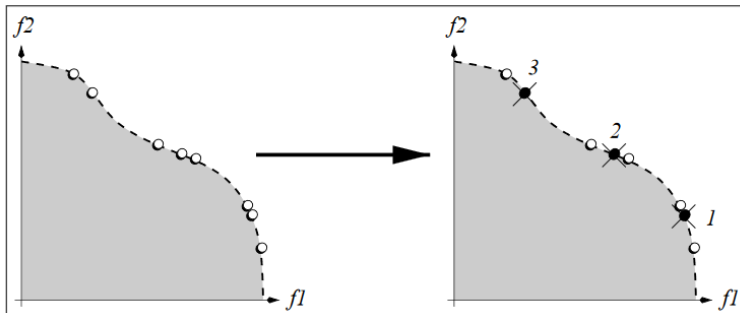
- Końcowe przystosowanie osobnika określa wzór:

$$F(i) = R(i) + D(i)$$

- Najpierw kopiowane zostają niezdominowane osobniki z populacji i archiwum do nowego archiwum:
 $\overline{\mathbf{P}}_{t+1} = \{i \mid i \in \mathbf{P}_t \cup \overline{\mathbf{P}}_t \wedge F(i) < 1\}.$
- Jeśli $|\overline{\mathbf{P}}_{t+1}| < \overline{N}$ to do $\overline{\mathbf{P}}_{t+1}$ zostają dodane najlepsze według przystosowania $\overline{N} - |\overline{\mathbf{P}}_{t+1}|$ osobniki.
- Jeśli $|\overline{\mathbf{P}}_{t+1}| > \overline{N}$ to iteracyjnie usuwane zostają osobniki z $\overline{\mathbf{P}}_{t+1}$ do momentu, aż $|\overline{\mathbf{P}}_{t+1}| = \overline{N}$. Procedura odcinania osobnika polega na wyborze takiego osobnika, który jest w najmniejszej odległości do innego osobnika. Dla osobników o równej minimalnej odległości porównywana jest odległość do drugiego najbliższego itd. Formalnie wybierany jest taki osobnik i , że $\forall_{j \in \overline{\mathbf{P}}_{t+1}} i \leq_d j$, gdzie:

$$i \leq_d j \Leftrightarrow \forall_{0 < k < |\overline{\mathbf{P}}_{t+1}|} \sigma_i^k = \sigma_j^k \vee \\ \exists_{0 < k < |\overline{\mathbf{P}}_{t+1}|} \left[\left(\forall_{0 < l < k} \sigma_i^l = \sigma_j^l \right) \wedge \sigma_i^k < \sigma_j^k \right]$$

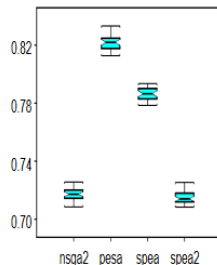
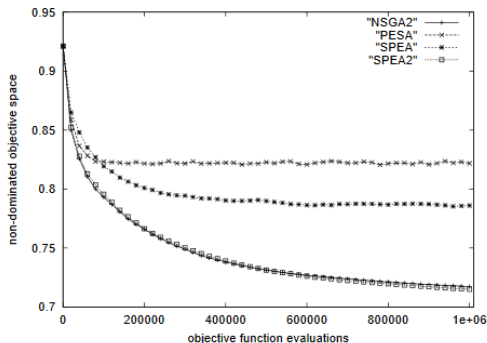
Ilustracja odcinania



Rysunek: Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar 2001.
SPEA2: Improving the strength pareto evolutionary algorithm

W SPEA2 algorytm selekcji to binarny turniej na zbiorze \overline{P}_{t+1} .

QV (Quagliarella and Vicini 1997)		
100 min	$[-5, 5]^n$	$f_1(x) = \left(\frac{1}{n} \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)\right)^{\frac{1}{4}}$ $f_2(x) = \left(\frac{1}{n} \sum_{i=1}^n ((x_i - 1.5)^2 - 10 \cos(2\pi(x_i - 1.5)) + 10)\right)^{\frac{1}{4}}$

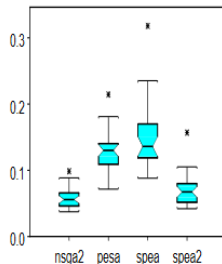
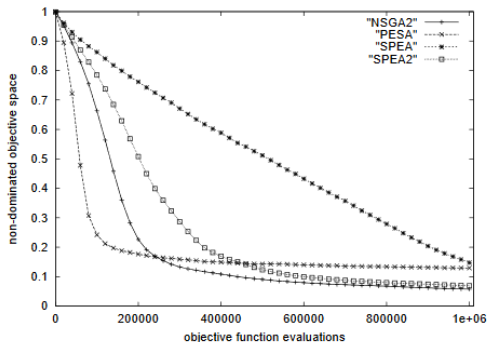


Rysunek: Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar 2001.

SPEA2: Improving the strength pareto evolutionary algorithm

Funkcje ciągłe cd.

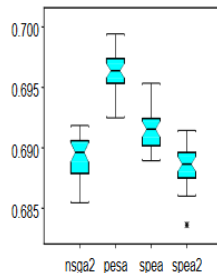
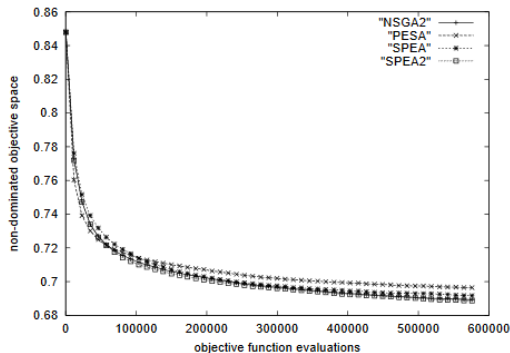
KUR (Kursawe 1991)		
100 min	$[-10^3, 10^3]^n$	$f_1(x) = \sum_{i=1}^{n-1} (-10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}})$ $f_2(x) = \sum_{i=1}^n (x_i ^{0.8} + \sin^3(x_i))$



Rysunek: Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar 2001.
SPEA2: Improving the strength pareto evolutionary algorithm

Problem plecakowy dla $m=3$

KP-750- m (Zitzler and Thiele 1999)		
750 max	$\{0, 1\}^n$	$f_j(x) = \sum_{i=1}^n x_i \cdot p_{i,j}$ s.t. $g_j(x) = \sum_{i=1}^n x_i \cdot w_{i,j} \leq W_j$ $p_{i,j}$ (profit values) and $w_{i,j}$ (weight values) randomly chosen $1 \leq j \leq m, \quad m = 2, 3, 4$

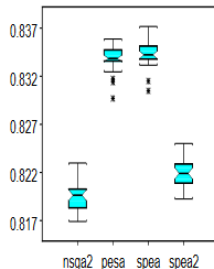
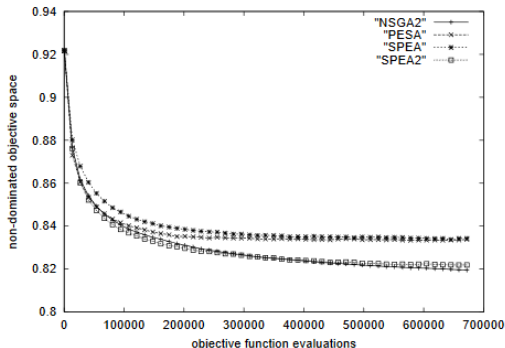


Rysunek: Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar 2001.

SPEA2: Improving the strength pareto evolutionary algorithm

Problem plecakowy dla $m=4$



KP-750- m (Zitzler and Thiele 1999)		
750 max	$\{0, 1\}^n$	$f_j(x) = \sum_{i=1}^n x_i \cdot p_{i,j}$ s.t. $g_j(x) = \sum_{i=1}^n x_i \cdot w_{i,j} \leq W_j$ $p_{i,j}$ (profit values) and $w_{i,j}$ (weight values) randomly chosen $1 \leq j \leq m, \quad m = 2, 3, 4$



Rysunek: Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar 2001.

SPEA2: Improving the strength pareto evolutionary algorithm

- SPEA2 daje znacząco lepsze wyniki od SPEA
- SPEA2 ma bardzo podobną wydajność do NSGA2
- NSGA2 znajduje szerszy front Pareto od SPEA2
- SPEA2 zwraca bardziej równomiernie rozmieszczone osobniki od NSGA2

-  Eckart Zitzler (1999). Diss. ETH No. 13398. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*
-  Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar (2001). *SPEA2: Improving the strength pareto evolutionary algorithm*