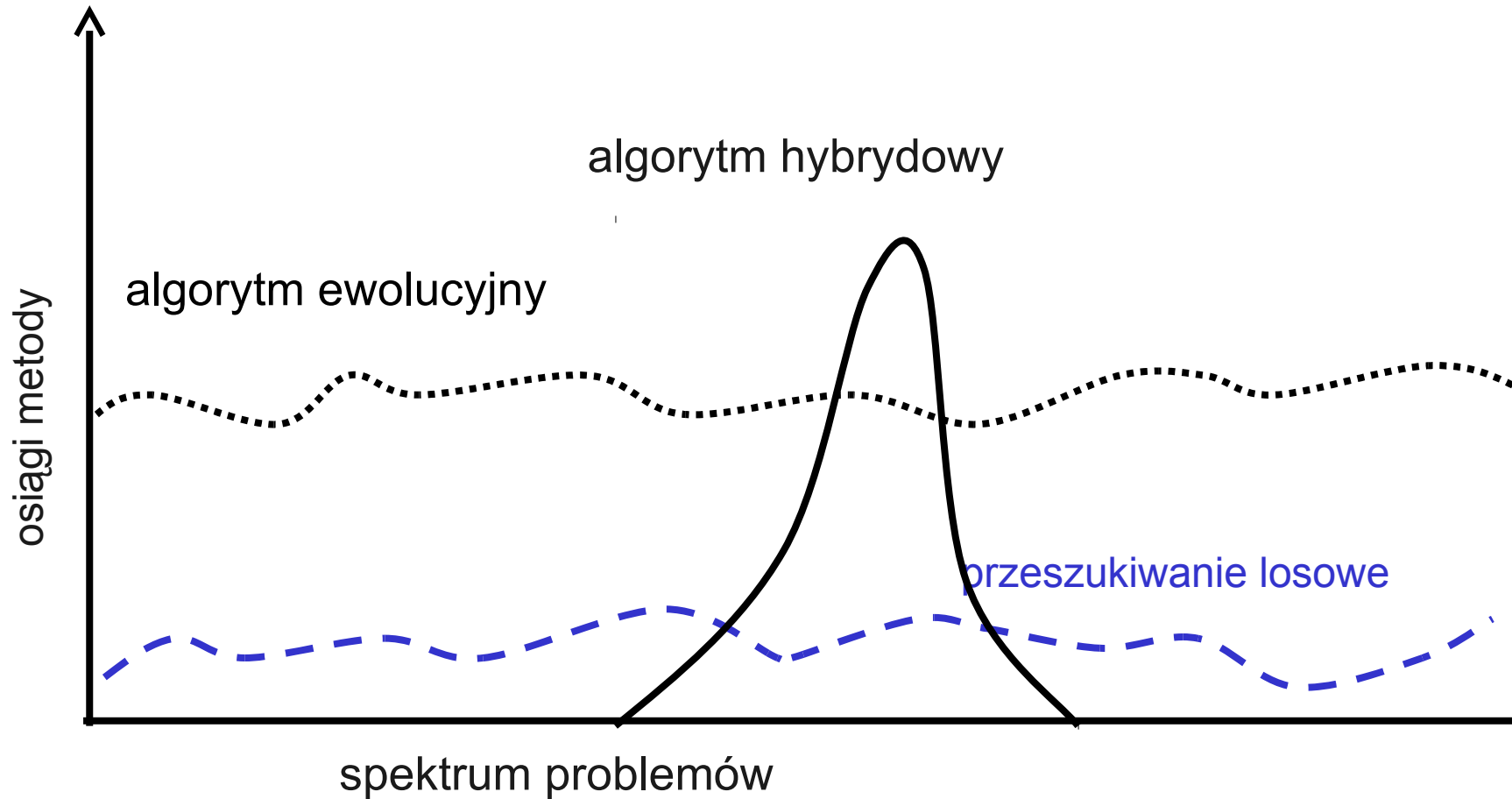


Algorytmy memetyczne (hybrydowe algorytmy ewolucyjne)

Wstęp

- Termin zaproponowany przez Pablo Moscato (1989).
- Kombinacja algorytmu ewolucyjnego z algorytmem poszukiwań lokalnych, tak że algorytm poszukiwań lokalnych jest zintegrowany w pętli algorytmu ewolucyjnego.
 - Nazywane również hybrydowymi algorytmami ewolucyjnymi.
- Termin stosuje się również do algorytmów używających wiedzy specyficznej dla problemu w operatorach rekombinacji. (krzyżowanie i mutacja).
- Algorytmy memetyczne są często o rząd wielkości szybsze od algorytmów ewolucyjnych i w wielu dziedzinach są najlepszymi znanymi algorytmami heurystycznymi.
- Praktyka wykazuje, że klasyczne algorytmy ewolucyjne są dobre w odnajdywaniu obszaru, w którym znajduje się optimum, ale kiepsko radzą sobie w odnalezieniu optimum lokalnego wewnątrz tego obszaru.
 - Co łatwo sprawdzić na Standardowym Algorytmie Genetycznym, z reprezentacją binarną i funkcją dopasowania=liczbą jedynek w łańcuchu binarnym.

Dlaczego algorytm hybrydowy



- Interpretacja prof. Zbigniewa Michalewicza

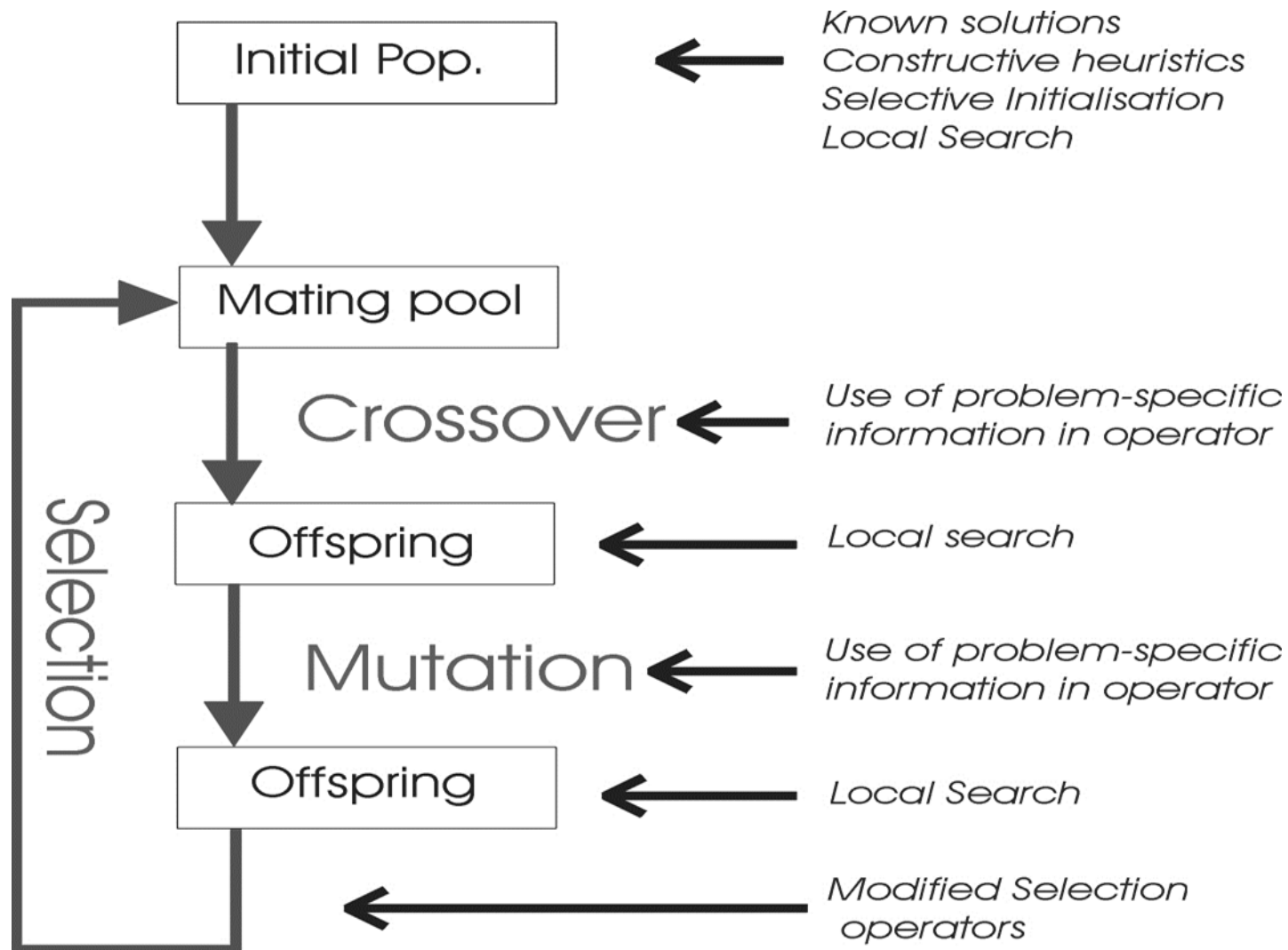
Optymalizacja lokalna - pseudokod

```
/* dane jest rozwiązanie początkowe x i funkcja sąsiedztwa N*/  
i=0;  
best=x  
while(!depth_condition()) {  
    count=0  
    while(!pivot_rule()) {  
        count=count+1  
        generate next neighbour  $y \in N(x)$   
        if  $f(y) > f(best)$  best=y  
    }  
    x=best  
    i=i+1  
return best
```

Algorytmy optymalizacji lokalnej - klasyfikacja

- pivot rule może to być reguła największego wzrostu (ang. steepest ascent) w której pętli wewnętrznej kończymy po przeszukaniu całego sąsiedztwa (i wybieramy rozwiązanie, które najlepiej zwiększa dopasowanie) albo reguła zachłannego wzrostu, w której pętli wewnętrznej kończymy po znalezieniu w sąsiedztwie pierwszego rozwiązania o większym dopasowaniu,
- dept_condition – warunek maksymalnej głębokości przeszukiwania lokalnego. Może zmieniać się od wykonania jednej iteracji, aż do przeszukiwania aż do odnalezienia punktu lokalnie optymalnego, którego nie możemy ulepszyć przeszukując sąsiedztwo.
- Funkcja sąsiedztwa $N(x)$ ma kluczowe znaczenie dla osiągnięcia algorytmu. Często jest definiowana jako zbiór punktów w przestrzeni rozwiązań, które możemy osiągnąć z punktu x stosując pewien operator zmiany. Przykładem są 2-sąsiedztwo i 3-sąsiedztwo w problemie komiwojażera.
- Metoda jest rzędu zerowego gdy używa wyłącznie wartości dopasowania, rzędu pierwszego gdy używa pierwszej pochodnej (wektora gradientu), rzędu drugiego gdy używa również drugiej pochodnej (macierzy hesjanu).

Możliwe miejsca hybrydyzacji w pętli algorytmu (Eiben & Smith, 2003)



„Inteligentna” inicjalizacja

- Wielokrotne przeszukiwanie lokalne (multi-start local search). Osobnik w populacji początkowej jest otrzymany poprzez zastosowanie algorytmu optymalizacji lokalnej do losowo wygenerowanego rozwiązania.
- Selektywna inicjalizacja stwórz b. duża populację $n \cdot S$ rozwiązań początkowych i wykorzystaj z niej S najlepszych.
 - Zalecany sposób to organizacja S turniejów, w każdym z nich udział bierze n losowo wybranych osobników. Najlepszy w turnieju przechodzi do populacji początkowej.
- Wprowadź do populacji kilka znanych rozwiązań uzyskanych klasycznymi heurystykami.
- Zastosuj losowe zachłanne algorytmy konstrukcyjne (np. dla TSP zacznij od losowego wierzchołka i włączaj kolejno losowo wierzchołki tak, aby długość cyklu cząstkowego była za każdym razem minimalna).
- Użyj jednej z powyższych metod do zidentyfikowania kilku obiecujących rozwiązań, a następnie sklonuj je w populacji, stosując do każdego kлона operator mutacji z dużym prawdopodobieństwem (masowa mutacja).
- **Uwaga na różnorodność** w populacji początkowej !

„Inteligentne” operatory krzyżowania i mutacji

- Operatory starające się stworzyć potomka o jak największym dopasowaniu.
- Operator DPX dla problemu komiwojażera, który kopiuje do potomka krawędzie występujące u obydwu rodziców, a następnie łączy je używając heurystyki najbliższego sąsiada.
- Operator krzyżowania jednopunktowego wybierający punkt przecięcia, tak aby potomek miał możliwie wysokie dopasowanie.
- Operator mutacji wybierający losowo kilka bitów, a następnie zmieniający bit który powoduje najkorzystniejszą zmianę dopasowania.
- Mnóstwo tego typu przykładów w literaturze.

Dodatkowy krok optymalizacji lokalnej na osobnikach uzyskanych w wyniku krzyżowania i mutacji

- Dodatkowy powoduje uruchomienie metody optymalizacji lokalnej na potomku otrzymanym w wyniku krzyżowania i mutacji. (potomek służy do zainicjalizowania algorytmu optymalizacji lokalnej).
- Może być interpretowany jako uczenie się podczas trwania życia osobnika. Istnieją dwa tryby pracy:
 - ewolucja Lamarckowska wynik optymalizacji lokalnej zastępuje w populacji osobnika który służył do zainicjalizowania algorytmu lokalnego. Odpowiada to sytuacji, w której osobnik jest w stanie przekazać do genotypu swoich potomków wiedzę nabytą podczas życia. Czy jest to zgodne z realiami biologicznymi ?
 - ewolucja Baldwinowska, osobnik otrzymuje dopasowanie, ale nie genotyp rozwiązania wyznaczonego w wyniku optymalizacji lokalnej.

Wariacje na temat optymalizacji lokalnej potomków

- Których potomków optymalizować ?
 - Wszystkich
 - Najlepszych
 - Najgorszych
 - Losowo wybranych
- Ile kroków optymalizacji ?
- Odpowiedź na powyższe dwa pytania staje się istotna, szczególnie w sytuacji gdy dysponujemy ograniczonym budżetem mocy obliczeniowej.
 - Zbyt mało optymalizacji lokalnej → nie uzyskamy zbieżności do optimum.
 - Zbyt mało ewolucji → utkniemy w optimum lokalnym.
- Czy przeszukiwanie lokalne wykonywane jest w przestrzeni genotypu czy w przestrzeni fenotypu ?

Funkcja sąsiedztwa

- Udowodniono teoretycznie (Krasnogor 2002) , że funkcja sąsiedztwa użyta przez algorytm przeszukiwania lokalnego powinna być istotnie różna od funkcji sąsiedztwa stosowanej w operatorze mutacji.
- Czy w kontekście projektu TSP (jedyne projektu obejmującego algorytm memetyczny) zastosowanie lokalnej optymalizacji opartej na algorytmie 2-opt oraz mutacji w postaci inwersji to dobry pomysł ?

Ochrona różnorodności w populacji

- Jeżeli przeszukiwanie lokalne prowadzimy do momentu odnalezienia rozwiązania lokalnie optymalnego, znacznie wzrasta ryzyko przedwczesnej zbieżności, zwłaszcza przy ewolucji lamarkowskiej.
 - Wynik optymalizacji lokalnej wykonanej na dwóch osobnikach ulokowanych w obszarze przyciągania tego samego minimum lokalnego będzie identyczny.
- Wiele różnych podejść w celu walki z tym problemem.
 - Stosuj regułę lamarkowską tylko dla 5% populacji.
 - Jeżeli inicjalizujesz populację kilkoma dobrymi rozwiązaniami, to pozostałą część inicjuj losowo.
 - Użyj operatorów krzyżowania, takich jak DPX zaprojektowanych aby zachowywać różnorodność.
 - Zmodyfikuj kryterium akceptacji rozwiązania podczas optymalizacji lokalnej (Krasnogor & Smith, 2002).

Zmodyfikowane kryterium akceptacji sąsiada

- Niech $\Delta E = F(\text{sąsiad}) - F(x)$ (zakładamy zadanie maksymalizacji)
- Prawdopodobieństwo akceptacji p dane jest wzorem:

$$p = \begin{cases} 1 & \text{jeżeli } \Delta E \geq 0 \\ \exp\left(\frac{k * \Delta E}{F_{\max} - F_{\text{avg}}}\right) & \text{w przeciwnym wypadku} \end{cases}$$

gdzie k jest stałym parametrem, F_{\max} i F_{avg} to największe i przeciętne dopasowanie w populacji.

- prawdopodobieństwo akceptacji tym większe im mniejsza różnorodność w populacji (mierzona różnicą F_{\max} i F_{avg})
- prawdopodobieństwo akceptacji tym mniejsze im gorszy jest sąsiad od aktualnego rozwiązania.

Algorytmy memetyczne kolejnych generacji

- Opisana przeze metoda, w której przeszukiwanie lokalne działało na potomku otrzymanym w wyniku krzyżowania i mutacji jest kwalifikowana jako prosty algorytm memetyczny albo algorytm memetyczny pierwszej generacji.
- Algorytmy memetyczne drugiej generacji. Stosuj kilka różnych metod przeszukiwań lokalnych.
 - Wybieraj je adaptacyjnie na podstawie pewnych reguł (np. gdy populacja jest różnorodna stosuj algorytm o niewielkim rozmiarze sąsiedztwa, w przeciwnym wypadku algorytm o dużym rozmiarze sąsiedztwa).
 - Wybieraj je samo-aptacyjnie, zakodowując w chromosomie numer algorytmu.
- Algorytmy memetyczne 3 generacji: samodzielnie konstruuaj algorytm przeszukiwań lokalnych i koduj go w chromosomie.

Memetic Overkill ?

- Craenen i Eiben (2005) zastosowali algorytmy memetyczne do rozwiązania problemów CSP (constraint satisfaction problem).
- 3 z 4 algorytmów memetycznych stały się istotnie lepsze po „wyłączeniu” komponentu ewolucyjnego.
 - selekcja rodziców i sukcesja losowa niezależnie od wartości dopasowania.
- Ironia losu: heurystyki dodano do algorytmów ewolucyjnych w celu ich ulepszenia, usunięcie ewolucji ulepsza je jeszcze bardziej.