ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2008
Seria: AUTOMATYKA z. 150 Nr kol. 1796

Czesław SMUTNICKI
Politechnika Wrocławska

KONWENCJONALNE I NOWE METODY ROZWIĄZYWANIA PROBLEMÓW DYSKRETNYCH

Streszczenie. Przedstawiono przegląd metod rozwiązywania problemów optymalizacji dyskretnej ze szczególnym uwzględnieniem nowych, obiecujących kierunków konstrukcji algorytmów przybliżonych.

CONVENTIONAL AND NEW SOLUTION METHODS FOR DISCRETE PROBLEMS

Summary. There is presented the review of solution methods for discrete optimization problems. Special attention has been paid to new, promising approaches leading to construction of approximate algorithms.

1. Wprowadzenie

Zdecydowana większość deterministycznych problemów planowania i sterowania w dyskretnych sytemach wytwarzania, projektowania struktur, planowania transportu, rozmieszczenia, podejmowania decyzji etc. jest formułowana w postaci zagadnień optymalizacji, w których wszystkie zmienne decyzyjne (bądź ich część) przyjmują wartości dyskretne, całkowitoliczbowe lub binarne. Zadania takie, nazywane dalej problemami optymalizacji *dyskretnej* (odpowiednio *mieszanej*), należą do klasy problemów wyjątkowo kłopotliwych z obliczeniowego punktu widzenia. W dalszym ciągu będziemy je zapisywać w postaci $\min_{x\in\mathcal{X}} K(x)$, gdzie $K(x)$ jest funkcją celu, zaś $\mathcal{X}$ jest zbiorem rozwiązań dopuszczalnych określonym przez zestaw warunków ograniczających. Głównymi powodami tych kłopotów są: brak "klasycznych" analitycznych własności (różniczkowalność, liniowość, itp.), wieloekstremalność z wykładniczą liczbą ekstremów lokalnych, NP-trudność przypadków pochodzących z praktyki oraz przekleństwo wymiarowości. Istnienie wielu ekstremów lokalnych, kłopotliwe już dla przypadku optymalizacji ciągłej (patrz np. funkcja testowa Langermanna), nabiera dla problemów dyskretnych szczególnego znaczenia, bowiem trajektoria przejścia przez przestrzeń rozwiązań przypomina przebiegi stochastyczne. Zbiór problemów lokowanych poniżej bariery NP-trudności jest mocno ograniczony. Z kolei bezpośrednią konsekwencją NP-trudności jest to, że czas obliczeń odpowiedniego algorytmu komputerowego jest funkcją wykładniczą od rozmiaru rozwiązywanego problemu. Przy uwzględnieniu rozmiaru realnych problemów praktycznych czas ten wyraża się dziesiątkami tysiącleci najszybszych znanych obecnie komputerów (to właśnie przekleństwo!).

Wielokrotnie, w celu uniknięcia kłopotów, zamiast rozwiązywać problem dokładnie, próbuje się wyznaczyć pewne jego rozwiązanie przybliżone. Dokładność tego przybliżenia posiada tendencję przeciwstawną do czasu obliczeń, tzn. uzyskanie dokładniejszego rozwiązania wymaga dłuższego czasu pracy algorytmu, przy czym ta ostatnia zależność ma charakter silnie nieliniowy. Z tego też powodu dziedzina dyskretnych procesów wytwarzania charakteryzuje się znaczną rozmaitością zarówno modeli, jak i metod rozwiązywania, zwykle dedykowanych dla wąskich klas zagadnień. Ograniczenie ogólności modeli ma na celu wykrycie tych szczególnych własności problemu, których umiejętne wykorzystanie w algorytmie zdecydowanie poprawia jego cechy numeryczne, takie jak czas obliczeń, szybkość zbiegania do rozwiązania optymalnego. Często dla tego samego problemu NP-trudnego występuje w literaturze kilka, kilkanaście różnych algorytmów o istotnie różnych cechach numerycznych. Znajomość dziedziny oraz metod rozwiązywania pozwala na właściwy dobór dla każdego nowo postawionego problemu odpowiedniego algorytmu satysfakcjonującego użytkownika. Należy przy tym pamiętać, że w rozważanej dziedzinie celem nadrzędnym *nie jest* sformułowanie jakiegokolwiek modelu i metody rozwiązania, lecz podanie *prostego* modelu oraz metody rozwiązywania *racjonalnej* z punktu widzenia implementowanego algorytmu komputerowego.

2. Metody dokładne

Metoda dokładna wyznacza rozwiązanie globalnie optymalne, tzn. $x^* \in \mathcal{X}$ takie, że $K^* \stackrel{\text{def}}{=} K(x^*) = \min_{x \in \mathcal{X}} K(x)$. W grupie tej, w zależności od przynależności problemu do klasy złożoności obliczeniowej, są: (a) efektywne algorytmy dedykowane, (b) metody oparte na schemacie podziału i ograniczeń (B&B), (c) metody oparte na schemacie programowania dynamicznego (PD), (d) metody oparte na programowaniu liniowym całkowitoliczbowym (PLC), (e) metody oparte na programowaniu liniowym binarnym (PLB), (f) metody subgradientowe. Metody (a) są uważane za tanie obliczeniowo metody specjalizowane dla problemów należących do P-klasy lub NP-trudnych problemów liczbowych. Metody (b) – (f) są kosztownymi obliczeniowo metodami dedykowanymi dla rozwiązywania problemów silnie NP-trudnych. Do lat końca lat 80. uważano je za "jedynie słuszne" podejście dla problemów silnie NP-trudnych, potem pojawiła się bariera wymiaru. Mimo iż dokonano znacznego postępu w ich rozwoju, praktycy wciąż uważają je za nieatrakcyjne bądź też ograniczają ich zastosowanie do bardzo wąskiego zakresu. Metody są czaso- i pamięciochłonne, zaś rozmiar problemów, które można rozwiązać w rozsądnym czasie, jest ciągle zbyt mały. Co więcej, implementacja bardziej złożonych algorytmów tego typu wymaga dużego doświadczenia programistycznego. Poważnym problemem jest także wiarygodność danych wejściowych, które często ulegają zaburzeniu tuż po kosztownym wyznaczeniu rozwiązania optymalnego, oraz tzw. nadmierna sztywność problemu. Można powiedzieć, że koszt poszukiwania rozwiązania optymalnego jest jeszcze zbyt duży w porównaniu z zyskami otrzymanymi z wdrożenia otrzymanego rozwiązania. Z drugiej strony istnieje grupa problemów, dla których zastosowanie metod dokładnych jest w pełni uzasadnione, a nawet wskazane. Przykładowo, poszukiwanie minimalnego czasu cyklu dla powtarzalnego procesu produkcyjnego z niewielkim repertuarem wyrobów zwielokrotnia nawet niewielkie zyski otrzymane w jednym cyklu, mnożąc je przez liczbę wykonanych cykli.

3. Błąd przybliżenia

Zbiór danych liczbowych problemu specyfikuje *przykład konkretny* Z tego problemu. Oznaczmy przez $\mathcal{X}(Z)$ zbiór wszystkich rozwiązań problemu dla tego przykładu, zaś przez $K(x;Z)$ wartość kryterium K dla rozwiązania x w przykładzie Z. Rozwiązanie $x^* \in \mathcal{X}(Z)$ takie, że $K(x^*;Z) = \min\{K(x;Z)\colon x \in \mathcal{X}(Z)\}$, jest nazywane rozwiązaniem optymalnym dla przykładu Z. Niech $x^A \in \mathcal{X}(Z)$ oznacza rozwiązanie przybliżone generowane przez algorytm A dla przykładu Z. Błąd przybliżenia algorytmu A oceniamy w relacji do odpowiedniej wartości optymalnej, np. jako $B^A(Z) = \left|K(x^A;Z) - K(x^*;Z)\right|$, $S^A(Z) = K(x^A;Z)/K(x^*;Z)$, czy też $T^A(Z) = (K(x^A;Z) - K(x^*;Z))/K(x^*;Z)$. Zależność błędu od Z może być badana eksperymentalnie lub analitycznie. Analiza eksperymentalna jest metodą subiektywną, bowiem zależy od wyboru próbki Z-tów, lecz w kontekście twierdzenia "no free lunch" jedynie ona pozwala uzasadnić wyższość wybranego algorytmu obserwowaną dla wąskiej grupy przykładów Z. Trudniejsze do wykonania analizy najgorszego przypadku i probabilistyczna dostarczają ocen niezależnych od przykładów Z. Analizy te dostarczają odmiennych, uzupełniających, czasami bardziej właściwych charakterystyk zachowania się algorytmu. Wyniki wszystkich analiz, w połączeniu z oceną złożoności obliczeniowej, stanowią kompletną charakterystykę algorytmu.

Analiza eksperymentalna jest najbardziej popularną metodą mimo jej niedoskonałości. Ocenia a posteriori zachowania się algorytmu (błąd przybliżenia, czas pracy algorytmu) w oparciu o wyniki otrzymane dla ograniczonej, *reprezentatywnej próbki* przykładów konkretnych Z. Ponieważ różni badacze różnie rozumieją pojęcie "próbka reprezentatywna", otrzymywane wyniki nie zawsze są wzajemnie porównywalne. Próbki mogą być ustalone (publiczne przykłady testowe) lub generowane losowo. Dla problemów NP-trudnych $K(x^*;Z)$ mogą być zastąpione pewną wartością referencyjną $K(x^{Ref};Z)$.

Analiza najgorszego przypadku ocenia a priori zachowanie się wybranego błędu na *całej* populacji przykładów konkretnych Z. Najczęściej stosowana jest ona do błędu $S^A(Z)$, dla którego definiuje się *współczynnik najgorszego przypadku* $\eta^A = \min\{y : S^A(Z) \leqslant y,\ \forall Z\}$ oraz *asymptotyczny współczynnik najgorszego przypadku* $\eta^A_\infty = \min\{y : S^A(Z) \leqslant y,\ \forall Z \in \{W : K(x^*;W) \geqslant L\}\}$, gdzie L jest pewną liczbą.

Analiza probabilistyczna zakłada, że każdy przykład konkretny Z został otrzymany jako realizacja n niezależnych zmiennych losowych o znanym rozkładzie prawdopodobieństwa (będziemy ten fakt oznaczać jako Z_n). Zatem wielkości $K(x^*;Z_n)$, $K(x^A;Z_n)$ oraz wszystkie błędy są zmiennymi losowymi, jako funkcje określone na Z_n. Analiza probabilistyczna dostarcza podstawowych informacji o zachowaniu się zmiennej losowej wybranego błędu, np. $T^A(Z_n)$, jej rozkładzie prawdopodobieństwa, momentach etc. Jednakże najbardziej interesujące charakterystyki dotyczą typu zbieżności $T^A(K;Z_n)$ do wartości stałej m (lub zera) wraz ze wzrostem n oraz szybkości tej zbieżności.

Schemat aproksymacyjny AS generuje rodzinę algorytmów A takich, że A dostarcza dla danego $\epsilon > 0$ rozwiązania x^A spełniającego $S^A(Z) \leqslant 1 + \epsilon$, $\forall Z$. AS jest *wielomianowym schematem aproksymacyjnym* (PTAS), jeśli dla każdego ustalonego ϵ posiada on złożoność obliczeniową wielomianową. Jeżeli dodatkowo ta złożoność jest wielomianem od $1/\epsilon$, to A jest *w pełni wielomianowym schematem aproksymacyjnym*

(FPTAS). W praktyce stopień wielomianu funkcji złożoności obliczeniowej rośnie bardzo szybko przy $\epsilon \to 0$, co powoduje, że schematy te są ciągle słabo konkurencyjne do innych metod.

4. Metody przybliżone

W ostatnich 10 – 20 latach, niezależnie od rozwoju metod teoretycznych, nastąpił burzliwy rozwój metod przybliżonych o dobrych i bardzo dobrych własnościach numerycznych potwierdzonych eksperymentalnie. Metody te są klasyfikowane jako *konstrukcyjne* lub *poprawiające*. Pierwsze z nich są metodami szybkimi, łatwo implementowalnymi, jednakże generują rozwiązania obarczone względnie dużym błędem przybliżenia. Te drugie są wolniejsze, wymagają rozwiązania początkowego poprawianego następnie krokowo, oraz dostarczają rozwiązań o bardzo dobrej i doskonałej jakości. Umożliwiają także elastyczne kształtowanie kompromisu pomiędzy jakością rozwiązania a czasem obliczeń. Teoretyczną ocenę błędu przybliżenia uzyskano do chwili obecnej dla dużej części metod konstrukcyjnych i bardzo nielicznych metod poprawiających, głównie ze względu na znaczny stopień skomplikowania analizy. Dla niektórych metod poprawiających wykazano teoretyczną zbieżność do optimum globalnego, jednakże przy pewnych założeniach w praktyce trudnych do spełnienia. Ostatecznie, przydatność praktyczna poszczególnych podejść i metod jest wypadkową własności i analiz teoretycznych oraz analizy eksperymentalnej.

Algorytm konstrukcyjny (AK) generuje pojedyncze rozwiązanie lub podzbiór rozwiązań, o ustalonej a priori niewielkiej liczności, z którego następnie wybierane jest rozwiązanie najlepsze w sensie wartości funkcji celu $K(x)$. Algorytmy te są oparte głównie na następujących podejściach: (a) reguły priorytetowe, (b) adaptacja rozwiązania otrzymanego dla problemu zrelaksowanego, (c) przybliżenie rozwiązania innym rozwiązaniem otrzymanym z pokrewnych problemów, (d) inne. Algorytmy tej klasy mogą być używane samodzielnie lub w grupach wzajemnie rywalizujących metod. Są tanie obliczeniowo. Powszechnie stosowane także jako generatory rozwiązań początkowych dla algorytmów poprawiających. Kombinując parametrycznie pewną liczbę algorytmów tego typu, w tzw. przestrzeni heurystyk, możemy syntetyzować nowe algorytmy dla nowych problemów.

Reguły priorytetowe (PR) są powszechnie używaną techniką szybkiego wyznaczania rozwiązania, zwłaszcza w systemach produkcyjnych i komputerowych. Są proste, tanie obliczeniowo, mało wrażliwe na zaburzenia danych wejściowych, nadają się dobrze do systemów niedeterministycznych. Rozróżnia się reguły statyczne (wartość priorytetu nie zmienia się w trakcie oczekiwania na obsługę) oraz dynamicznie (wartość ta ulega zmianie w trakcie). W zależności od problemu oraz zastosowanej reguły, dostarczane rozwiazania mogą mieć błąd T^A od 20% (przeciętnie) do 500% (skrajnie).

Obcięte branch-and-bound (CB) otrzymujemy, ograniczając arbitralnie zasoby procesu obliczeniowego używane w klasycznym schemacie B&B, co prowadzi do algorytmu zapewniającego pewien kompromis pomiędzy czasem rozwiązywania a dokładnością przybliżenia. Ograniczenia te mogą dotyczyć m. in. wybranych klas problemów częściowych, liczby węzłów w drzewie, czasu pracy, progu odrzucania. Algorytmy CB zachowują cechy przodków – objawiają eksplozję obliczeń dla problemów o większym rozmiarze.

Poszukiwania lokalne (LS) określają grupę metod, opartych na przeszukiwaniu zbioru rozwiązań $\mathcal{X}$, przez przegladanie wybranych lub wszystkich rozwiązań leżących w pewnym bliskim otoczeniu $\mathcal{N}(x) \subseteq \mathcal{X}$ wybranego rozwiązania x. Analiza $\mathcal{N}(x)$ dostarcza rozwiązań (jednego lub kilku), które stają się źródłami kolejnych lokalnych otoczeń, umożliwiając tym samym powtarzanie procesu poszukiwań. Odpowiednie metody LS łączą wiele zalet: duża szybkość zbieżności, prostota implementacji oraz mały błąd do rozwiązania optymalnego. Wielu badaczy uważa te metody za najbardziej obiecujące dla szczególnie trudnych problemów optymalizacji dyskretnej.

Poszukiwanie zstępujące (DS) jest najstarszą i najprostszą techniką poprawy danego rozwiązania $x^0 \in \mathcal{X}$. Metoda ta jest pokrewna technikom wspinaczki (*hill climbing*) dla problemów maksymalizacyjnych. Elementarny krok tej metody wykonuje, dla danego rozwiązania $x^k \in \mathcal{X}$, przeszukiwanie jego otoczenia lokalnego $\mathcal{N}(x^k) \subseteq \mathcal{X}$ w celu znalezienia rozwiązania $x^{k+1} \in \mathcal{N}(x^k)$ z najmniejszą wartością funkcji celu $K(x)$. Jeżeli tylko w sąsiedztwie $\mathcal{N}(x^k)$ istnieje rozwiązanie, dla którego $K(x^{k+1}) < K(x^k)$, poszukiwanie się powtarza, zaczynając od x^{k+1}, zaś cały proces jest kontynuowany dopóki wartość funkcji celu maleje. Trajektoria poszukiwań $x^0, x^1, \ldots$ zbiega monotonicznie w kierunku ekstremum lokalnego, gdzie metoda DS kończy swoje działanie. Startując wielokrotnie metodę DS z różnych rozwiązań początkowych x^0, możemy eliminować, w pewnym zakresie, jej wrażliwość na lokalne ekstrema. Pewną mutacją metody jest poszukiwanie zstępujące z *pełzaniem* (descending search with drift, DSD), które akceptuje na trajektorii także nieodwiedzone rozwiązania z tą samą wartością kryterium, tzn. $K(x^{k+1}) = K(x^k)$.

Poszukiwanie losowe (RS) generuje trajektorię poszukiwań $x^0, x^1, \ldots$ tak, że kolejne rozwiązanie x^{k+1} jest wybierane losowo w $\mathcal{N}(x^k)$ (często $\mathcal{N}(x^k) = \mathcal{X}$ dla każdego k). RS zwraca jako wynik rozwiązanie najlepsze z trajektorii. Chociaż metoda ta nie jest wrażliwa na ekstrema lokalne, jej zbieżność do dobrego rozwiązania jest ogólnie słaba. Istniejące modyfikacje tej metody uwzględniają zmiany rozkładu prawdopodobieństwa przy wyborze sąsiada w celu skierowania poszukiwań w bardziej obiecujący obszar zbioru rozwiązań $\mathcal{X}$.

Metody ulosowione (RM) nie określają konkretnej metody, lecz jedynie ogólny schemat postępowania polegający na zastępowaniu w znanych algorytmach przybliżonych pewnych parametrów deterministycznych (np. priorytety wyboru) ich odpowiednikami losowymi. Dla otrzymanych w ten sposób algorytmów zrandomizowanych można uzyskać teoretyczne, asymptotyczne oceny błędu przybliżenia.

Poszukiwanie snopowe (BS) "oświetla" wiązką (snopem światła) przestrzeń rozwiązań. Każdy "oświetlony" punkt przestrzeni staje się źródłem ukierunkowanej, ograniczonej wiązki świetlnej zawierającej pewną liczbę rozwiązań potomnych, zwykle najbardziej obiecujących z punktu widzenia procesu poszukiwań. Ocena przydatności rozwiazań może być prowadzona zarówno w oparciu o wartość funkcji celu, jak i w oparciu o inne funkcje definiowane przez użytkownika. Często BS występuje w odniesieniu do CB, gdzie powoduje ograniczenie liczby najbardziej obiecujących bezpośrednich następników węzła w drzewie rozwiązań podlegających rozwinięciu, wyrażone przez ustaloną a priori *szerokość wiązki*, prowadząc do metody *filtrowanego poszukiwania wiązką* (FBS).

Sterowane poszukiwanie lokalne (GLS) jest problemowo zorientowane i kieruje trajektorię poszukiwań w najbardziej obiecujące regiony zbioru $\mathcal{X}$, wykorzystując

deterministyczny scenariusz formułowany indywidualnie dla problemu. Regiony są wybierane a priori w oparciu o szczególne własności problemu. Znane warianty metody stosują techniki BS połączone z LS, co pociąga za sobą negatywne skutki eksplozji obliczeń.

Poszukiwanie genetyczne (GS) odwołuje się do Natury i teorii ewolucji Darwina, zakładając, że niejawnym celem ewolucji jest optymalizacja dopasowania osobników do środowiska. GS używa podzbioru rozproszonych rozwiązań $\mathcal{W} \subset \mathcal{X}$ zwanego *populacją* do prowadzenia poszukiwań równocześnie w wielu obszarach zbioru rozwiązań $\mathcal{X}$. Każde rozwiązanie $x \in \mathcal{W}$ zwane *osobnikiem* jest kodowane przez zbiór jego atrybutów zapisanych w materiale genetycznym (chromosomy, geny). Populacja jest kontrolowana zasadniczo przez cyklicznie następujące po sobie procesy *reprodukcja*, *krzyżowanie* i *mutacja* oraz *przeżycie* lub *selekcja*. W fazie reprodukcji osobniki są wpierw powielane proporcjonalnie do ich miary *przystosowania do środowiska*. Ten proces gwarantuje, że osobniki lepiej przystosowane będą miały więcej potomków w następnym pokoleniu. Osobniki wybrane z rozszerzonej populacji tworzą *pulę rodzicielską*. Z tej puli kojarzymy pary rodziców, którzy następnie dostarczą *odnowione pokolenie*. Każdy osobnik nowego pokolenia jest nowym rozwiązaniem x' otrzymanym z dwóch rozwiązań rodzicielskich x oraz y przez zastosowanie *operatora krzyżowania genetycznego*. Mutacja jest ubezpieczeniem na wypadek utraty istotnych atrybutów rozwiązań oraz powolnym mechanizmem wprowadzania atrybutów innowacyjnych. Powoduje sporadyczne, losowe (z małym prawdopodobieństwem) zmiany w materiale genetycznym. W fazie przeżycia wybierane są osobniki (spośród starego i nowego pokolenia), które wejdą w skład nowej populacji.

GS posiada wiele punktów swobody, których ciągle jeszcze właściwa kompozycja jest kluczem do sukcesu. Mimo wielu badań, pewne niedostatki algorytmów GS są wciąż obserwowane w praktyce. Wyrażają się one przedwczesną zbieżnością do lokalnego ekstremum lub słabą zbieżnością do rozwiązań bliskich optymalnemu. Podczas gdy GS zachowuje się zadowalająco dobrze dla małych przykładów, błędy przybliżenia dla przykładów o dużym rozmiarze mogą być znaczne. Wykazano, że odpowiedzialność za przedwczesną zbieżność do lokalnych ekstremów ponosi niewłaściwe sterowanie dynamiką rozwoju populacji. Aby poprawić dopasowanie populacji do środowiska, do reprodukcji w każdym pokoleniu są preferowane najlepsze osobniki, co pociąga stałe zmniejszanie rozproszenia genetycznego populacji. To z kolei zmniejsza możliwości znalezienia istotnie innego lepszego rozwiązania przez krzyżówkę genetyczną i wstrzymuje postęp do czasu, gdy dopiero mutacja (po ogromnej liczbie pokoleń) wprowadzi potrzebną zmianę w materiale genetycznym. Dlatego też celem nadrzędnym algorytmu GS jest utrzymać dopasowanie maksymalnie skrajne bez utraty rozproszenia genetycznego populacji.

Zaproponowano techniki kontrolowania zbieżności algorytmu GS przez *strategie kojarzenia rodziców*, *wprowadzenie struktur do populacji* oraz *wzorce zachowań społecznych*. Wprowadzono *funkcję współdzielenia* w dopasowaniu rodziców w celu zapobieżenia zbyt bliskim podobieństwom genotypowym. Inne bezpośrenie podejścia *zapobiegania kazirodztwu* posługują się odległością Hamminga przy ocenie podobieństwa genotypowego. Tworzenie struktur w ramach populacji może być otrzymane przez podział populacji na *wyspy* (model migracyjny) lub przez wprowadzenie *nakładających się sąsiedztw* pokrywających całą populację (model dyfuzyjny). Oba modele zakładają

ograniczenie bezpośredniej wymiany danych pomiędzy osobnikami z różnych obszarów zamieszkałych przez subpopulacje, przez co wprowadzają naturalne *nisze ekologiczne* umożliwiające zachowania pożądanego rozproszenia genetycznego. Z kolei podejście oparte na zachowaniach socjalnych przypisuje każdemu osobnikowi w populacji odpowiednią postawę społeczną, np. zaspokojony, zadowolony lub rozczarowany. Wzorzec postawy wynika m.in. z wartości funkcji celu wyznaczonej dla osobnika. Ocena aktualnego zachowania ma wpływ na postawę w przyszłości, zatem osobniki mogą różnie reagować w tej samej sytuacji środowiskowej. Każda postawa społeczna wpływa na gotowość osobnika do wchodzenia w krzyżówki, mutacje lub na *uśpienie, klonowanie* (przetrwanie do kolejnego pokolenia).

Ewolucja różnicowa (DE) jest podklasą metody GS. Demokratyzm tworzenia potomków i mutacji w GS został zastąpiony w DE realizacją *ukierunkowanych* zmian sondujących przestrzeń rozwiązań. DE startuje od losowej populacji rozwiązań. W każdym kroku wykorzystywane są mechanizmy mutacji i krzyżowania, jednak przebiegają one odmiennie niż w GS. Każdy osobnik populacji jest rodzicem dokładnie jednego nowego *rozwiązania próbnego* otrzymanego na bazie tego osobnika oraz dwóch rozwiązań *kierunkujących* wybranych losowo z populacji. Generowanie potomka ma cechy kombinacji liniowej rozwiązań z pewnymi elementami losowości. Oddzielny mechanizm losowości zapobiega generowaniu potomka przez proste powielenie rodzica. Ważna rola przypada mutacji, która dzięki specyficznej strategii jest samoadaptacyjna oraz celowa co do kierunku, skali i zakresu. Jeśli rozwiązanie próbne jest lepsze w sensie wartości funkcji celu, to zastępuje rodzica w populacji, inaczej jest odrzucane. Odnowienie populacji jest realizowane cyklicznie do chwili osiągnięcia zadanej liczby generacji lub stwierdzenia wystąpienia *stagnacji* poszukiwań. Metoda posiada wiele specyficznych operatorów oraz wiele punktów swobody (parametrów) dobieranych eksperymentalnie.

Poszukiwanie memetyczne (MS) odwołuje się do zapomnianej teorii ewolucji Lamarcka, w której przekazywanie cech wrodzonych oraz nabytych pomiędzy pokoleniami odbywa się z użyciem *memów* stanowiących informacyjną analogię genów występujących w GS. Zasadnicza różnica w stosunku do GS polega na *uczeniu* się osobników (zwykle z użyciem DS lub LS), zmieniających ich cechy nabyte, przed procesem krzyżowania. Otrzymywane wyniki są zdecydowanie lepsze niż dla GS, jednak kosztem czasu działania.

Podejście immunologiczne (AIS) naśladuje w swojej budowie i działaniu system naturalny. W Naturze system immunologiczny jest złożonym, rozproszonym, samoregulującym, samoadaptacyjnym systemem wykorzystującym uczenie, pamięć oraz odpowiednie wyszukiwanie informacji w celu obrony organizmu przed patogenami i toksynami powodującymi zaburzenie jego funkcjonowania. Jego podstawowym celem jest rozpoznanie i klasyfkacja komórek na własne (komórki gospodarza) oraz obce (inwazyjne). Komórki obce są następnie dalej klasyfikowane w celu uruchomienia odpowiedniego mechanizmu obronnego *wrodzonego* lub *nabytego*. Głównym typem komórek obronnych są *limfocyty B* oraz *T* posiadające własność rozpoznawania gospodarza, specyficzność, pamięć i rozproszenie. W systemie występują także inne komórki, tzw. fagocytowe (neutrofile, eozynofile, bazofile, monocyty) pełniące funkcje pomocnicze. Pojawieniu się komórki inwazyjnej towarzyszy obecność obcych białek (antygeny) wywołujących reakcję systemu immunologicznego, której pierwotnym celem jest wytworzenie specyficznych przeciwciał blokujących receptory antygenu, prowadzące w dalszej kolejności do zniszczenia intruza.

AIS buduje analogie do systemu naturalnego, przy czym tylko niektóre funkcje, elementy składowe oraz mechanizmy mogą podlegać przeniesieniu. *Antygen* (białko inwazyjne) reprezentuje chwilowe wymagania narzucone na rozwiązanie, np. na zakres niektórych lub wszystkich składowych wektora x, niewynikające z danych problemu. Repertuar możliwych antygenów jest bardzo duży (czasami nieskończony), zaś aktualna konfiguracja pojawiających się antygenów jest a priori nieznana. *Przeciwciało* (białko blokujące antygen) jest listą instrukcji (algorytmem) na utworzenie rozwiązania spełniającego wymagania określone antygenem. Repertuar dostępnych przeciwciał jest zwykle niewielki, lecz istnieje mechanizm ich agregacji i rekombinacji w celu uzyskiwania nowych przeciwciał o odmiennych właściwościach. Wzorce przeciwciał, kolekcjonowane w bibliotece, tworzą pamięć systemu. *Dopasowaniem* nazywamy dobranie przeciwciała do antygenu. *Wynik dopasowania* jest idealny, jeśli przeciwciało pozwala na wygenerowanie rozwiązania spełniającego wymagania antygenu. Inaczej, wynik dopasowania jest pewną miarą odchylenia otrzymanego rozwiązania od narzuconych wymagań. Złe dopasowanie zmusza system do poszukiwania nowych typów przeciwciał, zwykle w oparciu o ewolucję (GS, DE).

Ścieżki przejściowe. Dla wielu problemów dyskretnych stwierdzono występowanie ekstremów lokalnych skupionych w obszarze pojedynczej doliny w przestrzeni rozwiązań. Detekcja, czy rozważany problem posiada tę własność, lokalizacja doliny, jak również badanie topologii przestrzeni są możliwe przez śledzenie *ścieżek przechodzących* (PR) przez tę przestrzeń i łączących wybrane rozwiązania. W zależności od charakteru przestrzeni ścieżki mogą być one generowane przez kombinację (liniową) rozwiązań lub kierunkowane za pomocą specyficznej *miary* odległości pomiędzy rozwiązaniami.

Celowe ścieżki śledzące (GOTP) określają ścieżki przejściowe z ustalonym punktem docelowym, zarówno deterministyczne, jak i losowe, wykonywane w technice LS. Podane referencje odnoszą się do operatora genetycznego MSFX (multi-step fusion) wykonującego wielokrokową fuzję chromosomów rodziców w GS, który realizuje GOTP z elementami losowości podobnymi do SA. Jego użycie w problemach szeregowania przyniosło wyraźne korzyści dla jakości generowanych rozwiązań, kosztem czasu przebiegu.

Poszukiwania biochemiczne rozpoczyna się od zakodowania danych za pomocą łańcuchów DNA (o długości 10-30 nukleotydów) z lepkimi końcami, które "pasują" do siebie w ściśle określony sposób. Rozwiązaniem jest sekwencja połączonych łańcuchów, przy czym za rozwiązanie optymalne (dopuszczalne) jest przyjmowany łańcuch o określonej długości złożony z łańcuchów elementarnych. Poszukiwanie polega na losowym sklejaniu się łańcuchów elementarnych i zachodzi w warunkach laboratoryjnych (probówka) w medium zawierającym odpowiednio dużą liczbę powielonych łańcuchów elementarnych. Ze względu na submikroskopową skalę pojedynczego zjawiska sklejania, zachodząca reakcja jest równoważna znacznej liczbie (np. 10^{10}) równoległych procesów poszukiwań losowych. Stwierdzenie, czy zostało znalezione wymagane rozwiązanie, jest wykonywane testem biochemicznym wykrywającym czasteczki DNA o określonych wielkościach.

Symulowane wyżarzanie (SA) naśladuje proces odprężania ciała polegającego na powolnym studzeniu ciała ferromagnetycznego lub antyferromagnetycznego w celu eliminacji naprężeń wewnętrznych. Wykorzystuje się tu analogię pomiędzy wiel-

kością energii ciała a funkcją celu problemu optymalizacyjnego oraz pomiędzy konfiguracją wewnętrzną cząsteczek ciała a rozwiązaniem problemu optymalizacyjnego. SA generuje trajektorię poszukiwań $x^0, x^1, \ldots$ przechodzącą przez zbiór $\mathcal{X}$, gdzie $x^{k+1} \in \mathcal{N}(x^k)$ jest wybierane w następujący sposób. Wpierw *rozwiązanie zaburzone* $x' \in \mathcal{N}(x^k)$ jest wybierane w sposób uporządkowany lub losowo z równomiernym rozkładem prawdopodobieństwa. Jeśli $K(x') \leqslant K(x^k)$, to x' jest akceptowane natychmiast, tzn. $x^{k+1} := x'$. Inaczej x' jest akceptowane jako nowe rozwiązanie z prawdopodobieństwem $\min\{1, e^{-\Delta/T_i}\}$, gdzie $\Delta = K(x') - K(x^k)$, zaś T_i jest parametrem zwanym *temperaturą* w iteracji i. Jeśli x' nie zostało zaakceptowane (ani przez pierwszy, ani przez drugi warunek), podstawiamy $x^{k+1} := x^k$. Temperatura jest zmieniana w czasie iteracji według *schematu studzenia*. W każdej ustalonej temperaturze wykonywana jest pewna liczba, oznaczmy ją m, iteracji. Powszechnie stosowanych jest kilka schematów studzenia: geometryczny $T_{i+1} = \lambda_i T_i$, logarytmiczny, $T_{i+1} = T_i\,(1 + \lambda_i T_i)$, harmoniczny $T_i = \Gamma/\log(i+2)$, $i = 0, \ldots, N-1$, gdzie N jest całkowitą liczbą iteracji, λ_i jest pewnym parametrem liczbowym, zaś T_0 jest temperaturą początkową. Powinno zachodzić $T_N < T_0$ oraz T_N jest bliskie zeru.

Metoda SA posiada wiele parametrów, które należy dobrać eksperymentalnie. Parametr λ_i jest często wybierany jako stały. Zakładając znane (estymowane) wartości T_0, T_N i N, można wyznaczyć wartość parametru λ_i, wg zależności $\lambda_i = (T_0 - T_N)\,(NT_0T_N)$ dla schematu logarytmicznego. Dostarczono także wiele cennych rad dotyczących automatycznego wyboru x^0, T_0, λ_i, m oraz kryterium stopu. Schemat studzenia ma silny wpływ na zachowanie się metody. Jeżeli studzenie jest zbyt szybkie, SA zachowuje się podobnie do metody DS i zwykle szybko kończy swoje działanie w lokalnym ekstremum słabej jakości. Jeżeli studzenie jest zbyt powolne, czas przebiegu algorytmu staje się nieakceptowalnie długi. W praktyce, właściwy dobór wszystkich parametrów sterujących metody wymaga wykonania znacznej liczby testów komputerowych. Przy spełnieniu pewnych założeń metoda SA zbiega do rozwiązania globalnie optymalnego z prawdopodobieństwem jeden. Szczegółowa analiza została także przeprowadzona dla asymptotycznej zbieżności oraz szybkości zbieżności. Ponieważ niektóre z tych założeń są w praktyce niespełnialne, otrzymane rezultaty mają bardziej teoretyczne niż praktyczne znaczenie.

Symulowane przeskakiwanie (SJ) naśladuje proces odprężania ciała polegającego na powolnym studzeniu, jak i nagrzewaniu ciała typu spin-glass (wymieszany materiał ferromagnetyczny i antyferromagnetyczny) w celu przełamania barier energetycznych istniejących pomiędzy domenami. W praktyce SJ wprowadza do SA dodatkowo schemat podgrzewania, stosowany gdy zaburzone rozwiązanie x' nie zostało zaakceptowane, tzn. gdy musimy podstawić $x^{k+1} = x^k$. SJ ma więcej niż SA elementów wybieranych eksperymentalnie w celu zapewnienia odpowiednio dobrych własności numerycznych algorytmu. Szybkość zbieżności silnie zależy od parametrów sterujących.

Poszukiwanie z zakazami (TS) jest metodą powielającą naturalny proces poszukiwania rozwiązania wykonywany przez człowieka. Podstawowa wersja TS rozpoczyna działanie od pewnego rozwiązania początkowego $x^0 \in \mathcal{X}$. Elementarny krok tej metody wykonuje, dla podanego rozwiązania x^k, przeszukiwanie całego sąsiedztwa $\mathcal{N}(x^k)$ rozwiązania x^k w celu znalezienia $x^{k+1} \in \mathcal{N}(x^k)$ z najmniejszą wartością funkcji celu $K(x)$ lub wartością innej niekosztownie obliczanej funkcji oceny. Następnie proces poszukiwania jest powtarzany od najlepszego znalezionego rozwiązania, dostarczając

trajektorii poszukiwań $x^0, x^1, \ldots$. TS zwraca najlepsze rozwiązanie z trajektorii. Aby zapobiec cyklicznemu powtarzaniu się rozwiązań na trajektorii, zatrzymaniu w ekstremum lokalnym, oraz aby prowadzić trajektorię poszukiwań w obiecujące obszary zbioru rozwiązań, wprowadzono pamięć historii poszukiwań. Wśród wielu klas pamięci używanych dla TS najczęściej wymieniana jest pamięć krótkoterminowa zwana listą zabronień. Lista ta przechowuje przez ograniczony czas najświeższe rozwiązania z trajektorii poszukiwań, wybrane atrybuty tych rozwiązań, przejścia (ruchy) prowadzące do tych rozwiązań lub atrybuty ruchów, traktując je wszystkie jako formę zabronienia dla ruchów wykonywanych w przyszłości. Wykonanie ruchu posiadającego zabronione atrybuty jest zabronione (odpowiednie rozwiązanie musi zostać pominięte w poszukiwaniach). Zabronienie to może być anulowane, jeśli funkcja aspiracji uzna ten ruch za wystarczająco korzystny. Poszukiwanie zatrzymuje się, jeśli została wykonana pewna liczba iteracji bez poprawy wartości funkcji celu, wykonano założoną z góry całkowitą liczbę iteracji, przekroczono czas obliczeń, znaleziono rozwiązanie satysfakcjonujące użytkownika. Możliwość znalezienia rozwiązania optymalnego (bez gwarancji znalezienia go) zapewnia zachodzenie własności łączności dowodzonej dla oddzielnych problemów i sąsiedztw. Własność ta określa warunki, przy których dla dowolnego rozwiązania początkowego x^0 istnieje trajektoria poszukiwań $x^0, x^1, \ldots, x^\tau$ taka, że $x^{k+1} \in \mathcal{N}(x^k)$, prowadząca do rozwiązania optymalnego $x^\tau = x^*$.

Poszukiwanie z pamięcią adaptacyjną (AMS) jest stosowane do określenia bardziej zaawansowanych schematów TS, które wychodzą poza zbiór składników i narzędzi opisanych w rozdziale poprzednim. Zostało sprawdzone eksperymentalnie, że procesy poszukiwania powinny posiadać swoistą równowagę pomiędzy uszczegółowieniem (staranne przeszukiwanie niewielkich obszarów zbioru $\mathcal{X}$) a rozproszeniem (przeszukiwanie rozproszonych obszarów zbioru $\mathcal{X}$). Tak więc AMS wprowadza szerszy schemat, dodając bardziej wyrafinowane techniki pamięciowe (np. *pamięć średnio- i długoterminowa*, *pamięć częstotliwościowa*, *pamięć atrybutowa*, *haszowa lista zabronień*), nowe elementy strategiczne (np. *strategia listy kandydatów*, *oscylacje strategiczne*, *ścieżki przejściowe*) oraz zachowanie adaptacyjne z uczeniem (np. *zabronienia reagujące*).

Poszukiwanie mrówkowe (AS) odwołuje się do "inteligentnego" zachowania kolonii "nieinteligentnych", lecz współpracujących osobników, mającej analogię do społeczności mrówek. Poszukiwania są rozproszone pomiędzy osobniki z bardzo prostymi umiejętnościami, które przypominają w swych zachowaniach prawdziwe mrówki. Metoda ta jest inspirowana odkryciem znaczenia *ścieżki feromonowej* w poszukiwaniu przez mrówkę najkrótszej ścieżki prowadzącej z mrowiska do źródła pożywienia i z powrotem. Podczas gdy izolowana mrówka porusza się losowo, trafiając na pozostawioną na powierzchni gruntu ścieżkę feromonową, porusza się wzdłuż tej ścieżki z prawdopodobieństwem proporcjonalnym do intensywności śladu feromonowego. Każda mrówka w czasie ruchu znaczy ścieżkę swoim własnym feromonem, zwiększając intensywność śladu. Równocześnie feromon paruje ze ścieżek, aby zapobiec eksplozji procesu. Zakładając pewne dopuszczalne ścieżki dla mrówek (zależnie od rozważanego problemu), pewien model zachowania mrówki (mrówka ma pewną pamięć krótkoterminową, może używać wzroku w ograniczonym zakresie, posiada czujnik śladu ferromonowego), początkowy rozkład mrówek na ograniczonym obszarze oraz początkową intensywność ścieżek feromonowych, rozpoczyna się proces symulacji aktywności kolonii mrówek z dyskretnym czasem.

Algorytm mrówkowy dla problemu optymalizacji dyskretnej traktuje rozwiązanie dopuszczalne problemu jako analogiczne do trasy mrówki od mrowiska do źródła pożywienia (dla niektórych zagadnień także z powrotem), zaś wartość funkcji celu jako analogiczną do długości tej trasy. Każda mrówka, przechodząc po trasie, generuje pojedyncze rozwiązanie, przy czym mrówki przechodzące po trasach krótszych szybciej osiągają punkt docelowy. To z kolei implikuje częstsze krążenie mrówek na trasach najkrótszych, nasycając te trasy intensywniej feromonem. Spotykane są dwa odmienne schematy symulacji: (a) przejście mrówek na trasie mrowisko-pożywienie odbywa się w jednym cyklu, zaś mrówki są "premiowane" ilością ferromonu odwrotnie proporcjonalną do długości wykonanej trasy, (b) przejście mrówek na trasie mrowisko–pożywienie jest procesem dynamicznym, prędkość mrówki oraz ilość pozostawianego feromonu są stałe w czasie.

Poszukiwanie rozproszone (SS) operuje na zbiorze rozwiązań rozproszonych zwanych *rozwiązaniami referencyjnymi*, na który składają się rozwiązania elitarne otrzymane w dotychczasowym procesie obliczeniowym. SS generuje systematycznie nowe rozwiązania jako *liniową kombinację* rozwiązań referencyjnych. W odróżnieniu od demokratycznego GS, metoda SS skupia się na wyborze dobrych rozwiązań jako podstawie do tworzenia nowych kombinacji oraz na odmiennych metodach ich kombinacji, bez utraty zdolności produkowania rozwiązań rozproszonych.

Poszukiwanie ścieżkowe (PS) generuje pewną liczbę trajektorii poszukiwań zwanych *ścieżkami poszukiwań* z ustalonego rozwiązania początkowego. Każda trajektoria $x^0, \ldots, x^k$ zawiera rozwiązania takie, że $x^{k+1} = \min_{x \in \mathcal{N}(x^k)} F(x)$, gdzie $\mathcal{N}(x^k)$ jest sąsiedztwem, zaś $F(x)$ pewną *funkcją oceny*, np. $F(x) = K(x)$. Trajektoria jest kończona, jeśli dana a priori liczba ostatnio generowanych rozwiązań z tej trajektorii nie poprawia najlepszego znanego rozwiązania. Po zakończeniu nieperspektywicznej trajektorii generowana jest nowa trajektoria, począwszy od wybranego (nieodwiedzonego) rozwiązania wśród rozwiązań leżących na trajektoriach już wygenerowanych.

Spełnienie ograniczeń (CS) polega na zastąpieniu problemu optymalizacyjnego decyzyjnym problemem spełnialności, tzn. na znalezieniu rozwiązania x i wartości funkcji $K(x)$ spełniającego zmodyfikowany zestaw ograniczeń. Metoda ta posługuje się *zmiennymi decyzyjnymi* V (np. $(x, K(x))$ w naszym przypadku), *dziedziną* $\mathcal{D}$ tych zmiennych oraz *ograniczeniami* $\mathcal{C}$ nałożonymi na dwie lub więcej zmiennych z V. Podstawowy krok tej metody polega na konstrukcji w uporządkowany sposób rozwiązań przez rozszerzanie metodą *pierwszy w głąb* bieżącego rozwiązania częściowego. Każde takie rozszerzenie definiuje nowy problem CS posiadający zmodyfikowane $\mathcal{D}$ i $\mathcal{C}$. Nowa postać $\mathcal{D}$ jest otrzymywana przez *propagację ograniczeń* z $\mathcal{C}$. Jeżeli otrzymany problem CS jest niezgodny (niespełnialny), poszukiwania są wycofywane z tego stanu, zaś proces jest kontynuowany z innego zgodnego stanu. Algorytmy CS różnią się typem i poziomem wymuszania zgodności, mechanizmami powrotów ze stanów niezgodnych (np. *skoki powrotne*, *powroty kierunkowo-zależne*, *powroty dynamiczne*) i heurystykami wykorzystywanymi do tworzenia rozwiązań rozszerzonych. Metoda CS posiada pewne podobieństwa do metod B&B i CB, zatem przejawia zarówno pozytywne, jak i negatywne cechy tych dwóch podejść.

Zlepianie (chunking) (C) nie jest autonomiczną metodą poszukiwania rozwiązania, lecz narzędziem wspierającym skuteczność innych metod szukania. Zlepianie oznacza grupowanie podstawowych informacji w celu utworzenia zredukowanej liczby jed-

nostek informacyjnych wyższego poziomu. Jest uważane za krytyczny aspekt ludzkiej inteligencji. Człowiek, któremu przychodzi rozwiązać trudny problem, często rozpoczyna od zgrupowania własności postrzeganych jako pokrewne i przeniesienia ich w takiej formie do procesu rozwiązywania. Daje to możliwość organizacji danych oraz metody rozwiązywania w sposób hierarchiczny, pozwalający zdekomponować problem na podproblemy i tak go rozwiązać. Kiedy problem nie może być zdekomponowany lub metoda dekompozycji jest nieznana, powszechną strategią jest grupowanie atrybutów rozwiązań w celu redukcji wielkości zbioru rozwiązań, co może doprowadzić do odkrycia i wykorzystania zależności wyższego poziomu. Formy agregatów są specyficzne dla każdego rozwiązywanego problemu, mogą być predefiniowane przez człowieka lub wykryte w czasie procesu poszukiwań.

Podejście geometryczne (GES) odwołuje się do geometrycznej interpretacji rozwiązania i pewnych ogólnych twierdzeń z probabilistyki i algebry. Powszechnie stosowany jest następujący schemat postępowania: (1) znajdź oczywiste rozwiązanie x niekoniecznie dopuszczalne, (2) zaburz rozwiązanie x losowo, otrzymując x', (3) transformuj x' na rozwiązanie dopuszczalne $x'' \in \mathcal{X}$, używając specjalnego rozszerzającego podejścia. Chociaż GES dostarcza algorytmów o słabej ocenie eksperymentalnej, to pozwala na przeprowadzenie teoretycznej oceny jakości rozwiązań.

Poszukiwanie stadne (swarm intelligence) (SI) wykorzystuje skupiony zbiór rozwiązań $\mathcal{W}$ (stado) do przeszukiwania przestrzeni $\mathcal{X}$. Każdy osobnik stada porusza się po trajektorii w $\mathcal{X}$ wyznaczonej przy uwzględnieniu aktualnego położenia najbliższych mu osobników stada oraz przewodnika.

Sieci neuronowe (NN) są problemowo zorientowane, choć ich zastosowanie do rozwiązywania problemów dyskretnych jest wciąż niedostateczne. Większość algorytmów neuronowych używa deterministycznego, analogowego modelu sieci Hopfielda i Tanka. Poszczególne składowe rozwiązania mają swoje analogie do napięć w sieci, ograniczenia są przekładane na interakcje neuronów, zaś funkcja celu ma analogie do energii sieci. Do minimalizacji wykorzystuje się własność sieci polegającą na samorzutnym zbieganiu od pewnego stanu początkowego do stanu minimalnego energetycznie. Sieci te są często krytykowane za braki, takie jak przedwczesna zbieżność, problemy ze skalowaniem, niemożność wyznaczenia nawet rozwiązania dopuszczalnego.

Metody równoległe. Szereg z wymienionych podejść można zaadaptować do prowadzenia poszukiwań przy wykorzystaniu równoległych środowisk obliczeniowych, np. równoległe warianty SA, SJ, GA, TS, AMS. W wielu przypadkach taka implementacja jest całkiem nietrywialna. Oprócz naturalnego skrócenia czasu przebiegu algorytmu, obserwowany jest fenomen *ponadliniowego* przyśpieszenia obliczeń.

5. Podsumowanie

Przegląd nie wyczerpuje całkowicie potencjalnych kierunków rozwoju (pominięto m.in. algorytmy kulturowe, muzyczne, hybrydowe) oraz wszystkich modyfikacji wprowadzanych do istniejących podejść. Jakość metod łączy się z krajobrazem przestrzeni rozwiązań, szorstkością powierzchni, dużą doliną, rozkładem rozwiązań w $\mathcal{X}$, równowagą w rozpraszaniu i skupianiu poszukiwań. W chwili obecnej za najbardziej obiecujące uważa się: SA, SJ, GA, MS – dla problemów bez żadnych własności szczególnych (przy czym SA, SJ dla problemów o wysokim koszcie oceny pojedynczego

rozwiązania) oraz TS, AMS – dla problemów posiadających własności szczególne. Dla problemów o dużym rozmiarze polecane są metody równoległe, możliwe do implementacji już na sprzęcie osobistym.

BIBLIOGRAFIA

1. Smutnicki C.: Algorytmy szeregowania, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002.
2. Bożejko W.: Równoległe algorytmy szeregowania zadań produkcyjnych (praca doktorska), Raport PRE 29/2003, PWr, 2003.
3. Corne D., Dorigo M., Glover F.: New ideas in optimization, McGraw Hill, Cambridge, 1999.
4. Wierzchoń S.T.: Sztuczne systemy immunologiczne. Teoria i zastosowania. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001.

Recenzent: Prof. dr hab. inż. Andrzej Świerniak

Abstract

There is presented the review of solution methods for discrete optimization problems. Special attention has been paid to new, promising approaches leading to construction of approximate algorithms.