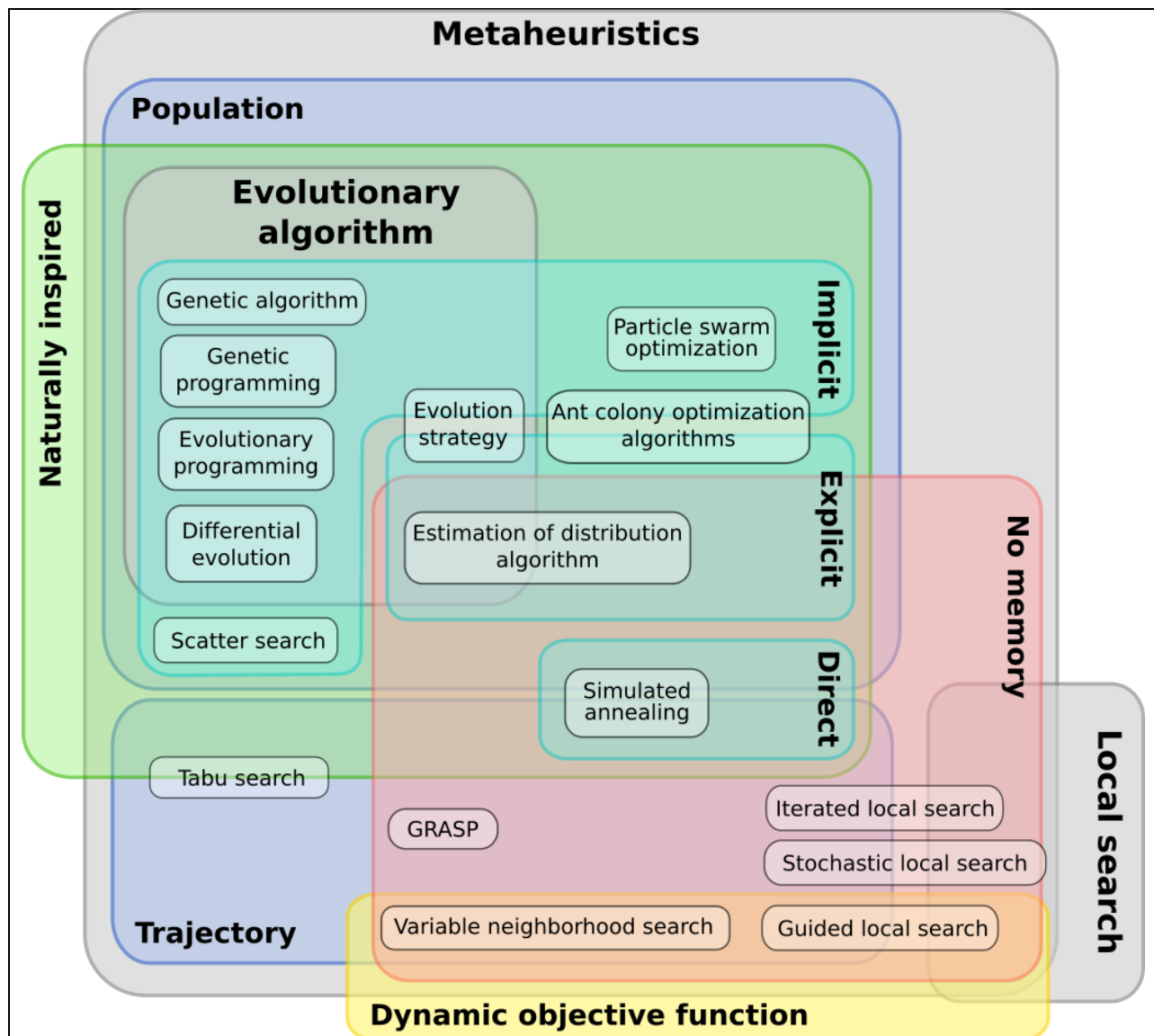


Funkcja celu	Odpowiada na pytanie: W jakim stopniu nasz cel jest zrealizowany / niezrealizowany?
Rozwiązywanie zadanie przez przeszukiwanie przestrzeni rozwiązań / dziedziny	Znalezienie sekwencji przejść między punktami przestrzeni przeszukiwań która doprowadzi do rozwiązania. Korzystamy ze zdefiniowanego sąsiedztwa.
Algorytm	Sekwencja instrukcji służąca do osiągnięcia określonego celu.
Funkcja heurystyczna ang. HEURISTIC FUNCTION	Funkcja służąca do szacowania / wskazywanie korzystnego kierunku prowadzenia obliczeń.
Heurystyka	Konkretna metoda prowadzenia obliczeń która nie daje gwarancji znalezienia najlepszego rozwiązania.
Metaheurystyka	Ogólna metoda prowadzenia obliczeń która pozwala na znajdowanie rozwiązań bez gwarancji znalezienia najlepszego rozwiązania.
Metaheurystyka a heurystyka	Metaheurystyka z ustalonymi parametrami staje się heurystyką.
Złożoność problemu	Ilość (pamięć / procesor) wymagana do wykonania zadania.
Asymptotyczne tempo wzrostu	Miara złożoności, która określa jak zachowuje się algorytm dla danego problemu. Określa ona funkcję, według której rośnie złożoność w zależności od rozmiaru danych.
Notacja asymptotyczna	Zapis określający asymptotyczne tempo wzrostu. Najczęściej stosuje się $O$ , $o$ , $\Omega$ oraz $\Theta$
Klasa złożoności	Zbiór problemów o tej samej złożoności asymptotycznej.
Klasa P, deterministyczno-wielomianowy ang. DETERMINISTIC POLYNOMIAL	Problem, dla którego można znaleźć rozwiązanie w czasie wielomianowym.
Problem NP, niedeterministycznie-wielomianowy ang. NONDETERMINISTIC POLYNOMIAL	Problem, dla którego rozwiązanie można zweryfikować w czasie wielomianowym.
Trudność problemu	Problem jest łatwy, jeśli każda ścieżka przeszukiwania dziedziny problemu wzdłuż nierosnącej (niemalejącej) funkcji celu

	prowadzi do rozwiązania. Długość zależy wielomianowo od ilości danych.
Optymalizacja - ścisła definicja	Poszukiwanie najlepszego rozwiązania.
Optymalizacja - podejście praktyczne	Szukanie odpowiednio dobrego rozwiązania.
Funkcja celu ang. OBJECTIVE FUNCTION	Optymalizowana funkcja. Określa co chcemy osiągnąć.
Funkcja kosztu ang. COST FUNCTION	Ocenia jakość rozwiązania. Funkcja kosztu $\approx$ Funkcja celu + kary za naruszenia ograniczeń
Rozwiązanie analityczne	Na podstawie wzoru wyznaczamy maksimum lub minimum.
Rozwiązanie numeryczne	Za pomocą próbkowania funkcji optymalizowanej.
Zdania ciągłe	Przestrzeń przeszukiwań to iloczyn kartezjański liczb rzeczywistych. Wyróżniamy zadania: <ul style="list-style-type: none"> <li>- Wypukłe (dziedzina i przeciwdziedzina są wypukłe)</li> <li>- Optymalizacji globalnej (niewypukła funkcja celu lub dziedzina funkcji)</li> </ul>
Zadanie dyskretne	Dziedzina należy do zbioru dyskretnego. Jeśli zmienne mają wartości binarne, wtedy jest to optymalizacja kombinatoryczna.
Zadania mieszane	Zmienne dziedziny mają wartości dyskretne i rzeczywiste.
Różniczkowalność	Istnienie pochodnej.
Wypukły obszar dopuszczalny	Dziedzina funkcji tworzy obszar, w którym każdy odcinek którego końce są w obszarze dozwolonym zawiera tylko punkty z dziedziny. Tak ograniczoną funkcję można przekształcić w funkcję celu z ograniczeniem kostkowym.
Niewypukły lub niespójny obszar dziedziny	W takim wypadku może być wiele minimów / maksimów na granicy obszaru dopuszczalnego.
Podział metod optymalizacyjnych	<ul style="list-style-type: none"> <li>- Dokładne – przybliżone – losowe</li> <li>- Lokalne – globalne</li> <li>- Populacyjne – jedno rozwiązanie</li> <li>- Równoległe</li> <li>- Inspirowane naturą</li> <li>- Hybrydowe – memetyczne</li> </ul>



Metody dokładne	Zawsze znajdują najlepsze możliwe rozwiązanie.
Metody przybliżone	Pozwalają na określenie dokładności przybliżenia.
Metody losowe	Nie dają gwarancji znalezienia rozwiązania ani nie pozwalają na określenie dokładności rozwiązania.
Metody lokalne	Metody szukające optimum lokalnego.
Metody globalne	Metody szukające najlepszego rozwiązania w całej przestrzeni rozwiązań.
Metody populacyjne	Algorytm operuje na wielu potencjalnych rozwiązaniach jednocześnie.
Metoda – jedno rozwiązanie	Algorytm operuje na jednym rozwiązaniu.

Metody równoległe	Algorytmy wykorzystujące równoległość do przyspieszenia obliczeń lub poprawy jakości.
Metody inspirowane naturą	Takie które zasady swojego działania czerpią z inspiracji przyrodniczych.
Metody hybrydowe	Algorytmy łączące różne techniki w celu poszukiwania optimum.
Metody memetyczne	Populacyjne algorytm optymalizacji globalnej które wykorzystują dodatkowo metody optymalizacji lokalnej.
Algorytm wspinaczkowy ang. HILL CLIMBING (HC)	Algorytm działa w ten sposób, że modyfikuje nieznacznie rozwiązanie i sprawdza czy uzyskane nowe rozwiązanie jest lepsze. Kategoria: metoda lokalna, losowa, operująca na jednym rozwiązaniu
Algorytm tabu albo metoda tabu ang. TABU SEARCH	Mamy jedno rozwiązanie. Modyfikujemy je w ten sposób, że staramy się znaleźć najlepszego sąsiada wykluczając sąsiadów już odwiedzonych (tabu). Rozmiar tabu jest parametrem metody. Kategoria: Metoda globalna, losowa, operująca na jednym rozwiązaniu
Która metaheurystyka jest najlepsza???	To zależy. Praca z metaheurystykami wymaga sprawdzenia dla jakich parametrów uzyskamy najlepsze wyniki. Niestety każdy problem jest trochę inny i należy się do tego dostosować.
Jak zacząć?	Należy zastanowić się: - z jakiego zakresu możemy podawać rozmiary zadania - w jaki sposób wygenerować różne przypadki testowe - zdecydować z ilu eksperymentów będziemy wyciągali średnia (polecam powyżej 25) - zdecydować jakie metody chcemy sprawdzić - określić co jest dla nas ważne (czas obliczeń, jakość)
Symulowane wyżarzanie, zmienne ang. SIMULATED ANNEALING	<ul style="list-style-type: none"> <li>- D – dziedzina</li> <li>- <math>f(x)</math> – funkcja oceny</li> </ul>

	<ul style="list-style-type: none"> <li>- <math>s_k</math> – kolejne rozwiązania, <math>s_0</math> wyznaczane losowo ze zbioru możliwych rozwiązań</li> <li>- <math>u_k</math> – wartości zmiennej losowej o rozkładzie jednostajnym na odcinku <math>(0, 1)</math></li> <li>- <math>N(s)</math> – sąsiedztwo rozwiązania <math>s</math></li> <li>- <math>T_k</math> – temperatura w kroku <math>k</math></li> </ul>
Symulowane wyżarzanie, schemat ang. SIMULATED ANNEALING	<ul style="list-style-type: none"> <li>- Wyznacz <math>s_0 \in D, V = \{s_0\}</math></li> <li>- Iteruj po <math>k</math> od 1 do <math>K</math> <ul style="list-style-type: none"> <li>o Generuj losowo <math>t_k \in N(s_{k-1})</math></li> <li>o Jeśli <math>f(t_k) \geq f(s_{k-1})</math> to <ul style="list-style-type: none"> <li>▪ <math>s_k = t_k</math></li> <li>▪ <math>V = V_{k-1} \cup \{s_k\}</math></li> </ul> </li> <li>o W przeciwnym przypadku <ul style="list-style-type: none"> <li>▪ Jeśli <math>u_k &lt; \exp\left(-\frac{ f(t_k) - f(s_{k-1}) }{T_k}\right)</math> wtedy <math>s_k = t_k</math> - akceptujemy gorsze rozwiązanie,  <math>V_k = V_{k-1} \cup \{s_k\}</math></li> <li>▪ W przeciwnym przypadku <math>s_k = s_{k-1}</math></li> </ul> </li> </ul> </li> </ul>
Algorytm genetyczny ang. GENETIC ALGORITHM (GA)	<ul style="list-style-type: none"> <li>- Genotyp (ciągi kodowe według tradycyjnej terminologii)</li> <li>- Funkcja dekodująca genotyp do fenotypu</li> <li>- Funkcja oceny</li> <li>- Operator selekcji</li> <li>- Operator krzyżowania</li> <li>- Operator mutacji</li> <li>- Warunek zakończenia</li> </ul>
Pomysł wziął się z inspiracji biologią	<ul style="list-style-type: none"> <li>- Zespół sztucznych organizmów</li> <li>- Pokolenia</li> <li>- Ewolucja</li> <li>- Przetwarzanie najlepszych (ocena)</li> </ul>
Algorytm genetyczny (AG)	Elementy odróżniające klasyczny AG od innych metod:

	<ul style="list-style-type: none"> <li>• AG nie przetwarzają bezpośrednio parametrów zadania, tylko ich zakodowaną postać (ciągi kodowe)</li> <li>• AG rozpoczynają poszukiwania z kilku punktów (populacja początkowa)</li> <li>• AG korzystają tylko z funkcji celu</li> <li>• AG stosują probabilistyczne metody wyboru</li> </ul>
Genetyka i algorytmy genetyczne, słownictwo	<p>Genetyka – Algorytmy genetyczne</p> <p>Chromosom - ciąg kodowy</p> <p>Gen – cecha, znak</p> <p>Allel – wariant cechy (0 albo 1)</p> <p>Locus – pozycja</p> <p>Genotyp – struktura składająca się z jednego lub wielu chromosomów (na tym operują AG)</p> <p>Fenotyp - zbiór parametrów, rozwiązanie/osobnik</p>
Uwaga odnośnie do algorytmu genetycznego	Ponieważ w praktyce najczęściej rozpatruje się genotypy składające się tylko z jednego chromosomu, więc często będę utożsamiał chromosom z genotypem.
Funkcja oceny w algorytmie genetycznym	Funkcja obliczająca przystosowanie danego osobnika.
Funkcja celu w algorytmie genetycznym	<p>Funkcja, którą maksymalizujemy lub minimalizujemy. W literaturze są czasami używane takie oznaczenia:</p> <ul style="list-style-type: none"> <li>- funkcja kosztu <math>g(x)</math> - minimalizujemy</li> <li>- funkcja zysku <math>u(x)</math> - maksymalizujemy</li> </ul>
Funkcja przystosowania / fitness $f(x)$	<p>Nieujemne kryterium jakości. Jeśli funkcję celu maksymalizujemy, to bardzo często jest ona taka sama jak funkcja celu.</p> <p>Funkcja fitness ma kluczowe znaczenie dla właściwości algorytmu genetycznego. Jeśli będzie źle zrobiona, to w skrajnym przypadku nasz algorytm utknie i nie da dobrego rozwiązania.</p>
Funkcja oceny / przystosowania / fitness	<p>Funkcja oceny przypisuje danemu rozwiązaniu <math>x</math> ocenę <math>f(x)</math>. Im wartość <math>f(x)</math> jest wyższa, tym dane rozwiązanie <math>x</math> jest lepsze.</p> <p>Funkcja <math>f(x)</math> przyjmuje wartości nieujemne.</p>

	<p>Za pomocą funkcji oceny możemy określić które z dwóch rozwiązań jest lepsze (ma wyższą wartość funkcji oceny).</p> <p>Mając gotowy algorytm genetycznym jedyne co musimy zdefiniować, to funkcja oceny oraz dobrać parametry.</p>
Warunki zakończenia	<ul style="list-style-type: none"> <li>- Liczba iteracji</li> <li>- Liczba wykonań obliczania funkcji fitness</li> <li>- Zadana jakość rozwiązań</li> <li>- Zadana różnorodność (albo raczej jej brak) rozwiązań</li> <li>- Czas obliczeń</li> </ul>
Ciągi kodowe – chromosomy	Ciągi które kodują rozwiązania. Tradycyjnie są to ciągi zer i jedynek.
Elementarny algorytm genetyczny	<ul style="list-style-type: none"> <li>- Inicjalizacja populacji początkowej</li> <li>- Powtarzanie, dopóki warunek nie jest spełniony: <ul style="list-style-type: none"> <li>- Reprodukacja (ocena + selekcja)</li> <li>- Krzyżowanie</li> <li>- Mutacja</li> </ul> </li> </ul>
Parametry elementarnego algorytmu genetycznego	<ul style="list-style-type: none"> <li>- Funkcja oceny</li> <li>- Rozmiar populacji</li> <li>- Metoda selekcji i jej parametry</li> <li>- Prawdopodobieństwo mutacji, krzyżowania (interpretowane w zależności od wybranych algorytmów)</li> <li>- Liczba iteracji lub inny warunek zakończenia</li> </ul>
Reprodukacja / selekcja	<p>Ciągi kodowe są powielane w zależności od tego, jakie wartości przyjmuje dla nich funkcja celu / przystosowania.</p> <p>Analogia do przeżywania najlepiej przystosowanych osobników w środowisku naturalnym.</p>
Pula rodzicielska	Selekcja osobników według wybranego algorytmu tworzy tak zwaną pulę rodzicielską dla nowego pokolenia.
Metoda ruletki – klasyczna metoda selekcji	Tworzymy wirtualne koło ruletki i losowo wybieramy osobniki.
Krzyżowanie	Wybieramy losowo pary osobników z puli rodzicielskiej i krzyżujemy je generując

	nowe osobniki. W tym momencie wymieniamy fragmenty ciągów kodowych.
Krzyżowanie jednopunktowe	W tradycyjnym AG stosuje się krzyżowanie jednopunktowe. Polega ono na tym, że wyznaczamy losowo punkt podziału ciągów kodowych i zmieniamy podciągi w obu ciągach kodowych.
Mutacja	Ta operacja zachodzi z małym prawdopodobieństwem. Polega na przypadkowym, niewielkim zmodyfikowaniu chromosomu (ciągu kodowego).
Adaptacja	Algorytmy genetyczne dobrze dostosowują się do warunków zadania (nawet zmiennych).
Eksploracja, krzyżowanie	Pozwala na dotarcie do nowych obszarów poszukiwań (potencjalnie ciekawych).
Eksploatacja, krzyżowanie, mutacja	Działa jak optymalizacja lokalna – poprawia rozwiązania w niewielkim otoczeniu.
Przedwczesna zbieżność	Jest to utrata przez algorytm optymalizacyjny zdolności przeszukiwania przestrzeni rozwiązań przed osiągnięciem ekstremum globalnego.
Skąd brak zbieżności?	<ul style="list-style-type: none"> <li>- Kodowanie powoduje, że algorytm przeszukuje przestrzeń niezawierającą optimum globalnego</li> <li>- Liczba iteracji musi być skończona</li> <li>- Liczebność populacji musi być skończona</li> </ul>
Superosobniki	<p>Osobniki o bardzo wysokim stopniu przystosowania</p> <ul style="list-style-type: none"> <li>- Przeszkadzają na początku algorytmu (przedwczesna zbieżność)</li> <li>- Bardzo przydatne pod koniec działania algorytmu (zawężenie przestrzeni poszukiwań)</li> </ul>
Metoda kary	Osobnik niepoprawny otrzymuje tak zwaną karę - oznacza to, że jego wartość dostosowania jest w znaczący sposób zmniejszana, a w skrajnym przypadku osobnik jest usuwany.
Algorytm Naprawy	Osobnik niepoprawny jest poprawiany tak, aby mieścił się w założeniach zadania.



Podwójna* selekcja turniejowa (selekcja turniejowa)	<ul style="list-style-type: none"> <li>- Wylosuj osobniki <math>s_1, s_2, \dots, s_n</math> z <math>P(t)</math></li> <li>- Wybierz najlepszego z nich</li> </ul> <p>Często turnieje rozmiaru 2.          * - trzeba jej dokonać 2x, aby uzyskać parę rodziców.</p>
Podwójna selekcja turniejowa – cechy	<ul style="list-style-type: none"> <li>- Świetnie nadaje się do maszyn wieloprocesorowych</li> <li>- Jest bardzo prosta w implementacji</li> <li>- Zapobiega przedwczesnej zbieżności algorytmu</li> <li>- Pozwala na przeszukanie szerszego zakresu potencjalnych rozwiązań nawet przy kiepskiej funkcji fitness</li> </ul>
Pojedyncza selekcja turniejowa	<ul style="list-style-type: none"> <li>- Mieshamy populację</li> <li>- Dzielimy na grupy rozmiaru <math>n</math></li> <li>- Z każdej grupy wybieramy 2 osobników o najlepszym przystosowaniu</li> <li>- Wybrane osobniki posłużą jako rodzice w następnym etapie AG</li> </ul>
Pojedyncza selekcja turniejowa – cechy	<ul style="list-style-type: none"> <li>- Gwarantuje, że zawsze wybierzemy co najmniej <math>n-2</math> najlepszych osobników</li> <li>- Nawet najlepiej przystosowane osobniki produkują jednego potomka (zapobiega przedwczesnej zbieżności)</li> </ul>
Selekcja elitarna (z elitą)	Zawsze na początku wybierz $n$ najlepszych osobników.
Selekcja elitarna – cechy	<ul style="list-style-type: none"> <li>- Zapewnia, że zawsze zachowamy najlepszego osobnika</li> <li>- Przy niewielkiej elicie, zwykle nie ma problemu z przedwczesną zbieżnością</li> <li>- Przy większych elitach może się zdarzyć, że elita z początkowych kroków ewolucji zdominuje całą populację (czyli pewnie nie znajdziemy optimum globalnego)</li> </ul>
Selekcja rangowa (porządkowa)	Sortujemy osobniki według funkcji przystosowania, rangi osobników służą za funkcję przystosowania.

Metody krzyżowania	<ul style="list-style-type: none"> <li>- Jednopunktowe</li> <li>- Wielopunktowe</li> <li>- Jednostajne (zamieniamy geny z zadany�m prawdopodobie�stwem)</li> </ul>
Metody mutacji	Mutacja: <ul style="list-style-type: none"> <li>- Punktowa</li> <li>- Wielopunktowa</li> <li>- Probabilistyczna</li> <li>- U�yteczna</li> <li>- Lamarckowska rz�du <math>k</math></li> </ul>
Mutacja punktowa	Polega na tym, �e modyfikujemy pojedynczy losowy bit chromosomu.
Mutacja wielopunktowa	Polega na tym, �e modyfikujemy $n$ pojedynczych losowych bit�w chromosomu.
Mutacja probabilistyczna (ta z klasycznego Algorytmu Genetycznego)	Sprawdzamy ka�dy bit chromosomu i z zadany�m prawdopodobie�stwem zmieniamy go na przeciwny.
Mutacja u�yteczna	Po wykonaniu mutacji (metoda dowolna) por�wnujemy nowego osobnika z rodzicem i wybieramy lepsze rozwi�zanie.
Mutacja lamarckowska rz�du $k$ (od Lamarkizmu od Jean-Baptiste de Lamarck)	Sprawdzamy wyniki mutacji $n$ punktowych dla $0 \leq n \leq k$ i wybieranie najlepszego rozwi�zania.
Algorytm genetyczny	Uniwersalny spos�b na rozwi�zywanie problem�w. Mamy zawsze taki sam algorytm, natomiast zadanie dostosowujemy do niego.
Program ewolucyjny	Mamy og�lny schemat algorytmu i dostosowujemy go do wymaga�n problemu. Uwaga: Algorytm genetyczny mo�na uzna� za szczeg�lny przypadek algorytmu ewolucyjnego.
Funkcja benchmarkowa	Jest to funkcja, s�u�ąca do testowania algorytmu optymalizacyjnego. Jest wiele funkcji s�u�ących do testowania algorytm�w optymalizacyjnych.
Element losowo�ci	Cech� algorytm�w genetycznych jest to, �e dla takiego samego problemu mo�na obserwowa� r��ne zachowania algorytmu.
Ocena algorytmu	Nale�y przeprowadzi� wiele niezale�nych uruchomie� (co najmniej 25). Je�li stosujemy za ka�dym razem tak� sam�

	populację początkową, wtedy można powiedzieć, że algorytm ma określone właściwości dla takiej populacji.
Porównanie algorytmów	Czasami zdarza się porównywać dwa algorytmy. W takim przypadku dla każdej populacji początkowej mogą być uruchamiane oba algorytmy. Wyniki niezależnych uruchomień należy poddać analizie statystycznej, to znaczy wyznaczyć co najmniej średnia i odchylenie standardowe. Warto oczywiście stworzyć także histogram wyników.
Elementarny algorytm genetyczny	<ul style="list-style-type: none"> <li>- Inicjalizacja populacji początkowej</li> <li>- Ocena</li> <li>- Powtarzanie, dopóki warunek nie jest spełniony: <ul style="list-style-type: none"> <li>- Selekcja</li> <li>- Krzyżowanie</li> <li>- Mutacja</li> <li>- Ocena + nowa populacja</li> </ul> </li> </ul>
Współbieżność obliczeń ang. CONCURRENCY	Wiele wątków wykonania. Mogą być wykonywane naprzemiennie (przetaczanie kontekstu).
Równoległość obliczeń ang. PARALLELISM	Wiele procesorów / rdzeni wykonuje wiele zadań w tym samym momencie. Program napisany współbieżnie może wykonywać się równolegle.
Model rozproszony ang. DISTRIBUTED COMPUTING	Oddzielne procesory połączone jakimiś kanałami komunikacyjnymi.
Wielokrotne uruchamianie	Po prostu uruchamiamy jednocześnie program na wielu komputerach / rdzeniach.
Model pan i niewolnik ang. MASTER-SLAVE	Procesor / komputer główny rozsyła zadania obliczenia funkcji oceny do podwładnych (SLAVE-ów).
Model rozproszony	Praca jest rozdysponowana po równo między workerami. Ustalony jest model wymiany osobników.
Algorytm wyspowy ang. DISTRIBUTED GA (dGAs)	Działa on w ten sposób, że tworzone jest $N$ wysp – instancji algorytmu - które co jakiś czas wymieniają się osobnikami.
Komórkowy Algorytm Genetyczny ang. CELLULAR GA (cGA)	Ustalone jest bliskie sąsiedztwo małych instancji algorytmu genetycznego.

Algorytm wyspowy i komórkowy	W obu przypadkach proces ujednolicania się populacji jest spowolniony. Daje to większą szansę na lepsze rozwiązanie.
Panmiksja / panmiktyczny ang. PANMIXIA / PANMICTIC	Pojedyncza populacja - każdy ma szansę z każdym. SJP.PL: swobodne, zazwyczaj przypadkowe kojarzenie się roślin lub zwierząt zapładniających się krzyżowo
Wyspy / demy ang. ISLANDS / DEMES	Podział na oddzielone instancje algorytmu SJP.PL: jednostka administracyjna starożytnego państwa ateńskiego
Algorytm wyspowy:	
Heterogeniczny AG	Każda wyspa może mieć inny zestaw parametrów
Przerwa migracyjna ang. MIGRATION GAP	Jak często osobniki są wymieniane (probabilistycznie lub nie).
Tempo migracji ang. MIGRATION RATE	Liczba migrantów w każdej fali migracji.
Sposób "goszczenia" migrantów ang. SELECTION / REPLACEMENT OF MIGRANTS	W jaki sposób migranci są dołączani - na przykład przez zastąpienie najgorszego.
Topologia	Które wyspy ze sobą sąsiadują
Jak porównywać wyniki – jak sobie radzić?	<ul style="list-style-type: none"> <li>- Można przed każdym etapem generować serie liczb pseudolosowych</li> <li>- Można mieć kilka generatorów / ziaren i każdy z nich przydzielać do odpowiednich zakresów wątków</li> <li>- Można skorzystać z tego, że jest to proces stochastyczny i liczyć statystykę</li> </ul>
OpenMP	Biblioteka dostępna w kompilatorach C/C++ do implementacji modelu równoległego z pamięcią współdzieloną.
MPI	Klasyczna biblioteka do obliczeń rozproszonych.
Reprodukcja	Dokonujemy selekcji ze zwracaniem $\lambda$ kopii osobników z populacji bazowej.
Programowanie genetyczne	Reprezentacja drzewiasta. Rozwinięcie AG o reprezentację drzewiastą programu w języku LISP. Schemat algorytmu taki jak przy Algorytmie Genetycznym.
Drzewiasta reprezentacja	Programy reprezentowane jako drzewa na podstawie języka LISP.

Programowanie genetyczne – operatory mutacji	<ul style="list-style-type: none"> <li>- Zmiana zawartości węzła terminalnego</li> <li>- Zamiast węzła terminalnego wstawiamy węzeł pośredni z losowym poddrzewem</li> <li>- Węzeł pośredni zamieniony na terminalny</li> <li>- Reorganizacja (zmiana) poddrzew węzła pośredniego</li> </ul>
Programowanie genetyczne – operatory krzyżowania	Zmiana poddrzew chromosomów rodzicielskich