

## Algorytmy ewolucyjne

### Czym jest ewolucja?

Kiedy patrzymy na otaczający nas świat, czasami zastanawiamy się, jak powstało wszystko, co widzimy i z czym mamy do czynienia. Jednym ze sposobów wyjaśnienia tego jest teoria ewolucji. Teoria ewolucji sugeruje, że żywe organizmy, które widzimy dzisiaj, nie nagle istniały w ten sposób, ale ewoluowały przez miliony lat subtelnych zmian, z każdym pokoleniem dostosowującym się do swojego środowiska. Oznacza to, że fizyczne i poznawcze cechy każdego żywego organizmu są wynikiem najlepszego dopasowania do ich środowiska w celu przetrwania. Ewolucja sugeruje, że organizmy ewoluują poprzez rozmnażanie, wytwarzając dzieci o mieszanych genach od swoich rodziców. Biorąc pod uwagę przydatność osobników w swoim środowisku, silniejsze osobniki mają większe prawdopodobieństwo przeżycia. Często popełniamy błąd, myśląc, że ewolucja jest procesem liniowym, z wyraźnymi zmianami następców. W rzeczywistości ewolucja jest znacznie bardziej chaotyczna, z rozbieżnościami w gatunku. W wyniku rozmnażania i mieszania genów powstaje wiele wariantów gatunku. Zauważalne różnice między gatunkami mogą się ujawnić i urzeczywistnić po tysiącach lat tylko poprzez porównanie przeciętnego osobnika w każdym z tych punktów czasowych. Charles Darwin zaproponował teorię ewolucji, która koncentruje się na doborze naturalnym. Dobór naturalny to koncepcja, zgodnie z którą silniejsi członkowie populacji mają większe szanse na przeżycie, ponieważ są bardziej przystosowani do swojego środowiska, co oznacza, że częściej rozmnażają się, a tym samym mają cechy które są korzystne dla przetrwania dla przyszłych pokoleń - które mogłyby potencjalnie działać lepiej niż ich przodkowie. Klasycznym przykładem ewolucji przystosowawczej jest ćma pieprzowa. Ćma pieprzowa była pierwotnie w jasnym kolorze, co zapewniało dobry kamuflaż przed drapieżnikami, ponieważ ćma mogła wtapiać się w jasne powierzchnie w swoim otoczeniu. Tylko około 2 procent populacji ćmy miało ciemniejszy kolor. Po rewolucji przemysłowej około 95 procent gatunków miało ciemniejszy wariant kolorystyczny. Jednym z wyjaśnień jest to, że jaśniejsze ćmy nie mogły już mieszać się z tyłoma powierzchniami, ponieważ zanieczyszczenia miały ciemne powierzchnie; w ten sposób jaśniejsze ćmy były bardziej zjadane przez drapieżniki, ponieważ te ćmy były bardziej widoczne. Ciemniejsze ćmy miały większą przewagę w wtapieniu się w ciemniejsze powierzchnie, dzięki czemu przetrwały dłużej i rozmnażały się więcej, a ich informacje genetyczne były szerzej rozpowszechniane wśród następców. Wśród ćmy pieprzowej cechą, która zmieniła się na wysokim poziomie, był kolor ćmy. Jednak ta właściwość nie tylko magicznie się zmieniła. Aby zmiana mogła nastąpić, geny ćmy o ciemniejszym kolorze musiały zostać przeniesione do następców. W innych przykładach naturalnej ewolucji możemy zobaczyć dramatyczne zmiany nie tylko w kolorze między różnymi osobnikami, ale w rzeczywistości na zmiany te wpływają różnice genetyczne niższego poziomu na przestrzeni wielu pokoleń. Ewolucja obejmuje ideę, że w populacji gatunków rozmnażają się pary organizmów. Potomstwo jest kombinacją genów rodzica, ale małe zmiany są wprowadzane w potomstwie poprzez proces zwany mutacją. Następnie potomstwo staje się częścią populacji. Jednak nie wszyscy członkowie populacji żyją dalej. Jak wiemy, choroby, urazy i inne czynniki powodują śmierć ludzi. Osoby, które są bardziej przystosowane do otaczającego ich środowiska, są bardziej skłonne do życia, co dało początek określeniu jako przetrwanie najlepiej przystosowanych. Opierając się na darwinowskiej teorii ewolucji, populacja ma następujące cechy:

- Różnorodność - osoby w populacji mają różne cechy genetyczne.
- Dziedziczna - dziecko dziedziczy właściwości genetyczne po rodzicach.
- Selekcja - mechanizm mierzący sprawność osobników, a silniejsze osobniki mają najwyższe prawdopodobieństwo przeżycia (przeżycie najlepiej przystosowanych).

Te właściwości sugerują, że podczas procesu ewolucji zachodzą następujące rzeczy:

- Rozmnażanie - Zwykle dwie osobniki w populacji rozmnażają się, aby stworzyć potomstwo.
- Krzyżowanie i mutacja - potomstwo stworzone w wyniku rozmnażania zawiera mieszankę genów swoich rodziców i ma niewielkie, losowe zmiany w kodzie genetycznym

Podsumowując, ewolucja jest cudownym i chaotycznym systemem, który wytwarza różnorodne formy życia, z których niektóre są lepsze od innych dla określonych rzeczy w określonych środowiskach. Teoria ta odnosi się również do algorytmów ewolucyjnych; wnioski płynące z ewolucji biologicznej są wykorzystywane do znajdowania optymalnych rozwiązań praktycznych problemów poprzez generowanie różnorodnych rozwiązań i łączenie się z lepszymi rozwiązaniami przez wiele pokoleń. Ten rozdział i rozdział 5 poświęcone są badaniu algorytmów ewolucyjnych, które są potężnymi, ale niedocenianymi podejściami do rozwiązywania trudnych problemów. Algorytmy ewolucyjne mogą być używane oddzielnie lub w połączeniu z konstrukcjami, takimi jak sieci neuronowe. Solidne zrozumienie tej koncepcji otwiera wiele możliwości rozwiązywania różnych nowych problemów.

### Problemy mające zastosowanie do algorytmów ewolucyjnych

Algorytmy ewolucyjne nie nadają się do rozwiązywania wszystkich problemów, ale są potężne w rozwiązywaniu problemów optymalizacyjnych, w których rozwiązanie składa się z dużej liczby permutacji lub wyborów. Te problemy zazwyczaj składają się z wielu ważnych rozwiązań, przy czym niektóre są bardziej optymalne niż inne. Rozważmy problem plecakowy, klasyczny problem używany w informatyce do badania działania algorytmów i ich wydajności. W Problemie plecakowym plecak ma określoną maksymalną wagę, jaką może pomieścić. W plecaku można przechowywać kilka przedmiotów, a każdy z nich ma inną wagę i wartość. Celem jest zmieszczenie w plecaku jak największej liczby przedmiotów, tak aby całkowita wartość była jak największa, a całkowita waga nie przekraczała limitu plecaka. Fizyczny rozmiar i wymiary przedmiotów są ignorowane w najprostszym wariacie problemu. Jako trywialny przykład, biorąc pod uwagę specyfikację problemu w tabeli, plecak może pomieścić całkowity ciężar 9 kg i może zawierać jedną z ośmiu przedmiotów o różnej wadze i wartości.

Item ID	Item name	Weight (kg)	Value
1	Pearls	3	4
2	Gold	7	7
3	Crown	4	5
4	Coin	1	1
5	Axe	5	4
6	Sword	4	3
7	Ring	2	5
8	Cup	3	1

Ten problem ma 255 możliwych rozwiązań, w tym następujące:

- Rozwiązanie 1 - uwzględnij pozycję 1, pozycję 4 i pozycję 6. Całkowita waga wynosi 8 kg, a łączna wartość to 8.

- Rozwiązanie 2 - uwzględnij pozycję 1, pozycję 3 i pozycję 7. Całkowita waga wynosi 9 kg, a łączna wartość to 14.
- Rozwiązanie 3 - Uwzględnij punkt 2, punkt 3 i punkt 6. Całkowita waga wynosi 15 kg, co przekracza pojemność plecaka.

Najwyraźniej rozwiązaniem o największej wartości jest Rozwiązanie 2. Nie przejmuj się zbyttno sposobem obliczania liczby możliwości, ale pamiętaj, że możliwości eksplodują wraz ze wzrostem liczby potencjalnych elementów. Chociaż ten trywialny przykład można rozwiązać ręcznie, Problem plecaka może mieć różne ograniczenia wagi, różną liczbę przedmiotów oraz różne wagi i wartości dla każdego przedmiotu, co uniemożliwia jego ręczne rozwiązanie, gdy zmienne rosną. Próba brutalnego wymuszenia każdej kombinacji elementów, gdy zmienne rosną, będzie również kosztowna obliczeniowo; dlatego szukamy algorytmów, które są skuteczne w znajdowaniu pożądanego rozwiązania. Zwróć uwagę, że najlepsze rozwiązanie, jakie możemy znaleźć, kwalifikujemy jako rozwiązanie pożyteczne, a nie optymalne. Chociaż niektóre algorytmy próbują znaleźć jedyne, prawdziwie optymalne rozwiązanie problemu plecaka, algorytm ewolucyjny próbuje znaleźć optymalne rozwiązanie, ale nie gwarantuje jego znalezienia. Algorytm znajdzie jednak rozwiązanie, które jest do przyjęcia dla przypadku użycia - subiektywną opinię na temat tego, jakie rozwiązanie jest akceptowalne, na podstawie problemu. Na przykład w przypadku systemu opieki zdrowotnej o znaczeniu krytycznym „dostatecznie dobre” rozwiązanie może tego nie wystarczyć, ale w przypadku systemu rekomendującego piosenki może być akceptowalne. Rozważmy teraz większy zbiór danych (tak, gigantyczny plecak) w tabeli poniżej, w którym liczba przedmiotów oraz różne wagi i wartości utrudniają rozwiązanie problemu ręcznie. Rozumiejąc złożoność tego zbioru danych, można łatwo zrozumieć, dlaczego wiele algorytmów informatycznych jest mierzonych na podstawie ich wydajności w rozwiązywaniu takich problemów. Wydajność jest definiowana jako dobrze, jak określone rozwiązanie rozwiązuje problem, niekoniecznie wydajność obliczeniowa. W przypadku problemu plecakowego rozwiązanie, które daje wyższą wartość całkowitą, byłoby skuteczniejsze. Algorytmy ewolucyjne zapewniają jedną metodę znajdowania rozwiązań problemu plecakowego.

Item ID	Item name	Weight (kg)	Value
1	Axe	32252	68674
2	Bronze coin	225790	471010
3	Crown	468164	944620
4	Diamond statue	489494	962094
5	Emerald belt	35384	78344
6	Fossil	265590	579152
7	Gold coin	497911	902698
8	Helmet	800493	1686515
9	Ink	823576	1688891
10	Jewel box	552202	1056157
11	Knife	323618	677562
12	Long sword	382946	833132
13	Mask	44676	99192
14	Necklace	169738	376418
15	Opal badge	610876	1253986
16	Pearls	854190	1853562
17	Quiver	671123	1320297
18	Ruby ring	698180	1301637
19	Silver bracelet	446517	859835

20	Timepiece	909620	1677534
21	Uniform	904818	1910501
22	Venom potion	730061	1528646
23	Wool scarf	931932	1827477
24	Crossbow	952360	2068204
25	Yesteryear book	928023	1746556
26	Zinc cup	978724	2100851

Jednym ze sposobów rozwiązania tego problemu jest zastosowanie metody brutalnej siły. To podejście polega na obliczeniu każdej możliwej kombinacji przedmiotów i określeniu wartości każdej kombinacji, która spełnia ograniczenie wagi plecaka, aż do znalezienia najlepszego rozwiązania. Rysunek przedstawia niektóre analizy porównawcze dla podejścia brutalnej siły. Zauważ, że obliczenia są oparte na sprzęcie przeciętnego komputera osobistego.

Combinations	$2^{26} = 67\,108\,864$
Iterations	$2^{26} = 67\,108\,864$
Accuracy	100%
Compute Time	~7 minutes

Pamiętaj o problemie plecakowym, ponieważ będzie on używany w tej części, gdy będziemy próbować zrozumieć, zaprojektować i opracować algorytm genetyczny w celu znalezienia akceptowalnych rozwiązań tego problemu.

**UWAGA** : Z punktu widzenia indywidualnego rozwiązania, wydajność oznacza to, jak dobrze rozwiązanie rozwiązuje problem. Z punktu widzenia algorytmu wydajność może zależeć od tego, jak dobrze określona konfiguracja radzi sobie w znalezieniu rozwiązania. Wreszcie wydajność może oznaczać cykle obliczeniowe. Pamiętaj, że ten termin jest używany w różny sposób w zależności od kontekstu.

Myśl o wykorzystaniu algorytmu genetycznego do rozwiązania problemu plecaka można zastosować do szeregu praktycznych problemów. Jeśli firma logistyczna chce na przykład zoptymalizować pakowanie ciężarówek na podstawie ich miejsc docelowych, przydatny byłby algorytm genetyczny. Gdyby ta sama firma chciała znaleźć najkrótszą trasę między kilkoma miejscami docelowymi, przydatny byłby również algorytm genetyczny. Jeśli fabryka przerabia elementy na surowiec za pomocą systemu przenośników taśmowych, a kolejność elementów wpływa na produktywność, do określenia tej kolejności przydatny byłby algorytm genetyczny. Kiedy zagłębimy się w myślenie, podejście i cykl życia algorytmu genetycznego, powinno być jasne, gdzie można zastosować ten potężny algorytm i być może pomyślisz o innych zastosowaniach w swojej pracy. Należy pamiętać, że algorytm genetyczny jest stochastyczny, co oznacza, że dane wyjściowe algorytmu mogą być różne przy każdym uruchomieniu.

### Algorytm genetyczny: Cykl życia

Algorytm genetyczny to specyficzny algorytm z rodziny algorytmów ewolucyjnych. Każdy algorytm działa na tym samym założeniu ewolucji, ale ma drobne poprawki w różnych częściach cyklu życia, aby rozwiązać różne problemy. Omówimy niektóre z tych parametrów w rozdziale 5. Algorytmy genetyczne służą do oceny dużych przestrzeni wyszukiwania w celu znalezienia dobrego rozwiązania. Należy zauważyć, że algorytm genetyczny nie gwarantuje znalezienia absolutnie najlepszego rozwiązania; stara się znaleźć najlepsze globalne, unikając jednocześnie najlepszych lokalnych rozwiązań.

Najlepsze rozwiązanie globalne to najlepsze możliwe rozwiązanie, a najlepsze rozwiązanie lokalne to rozwiązanie mniej optymalne. Rysunek przedstawia możliwe najlepsze rozwiązania, jeśli rozwiązanie musi być zminimalizowane - to znaczy im mniejsza wartość, tym lepiej. Jeśli celem było maksymalizacja rozwiązania, im większa wartość, tym lepiej. Algorytmy optymalizacyjne, takie jak algorytmy genetyczne, mają na celu stopniowe znajdowanie najlepszych lokalnych rozwiązań w poszukiwaniu najlepszego globalnego rozwiązania.



Przy konfiguowaniu parametrów algorytmu należy zachować ostrożność, aby na początku dążyć do różnorodności rozwiązań i stopniowo z każdym pokoleniem dążyć do lepszych rozwiązań. Na początku potencjalne rozwiązania powinny znacznie różnić się pod względem poszczególnych atrybutów genetycznych. Bez rozbieżności na początku zwiększa się ryzyko utknięcia w lokalnym najlepszym.

Przy konfiguowaniu parametrów algorytmu należy zachować ostrożność, aby na początku dążyć do różnorodności rozwiązań i stopniowo z każdym pokoleniem dążyć do lepszych rozwiązań. Na początku potencjalne rozwiązania powinny znacznie różnić się pod względem poszczególnych atrybutów genetycznych. Bez rozbieżności na początku zwiększa się ryzyko utknięcia w lokalnym najlepszym. Konfiguracja algorytmu genetycznego opiera się na przestrzeni problemu. Każdy problem ma unikalny kontekst i inna dziedzina, w której reprezentowane są dane, a rozwiązania są oceniane w różny sposób. Ogólny cykl życia algorytmu genetycznego wygląda następująco:

- Tworzenie populacji - tworzenie losowej populacji potencjalnych rozwiązań.
- Pomiar sprawności osób w populacji - określenie, jak dobre jest dane rozwiązanie. Zadanie to jest realizowane za pomocą funkcji sprawności, która ocenia rozwiązania, aby określić ich jakość.
- Wybór rodziców na podstawie ich sprawności - wybór par rodziców, którzy będą rozmnażać potomstwo.
- Rozmnażanie osobników od rodziców - tworzenie potomstwa od rodziców poprzez mieszanie informacji genetycznej i stosowanie niewielkich mutacji u potomstwa.
- Populowanie następnego pokolenia - wybieranie osobników i potomstwa z populacji, które przetrwają do następnego pokolenia.

Wdrażanie algorytmu genetycznego obejmuje kilka kroków. Kroki te obejmują etapy cyklu życia algorytmu.

Mając na uwadze problem plecakowy, w jaki sposób użylibyśmy algorytmu genetycznego do znalezienia rozwiązania problemu? Kolejna sekcja omawia ten proces.

### **Kodowanie przestrzeni rozwiązań**

Kiedy używamy algorytmu genetycznego, najważniejsze jest prawidłowe wykonanie kroku kodowania, co wymaga starannego zaprojektowania reprezentacji możliwych stanów. Stan to struktura danych z określonymi regułami, które reprezentują możliwe rozwiązania problemu. Ponadto zbiór stanów obejmuje populację.

### **Terminologia**

W odniesieniu do algorytmów ewolucyjnych indywidualne rozwiązanie kandydata nazywa się chromosomem. Chromosom składa się z genów. Gen jest logicznym typem jednostki, a allel to rzeczywista wartość przechowywana w tej jednostce. Genotyp jest reprezentacją rozwiązania, a fenotyp sam w sobie jest unikalnym rozwiązaniem. Każdy chromosom ma zawsze taką samą liczbę genów. Zbiór chromosomów tworzy populację. W przypadku problemu z plecakiem do plecaka można włożyć kilka przedmiotów. Prosty sposób opisania możliwego rozwiązania, które zawiera niektóre elementy, a innych nie jest kodowanie binarne. Kodowanie binarne reprezentuje wykluczone elementy z zerami i uwzględnione elementy z 1. Jeśli wartość w indeksie genu 3 wynosi na przykład 1, ten element jest oznaczony jako uwzględniony. Cały ciąg binarny ma zawsze ten sam rozmiar: liczbę elementów dostępnych do wyboru. Istnieje jednak kilka alternatywnych schematów kodowania, które zostały opisane w rozdziale 5.

### **Kodowanie binarne: przedstawienie możliwych rozwiązań za pomocą zer i jedynek**

Kodowanie binarne reprezentuje gen w zakresie 0 lub 1, więc chromosom jest reprezentowany przez ciąg binarnych bitów. Kodowanie binarne może być używane na wiele sposobów, aby wyrazić obecność określonego elementu, a nawet zakodować wartości liczbowe jako liczby binarne. Zaletą kodowania binarnego jest to, że jest ono zwykle bardziej wydajne ze względu na użycie używanego typu pierwotnego. Korzystanie z kodowania binarnego wiąże się z mniejszym zapotrzebowaniem na pamięć roboczą i w zależności od używanego języka operacje binarne są obliczeniowo szybciej. Należy jednak zastosować krytyczną myśl, aby upewnić się, że kodowanie ma sens dla danego problemu i dobrze przedstawia potencjalne rozwiązania; w przeciwnym razie algorytm może działać słabo. Biorąc pod uwagę problem plecakowy ze zbiorem danych, który składa się z 26 elementów o różnej wadze i wartości, do przedstawienia każdego elementu można użyć łańcucha binarnego. Wynikiem jest 26-znakowy ciąg, w którym dla każdego indeksu 0 oznacza, że odpowiednia pozycja jest wykluczona, a 1 oznacza, że zawarty jest odpowiedni element. Inne schematy kodowania - w tym kodowanie wartości rzeczywistych, kodowanie według kolejności i kodowanie drzewiaste.

### **Tworzenie populacji rozwiązań**

Na początku powstała populacja. Pierwszym krokiem w algorytmie genetycznym jest inicjalizacja losowych potencjalnych rozwiązań danego problemu. W procesie inicjalizacji populacji, chociaż chromosomy są generowane losowo, należy wziąć pod uwagę ograniczenia problemu, a potencjalne rozwiązania powinny być prawidłowe lub mieć fatalny wynik sprawności, jeśli naruszają ograniczenia. Każda osoba w populacji może nie rozwiązać problemu dobrze, ale rozwiązanie jest ważne. Jak wspomniano we wcześniejszym przykładzie pakowania przedmiotów do plecaka, rozwiązanie określające pakowanie tego samego przedmiotu więcej niż jeden raz powinno być rozwiązaniem nieprawidłowym i nie powinno stanowić części populacji potencjalnych rozwiązań. Biorąc pod uwagę, w jaki sposób reprezentowany jest stan rozwiązania problemu plecakowego, ta implementacja losowo decyduje, czy każdy przedmiot powinien znajdować się w torbie. To powiedziawszy, należy brać pod uwagę tylko rozwiązania, które spełniają ograniczenie wagi. Problem polega na prostym przechodzeniu od lewej do prawej i losowym wybieraniu, czy przedmiot zawiera odchylenie w kierunku elementów na lewym końcu chromosomu. Podobnie, jeśli zaczniemy od prawej strony, będziemy nastawieni w kierunku przedmiotów po prawej stronie. Jednym z możliwych sposobów obejścia tego problemu jest wygenerowanie całej osoby z przypadkowymi genami, a następnie określenie, czy rozwiązanie jest prawidłowe i nie narusza żadnych ograniczeń. Przypisanie strasznej oceny nieprawidłowym rozwiązaniom może rozwiązać ten problem.

### **Pseudo kod**

Aby wygenerować początkową populację możliwych rozwiązań, tworzona jest pusta tablica do przechowywania jednostek. Następnie dla każdego osobnika w populacji tworzona jest pusta tablica do przechowywania genów osobnika. Każdy gen jest losowo ustawiany na 1 lub 0, co wskazuje, czy pozycja w tym indeksie genu jest uwzględniona.

```

generate_initial_population (population_size, individual_size)
  let population be an empty array
  for individual in range 0 to population_size
    let current_individual be an empty array
    for gene in range 0 to individual_size
      let random_gene be 0 or 1 randomly
      append random_gene to current_individual
    append current_individual to population
  return population

```

### Pomiar sprawności osobników w populacji

Po utworzeniu populacji należy określić przydatność każdej osoby w populacji. Fitness określa, jak dobrze działa rozwiązanie. Funkcja przystosowania ma kluczowe znaczenie dla cyklu życia algorytmu genetycznego. Jeśli sprawność fizyczna osób zostanie zmierzona nieprawidłowo lub w sposób, który nie będzie dążył do znalezienia optymalnego rozwiązania, wpłynie to na proces selekcji rodziców nowych osób i nowych pokoleń - algorytm będzie wadliwy i nie będzie starał się znaleźć najlepsze możliwe rozwiązanie. Funkcje sprawności są podobne do heurystyk, które omówiliśmy w rozdziale 3. Są one wskazówkami do znajdowania dobrych rozwiązań. W naszym przykładzie rozwiązanie stara się zmaksymalizować wartość przedmiotów w plecaku przy jednoczesnym przestrzeganiu ograniczeń wagi. Funkcja sprawności mierzy całkowitą wartość przedmiotów w plecaku dla każdej osoby. W rezultacie osoby o wyższych wartościach całkowitych są bardziej sprawne. Zwróć uwagę, że na rysunku pojawia się osoba niepełnosprawna, aby podkreślić, że jej wynik sprawności dałby 0 - okropny wynik, ponieważ przekracza dopuszczalną wagę dla tego przypadku problemu, która wynosi 6 404 180.

A 

0	1	1	0	0	1	0	0	0	1	0	1	1	0	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 11393368

B 

0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 10865684

C 

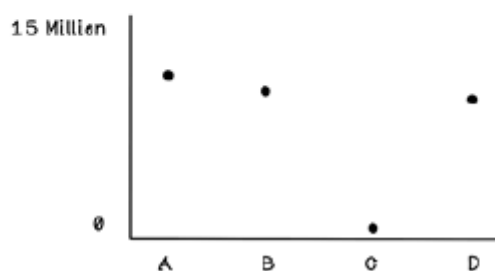
1	1	1	0	0	1	0	0	1	1	0	1	1	0	0	0	0	1	0	0	1	0	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 0 (Overweight)

D 

0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 10715475



W zależności od rozwiązywanego problemu, wynik funkcji dopasowania może wymagać minimalizacji lub maksymalizacji. W Problemie plecakowym zawartość plecaka można zmaksymalizować w ramach



ograniczeń lub zmniejszyć do minimum pustą przestrzeń w plecaku. Podejście zależy od interpretacji problemu.

### Pseudo kod

Aby obliczyć dopasowanie osoby do problemu plecakowego, należy określić sumę wartości każdego przedmiotu, który obejmuje dana osoba. Zadanie to jest wykonywane przez ustawienie całkowitej wartości na 0, a następnie iterowanie po każdym genie w celu określenia, czy element, który reprezentuje, jest uwzględniony. Jeśli pozycja jest uwzględniona, wartość pozycji reprezentowanej przez ten gen jest dodawana do całkowitej wartości. Podobnie obliczana jest waga całkowita, aby upewnić się, że rozwiązanie jest prawidłowe. Koncepcje obliczania zgodności i sprawdzania ograniczeń można podzielić w celu wyraźniejszego oddzielenia problemów.

```
calculate_individual_fitness (individual,
                             knapsack_items,
                             knapsack_max_weight)

let total_weight equal 0
let total_value equal 0
for gene_index in range 0 to length of individual
  let current_bit equal individual[gene_index]
  if current_bit equals 1
    add weight of knapsack_items[gene_index] to total_weight
    add value of knapsack_items[gene_index] to total_value
  if total_weight is greater than knapsack_max_weight
    return value as 0 since it exceeds the weight constraint
return total_value as individual fitness
```

### Wybór rodziców na podstawie ich sprawności

Kolejnym krokiem w algorytmie genetycznym jest wybór rodziców, którzy będą produkować nowe osobniki. W teorii darwinowskiej osoby bardziej sprawne mają większe prawdopodobieństwo reprodukcji niż inne, ponieważ zazwyczaj żyją dłużej. Ponadto osoby te posiadają pożądane cechy do dziedziczenia ze względu na ich lepsze wyniki w ich środowisku. To powiedziawszy, niektóre osobniki mogą się rozmnażać, nawet jeśli nie są najsilniejsze w całej grupie, a osoby te mogą mieć silne cechy, nawet jeśli nie są silne w całości. Każda osoba ma obliczoną sprawność, która służy do określenia prawdopodobieństwa, że zostanie wybrana na rodzica nowej osoby. Ten atrybut sprawia, że algorytm genetyczny jest stochastyczny w naturze. Popularną techniką wyboru rodziców na podstawie ich sprawności jest Wybór koła ruletki. Ta strategia zapewnia różnym osobom porcje koła na podstawie ich sprawności. Koło jest „kręco” i wybrana jest osoba. Wyższa sprawność daje osobie większy kawałek koła. Ten proces powtarza się, aż osiągnięta zostanie liczba pożądanych rodziców. Obliczając prawdopodobieństwo 16 osobników o różnej sprawności, koło przydziela każdemu kawałek. Ponieważ wiele osób zachowuje się podobnie, istnieje wiele plasterków o podobnej wielkości. Liczba rodziców wybranych do wykorzystania w celu reprodukcji nowego potomstwa jest określana na podstawie planowanej całkowitej liczby potrzebnego potomstwa, która jest określana przez pożądaną wielkość populacji dla każdego pokolenia. Wybiera się dwoje rodziców i tworzy się potomstwo. Ten proces powtarza się z wybranymi różnymi rodzicami (z szansą, że ta sama osoba będzie rodzicem więcej niż jeden raz), aż zostanie wygenerowana żądana liczba potomstwa. Dwójka rodziców może rozmnażać jedno dziecko mieszane lub dwoje dzieci mieszanych. Ta koncepcja zostanie wyjaśniona w dalszej części tego rozdziału. W naszym przykładzie Problemu z plecakiem osoby o większej sprawności to te,

które wypełniają worek o największej łącznej wartości przy jednoczesnym przestrzeganiu ograniczenia wagi. Modele populacyjne to sposoby kontrolowania różnorodności populacji, stan ustalony i pokoleniowy to dwa modele populacyjne, które mają swoje zalety i wady.

### **Stan stacjonarny: zastępowanie części populacji w każdym pokoleniu**

To wysokopoziomowe podejście do zarządzania populacją nie jest alternatywą dla innych strategii selekcji, ale schematem, który je wykorzystuje. Chodzi o to, że większość populacji zostaje zachowana, a niewielka grupa słabszych osobników jest usuwana i zastępowana nowym potomstwem. Proces ten naśladuje cykl życia i śmierci, w którym słabsze jednostki umierają, a nowe osobniki powstają poprzez rozmnażanie. Gdyby w populacji było 100 osobników, część populacji stanowiłyby istniejące osobniki, a mniejsza część to nowe osobniki stworzone w drodze reprodukcji. Może być 80 osób z obecnego pokolenia i 20 nowych osób.

### **Pokoleniowe: zastępowanie całej populacji w każdym pokoleniu**

To wysokopoziomowe podejście do zarządzania populacją jest podobne do modelu stanu ustalonego, ale nie jest alternatywą dla strategii selekcji. Model pokoleniowy tworzy potomstwo równej wielkości populacji i zastępuje całą populację nowym potomstwem. Gdyby w populacji było 100 osobników, każde pokolenie dałoby 100 nowych osobników poprzez rozmnażanie.

Steady State i Generational to nadrzędne pomysły dotyczące projektowania konfiguracji algorytmu.

### **Koło ruletki: wybór rodziców i osób, które przeżyły**

Chromosomy z wyższymi wynikami sprawności są bardziej prawdopodobne, że zostaną wybrane, ale chromosomy o niższych wynikach sprawności nadal mają niewielką szansę na wybranie. Termin „Wybór koła ruletki” pochodzi od koła ruletki w kasynie, które jest podzielone na plasterki. Zazwyczaj koło jest obracane, a kulka jest uwalniana do koła. Wybrany kawałek to ten, na który kulka ląduje, gdy koło przestaje się obracać. W tej analogii chromosomy są przypisane do wycinków koła. Chromosomy z wyższymi wynikami sprawności mają większe wycinki koła, a chromosomy z niższymi wynikami sprawności mają mniejsze wycinki. Chromosom jest wybierany losowo, podobnie jak kulka losowo ląduje na plasterku. Ta analogia jest przykładem selekcji probabilistycznej. Każda osoba ma szansę zostać wybrana, niezależnie od tego, czy jest ona mała, czy duża. Szansa doboru osobników wpływa na zróżnicowanie populacji i wskaźniki konwergencji.

### **Pseudo kod**

Najpierw należy określić prawdopodobieństwo wyboru dla każdej osoby. Prawdopodobieństwo to oblicza się dla każdej osoby, dzieląc jej przystosowanie przez całkowitą sprawność populacji. Ruletki można użyć wyboru koła. „Koło” jest „obracane” do momentu wybrania żądanej liczby osobników. Dla każdego wyboru obliczana jest losowa liczba dziesiętna z przedziału od 0 do 1. Jeśli kondycja danej osoby mieści się w ramach tego prawdopodobieństwa, zostaje ona wybrana. Do określenia prawdopodobieństwa każdej osoby można zastosować inne podejścia probabilistyczne, w tym odchylenie standardowe, w którym wartość jednostki jest porównywana ze średnią wartością grupy.

```

set_probabilities_of_population (population)
  let total_fitness equal the sum of fitness of the population
  for individual in population
    let the probability_of_selection of individual...
      ...equal it's fitness/total_fitness

roulette_wheel_selection (population, number_of_selections)
  let chosen_ones equal an empty array
  for selection in range 0 to number_of_selections
    let r equal a random number between 0 and 1
    for individual in population:
      if r <= individual probability_of_selection
        append individual to chosen_ones
        break
  return chosen_ones

```

### **Rozmnażanie osób od rodziców**

Kiedy wybiera się rodziców, musi nastąpić rozmnażanie, aby stworzyć nowe potomstwo od rodziców. Generalnie dwa kroki są związane z tworzeniem dzieci z dwojga rodziców. Pierwszą koncepcją to crossover, co oznacza mieszanie części chromosomu pierwszego rodzica z częścią chromosomu drugiego rodzica i odwrotnie. W wyniku tego procesu rodzi się dwoje potomstwa, które zawiera odwrócone mieszanki rodziców. Drugą koncepcją jest mutacja, która oznacza losową nieznaczną zmianę potomstwa w celu stworzenia zróżnicowania w populacji.

### **Krzyżowanie**

Krzyżowanie polega na mieszaniu genów między dwoma osobnikami w celu stworzenia jednego lub więcej osobników potomnych. Crossover inspirowany jest koncepcją reprodukcji. Osobniki będące potomstwem są częścią swoich rodziców w zależności od zastosowanej strategii krzyżowania. Zastosowane kodowanie ma duży wpływ na strategię krzyżowania.

### **Krzyżowanie jednopunktowe: Dziedziczenie jednej części po każdym z rodziców**

Wybrano jeden punkt w strukturze chromosomu. Następnie, odwołując się do dwóch rodziców, o których mowa, używana jest pierwsza część pierwszego rodzica, a druga część drugiego rodzica jest używana. Te dwie części razem tworzą nowe potomstwo. Drugie potomstwo można wytworzyć, wykorzystując pierwszą część drugiego rodzica i drugą część pierwszego rodzica. Krzyżowanie jednopunktowe ma zastosowanie do kodowania binarnego, kodowania porządkowego / permutacyjnego i kodowania wartości rzeczywistych. Te schematy kodowania zostaną omówione w dalszej części.

### **Pseudo kod**

Aby stworzyć dwa nowe osobniki potomne, tworzona jest pusta tablica do przechowywania nowych osobników. Wszystkie geny od indeksu 0 do pożądanego indeksu rodzica A są łączone ze wszystkimi genami od pożądanego indeksu do końca chromosomu rodzica B, tworząc jednego potomka. Odwrotność tworzy drugi osobnik potomstwa.

```

one_point_crossover (parent_a, parent_b, xover_point)
  let children equal empty array

  let child_1 equal genes 0 to xover_point from parent_a plus...
  ..genes xover_point to parent_b length from parent_b
  append child_1 to children

  let child_2 equal genes 0 to xover_point from parent_b plus...
  ..genes xover_point to parent_a length from parent_a
  append child_2 to children

  return children

```

### **Krzyżowanie dwupunktowe: Dziedziczenie większej liczby części od każdego z rodziców**

Wybrano dwa punkty w strukturze chromosomu; następnie, odnosząc się do dwóch omawianych rodziców, części są wybierane naprzemiennie, aby stworzyć osobne potomstwo. Ten proces jest podobny do krzyżowania jednopunktowego. Aby całkowicie opisać ten proces, potomstwo składa się z pierwszej części pierwszego rodzica, drugiej części drugiego rodzica i trzeciej części pierwszego rodzica. Pomyśl o dwupunktowej zwrotnicy jako o łączeniu tablic w celu utworzenia nowych. Ponownie, drugą osobę można stworzyć, używając odwrotnych części każdego z rodziców. Dwupunktowa zwrotnica ma zastosowanie do kodowania binarnego i kodowania wartości rzeczywistych.

### **Jednolite krzyżowanie: Dziedziczenie wielu części po każdym z rodziców**

Jednolita zwrotnica to krok dalej niż dwupunktowa zwrotnica. W krzyżowaniu jednolitym tworzona jest maska, która reprezentuje, które geny od każdego z rodziców zostaną użyte do wytworzenia potomstwa. W celu uzyskania drugiego potomstwa można zastosować proces odwrotny. Maskę można generować losowo za każdym razem, gdy tworzone jest potomstwo, aby zmaksymalizować różnorodność. Ogólnie rzecz biorąc, jednolite krzyżowanie tworzy bardziej zróżnicowane osobniki, ponieważ cechy potomstwa są zupełnie inne w porównaniu z którymkolwiek z rodziców. Jednolite krzyżowanie ma zastosowanie do kodowania binarnego i kodowania wartości rzeczywistych.

### **Mutacja**

Mutacja polega na nieznaczej zmianie osobników potomstwa, aby zachęcić do zróżnicowania populacji. Stosuje się kilka podejść do mutacji w zależności od natury problemu i metody kodowania.

Jednym z parametrów mutacji jest współczynnik mutacji - prawdopodobieństwo, że chromosom potomstwa zostanie zmutowany. Podobnie jak w przypadku organizmów żywych, niektóre chromosomy są zmutowane bardziej niż inne; potomstwo nie jest dokładną kombinacją chromosomów swoich rodziców, ale zawiera niewielkie różnice genetyczne. Mutacja może mieć kluczowe znaczenie dla wspierania różnorodności w populacji i zapobiegania utknięciu algorytmu w najlepszych lokalnych rozwiązaniach. Wysoki wskaźnik mutacji oznacza, że osoby mają duże szanse na wybranie do mutacji lub że geny w chromosomie osobnika mają duże szanse na mutację, w zależności od strategii mutacji. Wysoka mutacja oznacza większą różnorodność, ale zbyt duża różnorodność może skutkować pogorszeniem dobrych rozwiązań.

### **Mutacja ciągu bitowego dla kodowania binarnego**

W mutacji ciągu bitowego gen w chromosomie zakodowanym binarnie jest wybierany losowo i zmieniany na inną prawidłową wartość. Inne mechanizmy mutacji mają zastosowanie, gdy stosowane jest kodowanie niebinarne. Temat mechanizmów mutacji zostanie omówiony w rozdziale 5.

### Pseudo kod

Aby zmutować pojedynczy gen chromosomu danej osoby, wybiera się losowy indeks genu. Jeśli ten gen reprezentuje 1, zmień go na 0 i na odwrót.

```
mutate_individual (individual, chromosome_length)
  let random_index equal a random number between 0 and chromosome_length
  if gene at index random_index of individual is equal to 1:
    let gene at index random_index of individual equal 0
  else:
    let gene at index random_index of individual equal 1
  return individual
```

### Mutacja Flip-bit dla kodowania binarnego

W mutacji typu flip-bit wszystkie geny w chromosomie kodowanym binarnie są odwrócone do wartości przeciwnej. Tam, gdzie były jedynki, są zerami, a gdzie były 0, to jedynki. Ten typ mutacji może radykalnie pogorszyć dobrze działające rozwiązania i jest zwykle stosowany, gdy konieczne jest ciągle wprowadzanie różnorodności do populacji.

### Zapełnianie kolejnego pokolenia

Po zmierzeniu kondycji osobników w populacji i rozmnożeniu potomstwa, następnym krokiem jest wybór osobników, które będą żyły w następnym pokoleniu. Wielkość populacji jest zwykle stała, a ponieważ więcej osobników zostało wprowadzonych przez rozmnażanie, niektóre osobniki muszą wymrzeć i zostać usunięte z populacji. Dobrym pomysłem może być wybranie najlepszych osobników, które pasują do wielkości populacji, i wyeliminowanie pozostałych. Ta strategia może jednak spowodować stagnację w różnorodności osobników, jeśli jednostki, które przeżyją, mają podobny skład genetyczny. Strategie selekcji wymienione w tej sekcji można wykorzystać do określenia osobników, które mają stanowić część populacji dla następnego pokolenia.

### Eksploracja a eksploatacja

Uruchamianie algorytmu genetycznego zawsze wiąże się ze znalezieniem równowagi między eksploracją a eksploatacją. Idealna sytuacja to taka, w której istnieje różnorodność jednostek i całej populacji, która szuka w przestrzeni poszukiwań bardzo różnych potencjalnych rozwiązań; następnie wykorzystuje się silniejsze lokalne przestrzenie rozwiązań, aby znaleźć najbardziej pożądane rozwiązanie. Piękno tej sytuacji polega na tym, że algorytm bada jak najwięcej przestrzeni wyszukiwania, wykorzystując mocne rozwiązania w miarę ewolucji jednostek.

### Warunki zatrzymania

Ponieważ algorytm genetyczny jest iteracyjny w znajdowaniu lepszych rozwiązań w każdym pokoleniu, należy ustalić warunek zatrzymania; w przeciwnym razie algorytm może działać wiecznie. Warunek zatrzymania to warunek, który jest spełniony, gdy algorytm się kończy; najsilniejszy osobnik w populacji w tym pokoleniu jest wybierany jako najlepsze rozwiązanie. Najprostszym warunkiem zatrzymania jest stała - stała wartość wskazująca liczbę pokoleń, przez które algorytm będzie działał. Inne podejście kończy się po osiągnięciu określonej sprawności. Ta metoda jest przydatna, gdy znane jest pożądane

minimalne dopasowanie, ale rozwiązanie nie jest znane. Stagnacja jest problemem w algorytmach ewolucyjnych, w których populacja dostarcza rozwiązań podobną siłę od kilku pokoleń. W przypadku stagnacji populacji prawdopodobieństwo wygenerowania solidnych rozwiązań w przyszłych pokoleniach jest niskie. Warunek zatrzymania może polegać na zmianie sprawności najlepszego osobnika w każdym pokoleniu i, jeśli kondycja zmienia się tylko nieznacznie, może zdecydować o zatrzymaniu algorytmu.

### Pseudo kod

Poszczególne etapy algorytmu genetycznego są wykorzystywane w głównej funkcji, która określa cały cykl życia. Parametry zmienne obejmują wielkość populacji, liczbę pokoleń do uruchomienia algorytmu oraz pojemność plecaka dla funkcji przystosowania, oprócz zmiennej pozycji krzyżowania i wskaźnika mutacji dla etapów krzyżowania i mutacji.

```
run_ga (population_size, number_of_generations, knapsack_capacity):
    let best_global_fitness equal 0
    let global_population equal...
    ...generate_initial_population(population_size)
    for generation in range(number_of_generations):
        let current_best_fitness equal...
        ...calculate_population_fitness(global_population, knapsack_capacity)
        if current_best_fitness is greater than best_global_fitness:
            let best_global_fitness equal current_best_fitness
        let the_chosen equal...
        ...roulette_wheel_selection(global_population, population_size)
        let the_children equal...
        ...reproduce_children(the_chosen)
        let the_children equal...
        ...mutate_children(the_children)
        let global_population equal...
        ...merge_population_and_children(global_population, the_children)
```

Jak wspomniano na początku tego rozdziału, problem plecaka można rozwiązać, stosując podejście brutalnej siły, które wymaga wygenerowania i przeanalizowania ponad 60 milionów kombinacji. Porównując algorytmy genetyczne, które mają na celu rozwiązanie tego samego problemu, możemy zobaczyć znacznie większą wydajność obliczeń, jeśli parametry eksploracji i eksploatacji są poprawnie skonfigurowane. Pamiętaj, że w niektórych przypadkach algorytm genetyczny daje „wystarczająco dobre” rozwiązanie, które niekoniecznie jest najlepszym możliwym rozwiązaniem, ale jest pożądane. Ponownie, użycie algorytmu genetycznego do rozwiązania problemu zależy od kontekstu.

Jak wspomniano na początku, problem plecaka można rozwiązać, stosując podejście brutalnej siły, które wymaga wygenerowania i przeanalizowania ponad 60 milionów kombinacji. Porównując algorytmy genetyczne, które mają na celu rozwiązanie tego samego problemu, możemy zobaczyć znacznie większą wydajność obliczeń, jeśli parametry eksploracji i eksploatacji są poprawnie skonfigurowane. Pamiętaj, że w niektórych przypadkach algorytm genetyczny daje „wystarczająco dobre” rozwiązanie, które niekoniecznie jest najlepszym możliwym rozwiązaniem, ale jest pożądane. Ponownie, użycie algorytmu genetycznego do rozwiązania problemu zależy od kontekstu.

Konfiguracja parametrów algorytmu genetycznego

Projektując i konfigurując algorytm genetyczny, należy podjąć kilka decyzji, które mają wpływ na wydajność algorytmu. Obawy dotyczące wydajności można podzielić na dwa obszary: algorytm powinien starać się dobrze znajdować dobre rozwiązania problemu, a algorytm powinien działać wydajnie z punktu widzenia obliczeń. Projektowanie algorytmu genetycznego do rozwiązania problemu byłoby bezcelowe, gdyby rozwiązanie było bardziej kosztowne obliczeniowo niż inne tradycyjne techniki. Podejście zastosowane w kodowaniu, zastosowana funkcja dopasowania i inne parametry algorytmiczne wpływają na oba typy wyników w celu uzyskania dobrego rozwiązania i obliczenia. Oto kilka parametrów do rozważenia.

- Kodowanie chromosomów - metoda kodowania chromosomów wymaga przemyślenia, aby upewnić się, że ma ona zastosowanie do problemu i że potencjalne rozwiązania dążą do osiągnięcia globalnych maksimów. Schemat kodowania jest podstawą sukcesu algorytmu.
- Wielkość populacji - wielkość populacji jest konfigurowalna. Większa populacja zachęca do większej różnorodności możliwych rozwiązań. Większe populacje wymagają jednak więcej obliczeń w każdym pokoleniu. Czasami większa populacja równoważy potrzebę mutacji, co skutkuje różnorodnością na początku, ale brakiem różnorodności w kolejnych pokoleniach. Prawidłowym podejściem jest rozpoczęcie od mniejszej populacji i zwiększenie jej w oparciu o wyniki.
- Inicjalizacja populacji - chociaż osobniki w populacji są inicjowane losowo, zapewnienie, że rozwiązania są prawidłowe, jest ważne dla optymalizacji obliczeń algorytmu genetycznego i inicjowania osobników z odpowiednimi ograniczeniami.
- Liczba potomstwa - można skonfigurować liczbę potomstwa tworzonego w każdym pokoleniu. Biorąc pod uwagę, że po rozmnażaniu część populacji jest zabijana, aby zapewnić stałą wielkość populacji, więcej potomstwa oznacza większą różnorodność, ale istnieje ryzyko, że dobre rozwiązania zostaną zabite, aby pomieścić to potomstwo. Jeśli populacja jest dynamiczna, wielkość populacji może się zmieniać po każdym pokoleniu, ale takie podejście wymaga większej liczby parametrów do skonfigurowania i kontrolowania.
- Metoda wyboru rodziców - można skonfigurować metodę wyboru używaną do wyboru rodziców. Metoda wyboru musi opierać się na problemie i pożądanej możliwości eksportu w porównaniu z możliwością wykorzystania.
- Metoda krzyżowania - Metoda krzyżowania jest powiązana z używaną metodą kodowania, ale można ją skonfigurować tak, aby zachęcać lub zniechęcać do różnorodności w populacji. Potomstwo musi nadal znaleźć ważne rozwiązanie.
- Współczynnik mutacji - Współczynnik mutacji to kolejny konfigurowalny parametr, który wywołuje większą różnorodność potomstwa i potencjalnych rozwiązań. Wyższy wskaźnik mutacji oznacza większą różnorodność, ale zbyt duża różnorodność może pogorszyć dobre wyniki u osobników. Współczynnik mutacji może zmieniać się w czasie, tworząc większą różnorodność we wcześniejszych pokoleniach, a mniej w późniejszych. Wynik ten można określić jako eksplorację na początku, a następnie eksploatację.
- Metoda mutacji - Metoda mutacji jest podobna do metody krzyżowania, ponieważ zależy od zastosowanej metody kodowania. Ważną cechą metody mutacji jest to, że musi ona nadal dawać prawidłowe rozwiązanie po modyfikacji lub przypisanej strasznej sprawności wynik.
- Metoda selekcji pokoleń - Podobnie jak metoda selekcji stosowana przy wyborze rodziców, metoda selekcji pokoleń musi wybierać osoby, które przeżyją pokolenie. W zależności od zastosowanej metody selekcji algorytm może zbiegać się zbyt szybko i stagnacja lub eksploracja zbyt długo.

- Warunek zatrzymania - warunek zatrzymania algorytmu musi mieć sens w oparciu o problem i pożądaną wynik. Złożoność obliczeniowa i czas są głównymi problemami warunku zatrzymania.

### **Przypadki użycia dla algorytmów ewolucyjnych**

Algorytmy ewolucyjne mają wiele zastosowań. Niektóre algorytmy rozwiązują pojedyncze problemy; inne łączą algorytmy ewolucyjne z innymi technikami w celu stworzenia nowatorskich podejść do rozwiązywania trudnych problemów, takich jak:

- Przewidywanie zachowań inwestorów na giełdzie - Konsumenty, którzy inwestują, każdego dnia podejmują decyzje, czy kupić więcej określonych akcji, zatrzymać to, co mają, czy też sprzedać akcje. Sekwencje tych działań można ewoluować i mapować na wyniki portfela inwestora. Instytucje finansowe mogą wykorzystać tę wiedzę do proaktywnego zapewniania cennej obsługi klienta i wskazówek.

- Wybór funkcji w uczeniu maszynowym - uczenie maszynowe omówiono w rozdziale 8, ale kluczowemu aspektowi uczenia maszynowego przypisano szereg cech dotyczących czegoś, określających, do czego jest klasyfikowany. Jeśli patrzymy na domy, możemy znaleźć wiele atrybutów związanych z domami, takich jak wiek, materiał budowlany, rozmiar, kolor i lokalizacja. Ale

przewidywać wartość rynkową, być może tylko wiek, wielkość i lokalizacja mają znaczenie. Algorytm genetyczny może odkryć izolowane cechy, które mają największe znaczenie.

- Łamanie kodu i szyfry - szyfr to wiadomość zakodowana w określony sposób, aby wyglądała jak coś innego i często jest używana do ukrywania informacji. Jeśli odbiorca nie wie, jak odszyfrować wiadomość, nie można jej zrozumieć. Algorytmy ewolucyjne mogą generować wiele możliwości zmiany zaszyfrowanej wiadomości w celu odkrycia oryginalnej wiadomości.