MSI

Weronika Głuszczak, Kacper Słowikowski, Anna Szmurło, Krzysztof Tabeau, Vladyslav Yatsenko

23 marca 2022

Spis treści

1 Wstęp

2 Algorytm genetyczny

Algorytmy genetyczne są heurystykami wykorzystywanymi w celu wyszukiwania najlepszych rozwiązań zadanego problemu spośród wszystkich możliwości. Algorytmy te inspirowane są Darwinowską teorią ewolucji. Klasyczne algorytmy genetyczne nie wykorzystują wiedzy o rozwiązywanym problemie. Wybór rozwiązania polega na odrzucaniu gorszych jednostek i wybieraniu silniejszych oraz krzyżowaniu ich, aby uzyskać potencjalnie jeszcze lepsze rozwiązania.

2.1 Podstawowe pojęcia

Wszystkie rozwiązania problemu reprezentowane są przez *osobników*. Podzbiór osobników nazywany jest *populacją*. *Genotyp* to zbiór informacji przypisany osobnikowi. Jest on podstawą do utworzenia *fenotypu*, czyli zbioru cech podlegających ocenie podczas wyboru silniejszych osobników.
Operacja krzyżowania polega na losowym przecięciu dwóch genotypów w jednym punkcie i zamianie podzielonych części między genotypami.
Operacja mutacji polega na zmianie jednej, losowo wybranej informacji z genotypu.
Populacja osobników ma stały rozmiar. W kolejnych cyklach ewolucji wszystkie genotypy są modyfikowane.
Rozwiązaniem problemu jest najlepiej przystosowany osobnik z ostatniej wygenerowanej populacji.

2.2 Ogólny opis algorytmu genetycznego

1. Losowana jest pewna populacja początkowa.
2. Dokonywana jest selekcja, a najlepiej przystosowane osobniki biorą udział w procesie reprodukcji.
3. Genotypy wybranych osobników poddawane są ewolucji:
   (a) Niektóre genotypy są krzyżowane, czyli łączone ze sobą i zastępowane nową wersją
   (b) W niektórych genotypach odbywa się mutacja, czyli wprowadza się w nich drobne losowe zmiany.
4. Osobniki oceniane są funkcją oceniającą fenotyp. Następnie najlepsze jednostki są powielane, a najsłabsze usuwane. Jeśli najlepszy osobnik z populacji nie jest wystarczająco dobry, należy wrócić do kroku drugiego.

3 Planowanie zadań

Planowanie jest techniką rozwiązywania problemów z dziedziny AI, która polega na określeniu ciągu akcji (operacji) jakie należy podjąć, aby przejść z zadanego stanu początkowego do stanu końcowego będącego celem.

3.1 Przeszukiwanie w celu generowania planu

W algorytmach przeszukiwania to funkcja heurystyczna wskazuje potencjalnie najlepsze kierunki przeszukiwania. Wybór odbywa się trochę na zasadzie „zgadywania", zatem każda dopuszczalna zmiana (kierunek) musi być przeanalizowana. Funkcja oceny heurystycznej nie pozwala wyeliminować akcji w trakcie przeszukiwania,

lecz pomaga jedynie je uporządkować. Określona akcja jest analizowana nie dlatego, że prowadzi do osiągnięcia celu, lecz dlatego, że jest dopuszczalna w danym stanie.

- Rodzaje mechanizmów generacji planów:
  - Planowanie **w przód** od stanu początkowego do stanu końcowego - propagacja stanów w przód (progresja)
  - Planowanie **w tył** od stanu końcowego do stanu początkowego - propagacja stanów wstecz (regresja)

Systemy planowania w przód mają znaczenie tylko teoretyczne. Większość praktycznych systemów planowania, to systemy planowania wstecz.

4 Opis projektu

Jako zadanie do realizacji wybraliśmy klasyczny problem w dziedzinie planowania zadań - STRIPS. Musimy znaleźć ciąg akcji który doprowadzi nas do pożądanego stanu końcowego.

4.1 STRIPS

Problem potocznie jest również nazywany światem klocków i dzieje się tak nie bez powodu.
Tłumaczą to założenia problemu:

- powierzchnia/płaszczyzna/podłoże, na którym umieszczamy klocki jest gładka i nieograniczona
- wszystkie klocki mają takie same rozmiary
- klocki mogą być umieszczone jeden na drugim
- klocki mogą tworzyć stosy
- położenie horyzontalne klocków jest nieistotne, liczy się ich wertykalne położenie względem siebie
- manipulujemy klockami tylko za pomocą ramienia robota
- w danej chwili w ramieniu robota może znajdować się tylko jeden klocek

Następnie mamy zdefiniowane predykaty oraz operatory na predykatach za pomocą których możemy zmieniać stan świata klocków. Celem problemu jest stworzenie planu lub sekwencji stosowania operatorów, aby doprowadzić świat do pewnych stanów tzn. chcemy doprowadzić aby zostały spełnione warunki definiowane na początku zadania. Zbiór operatorów z ich opisami:

- STACK(x,y): umieszczenie klocka x na klocku y; w ramieniu robota musi znajdować się klocek x, a na klocku y nie może znajdować się żaden klocek
- UNSTACK(x,y): zdjęcie klocka x z klocka y; ramię robota musi być puste/wolne a na klocku x nie może znajdować się inny klocek
- PICKUP(x): podniesienie klocka x z podłoża; ramię robota musi być puste/wolne a na klocku x nie może znajdować się inny klocek

- PUTDOWN(x): umieszczenie klocka x na podłożu; w ramieniu robota musi znajdować się klocek x

Oraz zbiór predykatów:

- ON(x,y) - spełniony, gdy klocek x znajduje się na klocku y
- ONTABLE(x) - spełniony, gdy klocek x znajduje się bezpośrednio na podłożu
- CLEAR(x) - spełniony, gdy powierzchnia klocka x jest pusta tzn. nie znajduje się na nim żaden inny klocek
- HOLDING(x) - spełniony, gdy w ramieniu robota znajduje się klocek x
- ARMEMPTY - spełniony, gdy ramię robota jest puste/wolne

Ponad to każdy operator zawiera listę predykatów, które muszą być spełnione przed jego użyciem (PRECONDITION), listę predykatów które staną się prawdziwe po jego zastosowaniu (ADD) i listę predykatów, która przestanie być prawdziwa (DELETE). Warto dodać, że predykaty które nie zawierają się w ADD i DELETE zostają niezmienione. Wiedząc to, możemy formalnie zdefiniować operatory na podstawie powyższych list:

- STACK(x,y):
  - PRECONDITION: CLEAR(y) & HOLDING(x)
  - DELETE: CLEAR(y) & HOLDING(x)
  - ADD: ARMEMPTY & ON(x,y)
- UNSTACK(x,y):
  - PRECONDITION: ON(x,y) & CLEAR(x) & ARMEMPTY
  - DELETE: ON(x,y) & ARMEMPTY
  - ADD: HOLDING(x) & CLEAR(y)
- PICKUP(x):
  - PRECONDITION: CLEAR(x) & ONTABLE(x) & ARMEMPTY
  - DELETE: ONTABLE(x) & ARMEMPTY
  - ADD: HOLDING(x)
- PUTDOWN(x):
  - PRECONDITION: HOLDING(x)
  - DELETE: HOLDING(x)
  - ADD: ONTABLE(x) & ARMEMPTY

4.2 Postać genotypu

Każdy osobnik jest kandydatem na rozwiązanie zadania, w naszym przypadku każdy osobnik reprezentuje plan - ciąg akcji. Każdy gen jest pewną akcją. Akcja jest reprezentowana przez pewną liczbę naturalną - od 0 do liczby wszystkich możliwych akcji.
W klasycznym algorytmie genetycznym długość chromosomów każdego osobnika jest taka sama. Zaś w naszym przypadku długość planu może być różna - zatem musimy założyć że długości chromosomów różnych osobników mogą być różne.
Zatem genotyp każdego osobnika będzie się składał z jednego chromosomu - ciągu akcji, niekoniecznie stałej długości.

4.3 Funkcja jakości

Funkcja jakości mierzy jakość chromosomu. Przede wszystkim porównuje stan końcowy planu z pożądanym stanem końcowym. Ale nie każdy ciąg akcji jest poprawnym planem i prowadzi do pewnego stanu - żeby akcja była wykonywalna, obecny stan powinien spełniać pewne kryteria.
Załóżmy chromosom ma wartość $g_1, g_2, \ldots, g_n$. Możliwa jest sytuacja że akcja g_1 nie jest wykonywalna ze stanu początkowego, ale g_2 już tak. g_3 nie jest wykonywalna ze stanu osiągniętego po wykonaniu akcji g_2, ale g_4 jest. Można nie pozwalać na takie sytuacje, ale to powoduje znaczne skomplikowanie podstawowych operacji genetycznych. Zatem przyjęliśmy inny model. Dla każdego chromosomu reprezentującego plan definiujemy wektor binarny o takiej samej długości. Rozważmy pierwszą niezerową liczbę w nim - załóżmy że znajduje się ona na pozycji i. Będzie to znaczyło że wszystkie akcje o numerach $1 \ldots i-1$ w chromosomie nie są wykonywalne ze stanu początkowego, ale akcja pod numerem i jest. Załóżmy teraz że następny niezerowy element znajduje się na pozycji j - to oznacza że wszystkie akcje o numerach $i+1 \ldots j-1$ nie są wykonywalne ze stanu osiągniętego po wykonaniu akcji i, ale akcja o numerze j jest wykonywalna. W taki sposób na miejscach gdzie wartości są niezerowe będziemy mieli tylko wykonywalne akcje, zatem będą one tworzyły poprawny plan.
Przydatne będzie również zdefiniowanie kolejnego wektora binarnego, tworzonego w ten sam sposób co poprzedni, ale licząc akcje od tyłu zaczynając od pożądanego stanu końcowego - w wektorze wartość 1 będzie znaczyła że kolejny stan jest osiągalny przez wykonanie akcji.
Więc nasza funkcja jakości będzie złożeniem parametrów:

1. Liczba wykonywalnych akcji do przodu zaczynając od stanu początkowego - czyli liczba jedynek w pierwszym wektorze binarnym.

2. Liczba wykonywalnych akcji do tyłu zaczynając od pożądanego końcowego - czyli liczba jedynek w drugim wektorze binarnym.

3. Podobieństwo stanu osiągniętego wykonywalnymi akcjami do przodu do pożądanego stanu końcowego.

4. Podobieństwo stanu osiągniętego wykonywalnymi akcjami do tyłu do stanu początkowego.

Taka funkcja jakości pozwoli na skuteczne, z punktu widzenia jakości potomstwa, krzyżowanie osobników które dobrze sobie radzą na początku bądź na końcu planu.

4.4 Operatory genetyczne

4.4.1 Krzyżowanie

Jest zwykłym krzyżowaniem jednopunktowym z wyborem losowego punktu z klasycznego algorytmu genetycznego.

4.4.2 Mutacja

Jest taka sama jak w klasycznym algorytmie genetycznym, czyli z pewnym prawdopodobieństwem zmienia wartość losowego genu na losową wartość.

4.4.3 Kompresja

Dodatkowo wprowadzamy trzeci operator - kompresja. Polega on na tym, że z chromosomu są usuwane wszystkie niewykonywalne akcje, czyli wszystkie akcje dla których wartością w odpowiednim wektorze binarnym jest 0.

4.5 Selekcja

Na razie zakładamy selekcję elitarną - w nowej populacji zostają najlepsze osobniki z poprzedniej generacji oraz nowe potomstwo otrzymane w wyniku stosowania operatorów genetycznych.

5 Przykłady działania wybranych elementów algorytmu

5.1 Operator kompresji

Załóżmy że chromosom $\{5, 17, 2, 11, 12\}$ został wybrany przez algorytm jako chromosom na którym trzeba wykonać operator kompresji. Załóżmy również że dla danego zadania wektor wykonywalnośći tego chromosomu jest równy $\{1, 0, 1, 1, 0\}$. Zatem po zaaplikowaniu operatora kompresji w populacji zamiast pierwotnego chromosomu pojawi się chromosom $\{5, 2, 11\}$.

5.2 Operator mutacji

Załóżmy że chromosom $\{5, 17, 2, 11, 12\}$ został wybrany przez algorytm jako chromosom na którym trzeba wykonać operator mutacji. W takim przypadku jest losowane miejsce w którym należy zmienić wartość genu oraz na co należy ją zmienić. Załóżmy że zostało wylosowane miejsce o numerze 3 (numerujemy od 0) i wartość 17. Wtedy po wykonaniu operatora mutacji chromosom będzie miał postać $\{5, 17, 2, 17, 12\}$.

5.3 Operator krzyżowania

Załóżmy że do krzyżowania zostały wybrane chromosomy $\{5, 17, 2, 11, 12\}$ oraz $\{9, 8, 3, 15\}$. Krzyżowanie jest jednopunktowe, zatem następnie jest losowany jeden punkt krzyżowania - załóżmy że wylosowano wartość 3. To znaczy że z każdego chromosomu w parze są brane pierwsze 3 geny, a końcówki są zamieniane miejscami - czyli z wybranej pary chromosomów utworzy się kolejna para $\{5, 17, 2, 15\}$ i $\{9, 8, 3, 11, 12\}$.