Algorytm 10 Algorytm genetyczny

```
1: procedura ALGORYTMGENETYCZNY(T)                    ▷ max. liczba generacji: T
2:     wygeneruj populację początkową
3:     dla t = 1,...,T wykonaj                        ▷ sprawdzenie warunków zatrzymania
4:         ocena przystosowania chromosomów w populacji   ▷ funkcja przystosowania
5:         selekcja chromosomów                           ▷ funkcja selekcji
6:         zastosowanie operatorów genetycznych           ▷ krzyżowanie i mutacja
7:         utworzenie nowej populacji
8:     zwróć „najlepszy" chromosom                    ▷ wyprowadznie „najlepszego" rozwiązania
```

4.1.1 Wybór populacji początkowej

Polega na losowym lub deterministycznym wyborze danej liczby osobników o określonej długości. W klasycznej wersji algorytmu rozmiar populacji jest stały. Niezmienna jest również długość chromosomów, która jest ściśle związana z problemem, dla którego została wygenerowana. Współcześnie stosowane są także inne warianty kształtu populacji i dopuszczają one miedzy innymi: zmienną liczbę osobników w populacji, podział populacji na grupy. Pojedynczy chromosom może zawierać informacje zakodowane:

- liczbami binarnymi, np.: $(1,1,0,1,0,0,1)$,
- liczbami rzeczywistymi, np.: $(1.23,1.11,4.26,9,9.04)$,
- liczbami całkowitymi, np.: $(2,3,1,4,5,14)$,
- symbolami z dowolnego alfabetu (w praktyce sprowadza się do kodowania liczbami całkowitymi).

W niniejszym skrypcie przybliżymy jedynie problemy oraz operatory genetyczne używające kodowania liczbami binarnymi i całkowitymi, przy zastrzeżeniu, że przy kodowaniu liczbami całkowitymi wartości poszczególnych genów nie mogą się powtarzać w danym chromosomie (stanowią permutację n liczb).

4.1.2 Sprawdzenie warunków zatrzymania

Podstawowym warunkiem zatrzymania się klasycznego algorytmu genetycznego jest osiągnięcie maksymalnej liczby generacji (iteracji). Jednocześnie lub alternatywnie można stosować inne warunki stopu, takie jak:

- upłynięcie określonego czasu od momentu rozpoczęcia działania algorytmu,
- brak poprawy rozwiązania w czasie działania algorytmu przez dany okres (liczbę iteracji),
- osiągnięcie zadanej z góry wartości funkcji przystosowania.

Ostatni warunek ma zastosowanie w zadaniach optymalizacji, gdy znana jest maksymalna (lub minimalna) wartość funkcji przystosowania.

4.1.3 Ocena przystosowania chromosomów w populacji

Funkcja przystosowania ocenia, na ile dobrze dany osobnik jest przystosowany (z punktu widzenia rozwiązywanego problemu). Jest ona ściśle powiązana z problemem, dla którego została zaprojektowana i musi zostać obliczona dla każdego rozwiązania w populacji (dla każdego chromosomu).

4.1.4 Selekcja chromosomów

Metoda selekcji ma za zadanie wybranie najlepiej przystosowanych osobników przy jednoczesnym zachowaniu różnorodności genetycznej populacji. Chromosomy, które zostaną wybrane, będą brały udział w tworzeniu nowej populacji. Wybór zawsze odbywa się zgodnie z zasadą naturalnej selekcji, czyli największe szanse na udział w tworzeniu nowych osobników mają chromosomy o największej wartości funkcji przystosowania. Ze względu na ograniczenie, że rozmiar populacji jest zawsze stały, w wyniku selekcji dany osobnik może zostać wybrany więcej niż jeden raz.

Selekcja koła ruletki

Polega na budowaniu wirtualnego koła, którego wycinki odpowiadają poszczególnym osobnikom. Rozmiar wycinków zależy od wartości funkcji oceny. Każdemu osobnikowi przydzielany jest obszar koła ruletki proporcjonalny do wartości funkcji przystosowania danego chromosomu. Tym samym im lepszy osobnik, tym większy wycinek koła zajmuje, a tym samym ma większe prawdopodobieństwo na zostanie wylosowanym. Zajętość koła dla chromosomu x_i w populacji złożonej z m osobników definiujemy z następującego wzoru:

$$p(x_i) = \frac{f(x_i)}{\sum_{j=1}^{m} f(x_j)}. \tag{4.1}$$

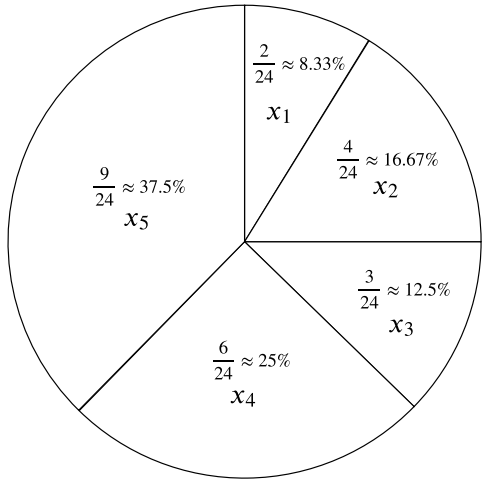
Dla przykładu, mając 5 chromosomów o następujących wartościach funkcji przystosowania:

$$f(x_1) = 2, \quad f(x_2) = 4, \quad f(x_3) = 3, \quad f(x_4) = 6, \quad f(x_5) = 9,$$

otrzymujemy następujące prawdopodobieństwo wylosowania poszczególnych chromosomów:

$$p(x_1) = \frac{2}{24}, \quad p(x_2) = \frac{4}{24}, \quad p(x_3) = \frac{3}{24}, \quad p(x_4) = \frac{6}{24}, \quad p(x_5) = \frac{9}{24}.$$

Selekcja chromosomu, może zostać zwizualizowana jako obrót kołem ruletki. Koło ruletki zbudowane dla powyższego przykładu zostało pokazane na Rys. 4.1.



Rys. 4.1: Koło ruletki — przykładowy podział (źródło: *opracowanie własne*).

Selekcja turniejowa (turniej o rozmiarze k)

Polega na wylosowaniu bez powtórzeń grupy k-elementowej z populacji, a następnie wyborze osobnika o najlepszym przystosowaniu. Całą operację powtarzamy tyle razy, ile jest osobników w populacji. Podczas wyboru najlepszego osobnika możliwe są dwie strategie: wybór deterministyczny lub losowy. Przy wyborze deterministycznym wyboru zawsze dokonujemy z prawdopodobieństwem równym 1 (zawsze wybieramy osobnika najlepiej przystosowanego), w drugim przypadku z prawdopodobieństwem mniejszym od 1. W metodzie można zmieniać wielkość k w trakcie działania algorytmu.

Selekcja rankingowa

Zwana inaczej selekcją rangową. Osobniki ustawiane są kolejno zgodnie z wartością funkcji przystosowania. Rangi zwykle ustala się jako kolejne liczby naturalne, czyli $\{1,2,\ldots,m\}$, gdzie m reprezentuje rangę najlepszego osobnika (o największej wartości funkcji przystosowania). Każdemu osobnikowi przydzielane jest prawdopodobieństwo wylosowania proporcjonalne do jego rangi, które definiujemy według następującego wzoru:

$$p(x_i) = \frac{rank(x_i)}{\sum_{j=1}^{m} rank(x_j)}. \tag{4.2}$$

Sumę rang z mianownika można obliczyć jako sumę ciągu arytmetycznego równą: $m(m+1)/2$.

4.1.5 Zastosowanie operatorów genetycznych

Operatory genetyczne mają na celu rekombinację genów w chromosomach. Wyróżniamy dwa operatory genetyczne: krzyżowanie i mutację. Każdy z operatorów genetycznych wykonywany jest z pewnym prawdopodobieństwem, które definiujemy na początku programu, i które jest stałe w trakcie jego trwania. Typowe zakresy prawdopodobieństw dla poszczególnych operatorów genetycznych są następujące:

- 0.5–1.0 dla krzyżowania,
- 0.0–0.1 dla mutacji.

W algorytmie genetycznym kolejność zastosowania obu operatorów nie ma znaczenia — mutacji można dokonać na pokoleniu rodziców przed krzyżowaniem lub na pokoleniu potomków po krzyżowaniu.

Operator krzyżowania

Krzyżowanie jest operacją mającą na celu wymianę materiału genetycznego między osobnikami. W procesie krzyżowania dzielimy całą populację na pary. Następnie dla każdej z par dokonujemy krzyżowania z zadeklarowanym wcześniej prawdopodobieństwem. W przypadku gdy nie jest stosowany operator krzyżowania, wartości genów rodziców są bezpośrednio kopiowane do potomków. Poniżej zostanie przybliżonych kilka wybranych operatorów krzyżowania.

Krzyżowanie jednopunktowe (ang. *1-point crossover*) (1-PX)

Krzyżowanie to polega na wylosowaniu jednego punktu krzyżowania, a następnie wymianie materiału genetycznego, która dokonuje się w następujący sposób:

Pokolenie rodziców:		Pokolenie potomków:
$x_1 = (\mathbf{00110}\vert\mathbf{011})$	$\xrightarrow{\text{krzyżowanie}}$	$(\mathbf{00110}\vert 101)$
$x_2 = (01101\vert 101)$		$(01101\vert\mathbf{011})$

Krzyżowanie wielopunktowe (ang. *multi-point crossover*) (k-PX)

Krzyżowanie wielopunktowe jest wykonywane analogicznie do krzyżowania jednopunktowego, z tą różnicą, że losowanych jest więcej punktów krzyżowania. Poniżej znajduje się przykład krzyżowania dwupunktowego (2-PX):

Pokolenie rodziców:		Pokolenie potomków:
$x_1 = (\mathbf{001}\|\mathbf{10}\|\mathbf{011})$	$\xrightarrow{\text{krzyżowanie}}$	$(011\|\mathbf{10}\|101)$
$x_2 = (011\|01\|101)$		$(\mathbf{001}\|01\|\mathbf{011})$

W przypadku rozwiązań kodowanych liczbami całkowitymi, w których wartości poszczególnych genów nie mogą się powtarzać w danym chromosomie (np. stanowią permutację n liczb), nie można zastosować powyższych operatorów. Doprowadziłby to do sytuacji, w której w danym chromosomie dwa różne geny miałyby tę samą wartość. Kodowanie tego typu ma miejsce np. w problemie komiwojażera, gdzie wartość każdego genu stanowi numer miasta. Przy powyższym typie chromosomu należy stosować operatory takie jak krzyżowanie: z zachowaniem porządku, z częściowym odwzorowaniem, cykliczne.

Krzyżowanie z częściowym odwzorowaniem (ang. *partially-mapped crossover*) (PMX)

Proces krzyżowania rozpoczyna się od wylosowania dwóch punktów krzyżowania i przekopiowania części genów [13]:

$x_1 = (\mathbf{58}\|\mathbf{213}\|\mathbf{764})$	$\xrightarrow{\text{krzyżowanie}}$	$(\ -\ -\ \|\mathbf{213}\|\ -\ -\ -\ )$
$x_2 = (78\|462\|531)$		

W analogicznym segmencie w x_2 znajdują się elementy, które nie zostały skopiowane. Należy je umieścić w potomku. Po wylosowaniu dwóch punktów krzyżowania posiadamy następujące odwzorowania: $4 \to 2, 6 \to 1, 2 \to 3$. Dla przykładu $6 \to 1$ oznacza, że należy skopiować 6 w miejsce 1:

$x_1 = (\mathbf{58}\|\mathbf{213}\|\mathbf{764})$	$\xrightarrow[6\to1]{\text{krzyżowanie}}$	$(\ -\ -\ \|\mathbf{213}\|\ -\ -\ 6)$
$x_2 = (78\|462\|531)$		

Podobnej operacji nie możemy wykonać dla $4 \to 2$ gdyż **2** zostało już skopiowane. Dlatego dalej sprawdzamy, który gen został skopiowany na miejsce 4. Dokonujemy następującego przejścia $4 \to 2 \to 3$, czyli kopiujemy 4 w miejsce 3:

$x_1 = (\mathbf{58}\|\mathbf{213}\|\mathbf{764})$	$\xrightarrow[4\to2\to3]{\text{krzyżowanie}}$	$(\ -\ -\ \|\mathbf{213}\|\ -\ 46)$
$x_2 = (78\|462\|531)$		

Pozostałe geny kopiujemy z x_2:

$x_1 = (\mathbf{58}\|\mathbf{213}\|\mathbf{764})$	$\xrightarrow{\text{krzyżowanie}}$	$(78\|\mathbf{213}\|546)$
$x_2 = (78\|462\|531)$		

Drugiego potomka generujemy zamieniając x_1 i x_2 miejscami:

Pokolenie rodziców:		Pokolenie potomków:
$x_1 = (\mathbf{58}\vert\mathbf{213}\vert\mathbf{764})$	$\xrightarrow{\text{krzyżowanie}}$	$(78\vert\mathbf{213}\vert546)$
$x_2 = (78\vert462\vert531)$		$(\mathbf{58}\vert462\vert\mathbf{713})$

Krzyżowanie z zachowaniem porządku (ang. *order crossover*) (OX)

Podobnie jak w przypadku poprzednim losowane są dwa punktu w chromosomach rodziców [9]. Do potomków przepisywane są fragmenty pomiędzy wylosowanymi punktami. Puste fragmenty uzupełniane są począwszy od drugiego punktu krzyżowania elementami z drugiego rodzica, które jeszcze nie są obecne w potomku. Rozpisując przykład bardziej szczegółowo mamy:

$x_1 = (\mathbf{58}\vert\mathbf{213}\vert\mathbf{764})$	$\xrightarrow{\text{krzyżowanie}}$	$(\ -\ -\ \vert\mathbf{213}\vert\ -\ -\ -\ )$
$x_2 = (78\vert462\vert531)$		$(\ -\ -\ \vert462\vert\ -\ -\ -\ )$

Następnie geny z chromosomu x_2 układamy rozpoczynając od drugiego wylosowanego punktu (53178462), a następnie wykreślamy z niego elementy, które już znajdują się w pierwszym potomku (**213**) i otrzymujemy (57846). Podobnie postępujemy z genami z chromosomu x_1 (**76458213**), z których pomijamy geny znajdujące się w drugim potomku (462) co daje nam (**75813**). Tak wygenerowane ciągi genów wpisujemy do odpowiednich potomków rozpoczynając od drugiego punktu krzyżowania, w efekcie uzyskując:

Pokolenie rodziców:		Pokolenie potomków:
$x_1 = (\mathbf{58}\vert\mathbf{213}\vert\mathbf{764})$	$\xrightarrow{\text{krzyżowanie}}$	$(46\vert\mathbf{213}\vert578)$
$x_2 = (78\vert462\vert531)$		$(\mathbf{13}\vert462\vert\mathbf{758})$

Krzyżowanie cykliczne (ang. *cycle crossover*) (CX)

Ten rodzaj krzyżowania kreuje potomków w ten sposób, że każdy gen wraz z jego miejscem pochodzi od jednego z rodziców [35]. Na początek losujemy gen i rodzica, od którego zaczynamy krzyżowanie. Powiedzmy, że zaczniemy od **5** w pierwszym rodzicu x_1:

$x_1 = (\mathbf{58213764})$	$\xrightarrow{\text{krzyżowanie}}$	$(\mathbf{5}\ -\ -\ -\ -\ -\ -\ -\ )$
$x_2 = (78462315)$		

5 leży na tym samym miejscu co 7, więc do chromosomu dodajmy **7** w miejscu z pierwszego chromosomu:

$$x_1 = (\mathbf{58213764}) \qquad x_2 = (78462315) \qquad \xrightarrow{\text{krzyżowanie}} \qquad (\mathbf{5}\text{ - - - - }\mathbf{7}\text{ - - })$$

7 leży na tym samym miejscu co 3, dlatego do potomka dodajemy **3** w miejscu z pierwszego chromosomu:

$$x_1 = (\mathbf{58213764}) \qquad x_2 = (78462315) \qquad \xrightarrow{\text{krzyżowanie}} \qquad (\mathbf{5}\text{ - - - }\mathbf{37}\text{ - - })$$

3 leży na tym samym miejscu co 2, zatem do chromosomu dodajemy **2**. Całość kontynuujemy aż do momentu zamknięcia się cyklu, czyli do natrafienia na gen, który już dodaliśmy do chromosomu (w poniższym przypadku to 5):

$$x_1 = (\mathbf{58213764}) \qquad x_2 = (78462315) \qquad \xrightarrow{\text{krzyżowanie}} \qquad (\mathbf{5}\text{ - }\mathbf{2}\text{ - }\mathbf{37}\text{ - }\mathbf{4})$$

Resztę genów uzupełniamy genami z chromosomu x_2:

$$x_1 = (\mathbf{58213764}) \qquad x_2 = (78462315) \qquad \xrightarrow{\text{krzyżowanie}} \qquad (\mathbf{5}8\mathbf{2}6\mathbf{3}7\mathbf{1}\mathbf{4})$$

Dla drugiego potomka uzupełnianie zaczynamy od 7. A wolne miejsca pozostałe po zakończeniu cyklu uzupełniamy z chromosomu x_1:

Pokolenie rodziców:		Pokolenie potomków:
$x_1 = (\mathbf{58213764})$	$\xrightarrow{\text{krzyżowanie}}$	$(\mathbf{5}8\mathbf{2}6\mathbf{3}7\mathbf{1}\mathbf{4})$
$x_2 = (78462315)$		$(7\mathbf{8}4\mathbf{1}23\mathbf{6}5)$

Operator mutacji

Mutacja ma za zadanie wprowadzić różnorodność genetyczną populacji. Dla chromosomów kodujących informację binarnie mutacja polega na zamianie wartości bitu na przeciwny:

$$(00110010101) \qquad \xrightarrow{\text{mutacja}} \qquad (00110011101)$$

Prawdopodobieństwo dokonania mutacji można zastosować w dwóch wariantach. W pierwszym wariancie prawdopodobieństwo mutacji jest losowane dla całego

chromosomu. W sytuacji gdy mutacja ma mieć miejsce, losuje się miejsce mutacji. W drugim wariancie prawdopodobieństwo mutacji jest losowane osobno dla każdego genu w każdym chromosomie.

Podobnie jak w przypadku krzyżowania, tak samo w przypadku mutacji dla problemów kodowanych liczbami całkowitymi nie można zastosować powyższego operatora. Alternatywnym podejściem jest zastosowanie **mutacji poprzez inwersję**. Inwersja polega na wylosowaniu podzakresu w chromosomie i wstawieniu genów w tym podzakresie w kolejności przeciwnej:

$$(12\mathbf{3456}789) \xrightarrow{\text{mutacja}} (12\mathbf{6543}789)$$

Alternatywnie można wybrać dany zakres genów w chromosomie (lub wybrać nawet pojedyncze geny) i dokonać permutacji znajdujących się w nich wartości:

$$(12\mathbf{3456}789) \xrightarrow{\text{mutacja}} (12\mathbf{5364}789)$$

4.1.6 Utworzenie nowej populacji

Chromosomy otrzymane w wyniku działania operatorów genetycznych stanowią nową populację. W następnej iteracji nowa populacja staje się populacją bieżącą i to właśnie na niej będzie pracował algorytm genetyczny opisany w poprzednich akapitach. W klasycznym algorytmie genetycznym nowa populacja zawsze jest tak samo liczna, jak poprzednia.

4.1.7 Wyprowadzenie „najlepszego" chromosomu

Po zatrzymaniu algorytmu należy zwrócić wynik jego działania. Wynikiem jest najlepiej przystosowany chromosom, czyli taki, w którym wartości funkcji przystosowania jest największa. Istnieją tutaj dwa podejścia wyboru najlepszego chromosomu: najlepszy z ostatniego pokolenia oraz najlepszy w całej historii. Te dwa pojęcia wcale nie muszą być sobie równoważne, ponieważ najlepsze rozwiązanie w całej historii, może zostać utracone w wyniku działania algorytmu. Czasami aby zapobiegać sytuacji tego typu istnieje opcja „elitaryzmu", która w każdej iteracji zastępuje najsłabszy chromosom najlepszym, aby go nie utracić.

4.2 Przykładowe problemy

Dla lepszego zrozumienia tematu poniżej zostaną przybliżone dwa klasyczne przykłady, w których postawione problemy rozwiązywane są przy pomocy algorytmu genetycznego: dyskretny problem plecakowy i problem komiwojażera.

4.2.1 Dyskretny problem plecakowy

Dyskretny problem plecakowy często przedstawia się jako problem złodzieja rabującego sklep — znalazł on n towarów; i-ty przedmiot jest wart v_i oraz posiada określoną objętość c_i. Złodziej dąży do zabrania ze sobą jak najbardziej wartościowszego łupu, przy czym nie może przekroczyć objętości plecaka C. Przedmiotów nie można dzielić (dzielenie byłoby możliwe w ciągłym problemie plecakowym).

Formalna definicja problemu jest następująca. Do dyspozycji posiadamy n przedmiotów, a każdy z nich jest opisany jako para (v_i, c_i):

$$A = \{(v_i, c_i)\}_{i=1\ldots n}. \tag{4.3}$$

Zadaniem jest znalezienie takiego podzbioru B, w którym wartość przedmiotów jest maksymalna, ale ich objętość nie przekracza maksymalnej objętości plecaka, czyli:

$$B \subseteq A, \qquad \sum_{(v_i,c_i)\in B} v_i \longrightarrow \max, \qquad \sum_{(v_i,c_i)\in B} c_i \leqslant C. \tag{4.4}$$

Na potrzeby niniejszego zadania użyjemy chromosomów binarnych, w których każdy z genów może przyjmować tylko jedną wartość 0 lub 1.

Każdy chromosom będzie reprezentowała jedno możliwe rozwiązanie tj. pewien podzbiór przedmiotów, które mogą zostać włożone do plecaka. Zatem długość chromosomu będzie równa liczbie wszystkich przedmiotów n. Pojedynczy gen będzie definiował, czy dany przedmiot znajduje się w plecaku (wartość genu 1) czy też danego przedmiotu w tym plecaku nie ma (wartość genu 0). Przykładowy chromosom, dla zadania w którym występuje 15 przedmiotów, prezentuje diagram poniżej. Chromosom koduje informacje, o tym że w plecaku znajdują się przedmioty: $1, 2, 6, 8, 14$.

nr przedmiotu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
chromosom	1	1	0	0	0	1	0	1	0	0	0	0	0	1	0

Funkcja przystosowania dla dyskretnego problemu plecakowego będzie sumą wartości przedmiotów, jeżeli ich sumaryczna objętości nie przekracza maksymalnej objętości plecaka C, w przeciwnym razie będzie to 0. Matematycznie funkcję przystosowania dla danego osobnika można zapisać jako:

$$p(B) = \begin{cases} \sum_{(v_i,c_i)\in B} v_i, & \sum_{(v_i,c_i)\in B} c_i \leqslant C; \\ 0, & \sum_{(v_i,c_i)\in B} c_i > C. \end{cases} \tag{4.5}$$

Podczas losowania populacji początkowej należy pamiętać, żeby wszystkie chromosomy w pierwszej generacji nie zostały ocenione przez funkcje przystosowania opisaną wzorem (4.5) na 0. Taka sytuacja może mieć miejsce, kiedy geny

w populacji początkowe losujemy np. z prawdopodobieństwem 0.5. Powoduje to, że w każdym plecaku znajduje się około połowa wszystkich przedmiotów. W połączeniu z relatywnie małą wartość C ($C \ll \sum_{i=1}^{n} c_i$), funkcja przystosowania oceni każdy plecak jako przepełniony. Rozwiązaniem tego problemu może być zmniejszenie prawdopodobieństwa pojawienia się przedmiotu w plecaku w populacji początkowej. Ewentualnie należy zmodyfikować wzór (4.5), tak żeby algorytm w pierwszej kolejności starał się znaleźć taki podzbiór przedmiotów, który będzie mieścił się w plecaku:

$$p(B) = \begin{cases} \sum_{(v_i,c_i)\in B} v_i, & \sum_{(v_i,c_i)\in B} c_i \leqslant C; \\ \frac{1}{\sum_{(v_i,c_i)\in B} v_i}, & \sum_{(v_i,c_i)\in B} c_i > C. \end{cases} \tag{4.6}$$

Rozwiązanie dokładne

Dyskretny problem plecakowy może zostać rozwiązany przy pomocy programowania dynamicznego. Korzystając z indukcji staramy się odnaleźć $V_{i,j}$ — wartość najlepszego upakowania plecaka o objętości j, za pomocą przedmiotów o numerach $1,\ldots,i$. Wartość tą definiujemy następująco przy użyciu rekurencji:

$$\begin{aligned} V_{0,j} &= 0, \\ V_{i,0} &= 0, \\ V_{i,j} &= V_{i-1,j}, && \text{jeżeli} \quad c_i > j, \\ V_{i,j} &= \max(V_{i-1,j},\, V_{i-1,j-c_i} + v_i), && \text{jeżeli} \quad c_i \leqslant j. \end{aligned} \tag{4.7}$$

Przekładając powyższy wzór na pseudokod otrzymujemy Algorytm 11. Podany algorytm podaje tylko wartość najlepszego upakowania, ale nie definiuje jaki jest to podzbiór przedmiotów. W celu znalezienia podzbioru przedmiotów należałoby wprowadzić drugą tablicę tzw. wskazań wstecznych (ang. *back-pointers*) i na końcu ją prześledzić. Tego typu rozwiązanie stosuje się w wielu podejściach opartych na programowaniu dynamicznym. Złożoność obliczeniowa zaprezentowanego algorytmu wynosi $\Theta(n\,C)$. Złożoność ta pozornie wydaje się liniowa i może dojść do sytuacji kiedy C będzie skalowało się np. proporcjonalnie do 2^n, co powoduje że problem może wymagać czasu wykładniczego (dlatego zadanie jest problemem NP-trudnym).

4.2.2 Problem komiwojażera

Drugim klasycznym problemem na, przykładzie którego bardzo często tłumaczy się działanie algorytmu genetycznego, jest problem komiwojażera (ang. *travelling salesman problem*) (TSP). Problem ten definiuje się jako problem komiwojażera, który ma odwiedzić n miast i chce pokonać jak najkrótszą odległość. Jest to problem NP-trudny. W implementacji rozwiązania tego problemu należy posłużyć się

Algorytm 11 Algorytm rozwiązywania dyskretnego problemu plecakowego

```
1:  procedura AlgorytmDyskretnyProblemPlecakowy
2:      dla i = 0,...,n wykonaj
3:          V_{i,0} := 0
4:      dla j = 0,...,C wykonaj
5:          V_{0,j} := 0
6:      dla i = 0,...,n wykonaj                ▷ bierzemy pod uwagę i pierwszych przedmiotów
7:          dla j = 0,...,C wykonaj
8:              jeżeli c_i > j to              ▷ sprawdzenie czy przedmiot mieści się
9:                  V_{i,j} = V_{i-1,j}        ▷ w plecaku o rozmiarze j
10:             w przeciwnym razie
11:                 V_{i,j} = max(V_{i-1,j}, V_{i-1,j-c_i} + v_i)
12:     zwróć V_{n,C}
```

chromosomami kodowanymi liczbami całkowitymi, w których wartości poszczególnych genów nie mogą się powtarzać w danym chromosomie. Każdy chromosom będzie w istocie permutacją liczb od 1 do n reprezentującą pewną ścieżkę komiwojażera. Wartość każdego genu będzie stanowić numer miasta. Przykład takiego chromosomu znajduje się na diagramie poniżej.

5	2	14	8	4	3	10	15	6	7	13	9	12	1

Zaprezentowany chromosom definiuje kolejność, w jakiej komiwojażer odwiedzałby miasta. Wartość funkcji przystosowania można w prosty sposób zdefiniować jako sumę odległości poszczególnych połączeń pomiędzy miastami, które musi pokonać komiwojażer:

$$f(x_i) = \|x_{i,n}\, x_{i,1}\| + \sum_{j=1}^{n-1} \|x_{i,j}\, x_{i,(j+1)}\|, \tag{4.8}$$

gdzie przez $x_{i,j}$ rozumiemy j-ty gen w i-tym chromosomie, $\|\cdot\|$ jest to odległość pomiędzy dwoma miastami. Jednakże problem postawiony w ten sposób przybiera postać zadania minimalizacji. Teoretycznie w klasycznym algorytmie genetycznym zawsze powinno sprowadzić się zadanie do zadania maksymalizacji. Przykład funkcji selekcji spełniającej to kryterium w omawianym problemie jest następująca:

$$f(x_i) = 1 - \frac{\|x_{i,n}\, x_{i,1}\| + \sum_{j=1}^{n-1} \|x_{i,j}\, x_{i,(j+1)}\|}{f_{\max} - f_{\min}}, \tag{4.9}$$

gdzie $f_{\min}$ to najkrótsza, a $f_{\max}$ to najdłuższa możliwa odległość łącząca wszystkie miasta. W praktyce nie znamy wartości stałych z mianownika, ale $f_{\min}$ możemy

przyjąć jako 0 i pominąć. $f_{\max}$ możemy przyjąć jako najdłuższą odległość łączącą miasta spośród wszystkich odległości kodowanych przez chromosomy z populacji początkowej. Należy zauważyć, że część funkcji selekcji może zostać użyta również w zadaniach minimalizacji — taką funkcją jest m.in. selekcja turniejowa. W przypadku problemu komiwojażera należy stosować odpowiednie operatory genetyczne, takie jak np. krzyżowania z częściowym odwzorowaniem i mutację przez inwersję.

4.3 Ćwiczenia laboratoryjne (MATLAB)

E **Ćwiczenie 4.1** **Napisz skrypt rozwiązujący dyskretny problem plecakowy za pomocą algorytmu genetycznego.** Polecenia do wykonania:

- Napisz ogólny skrypt realizujący kroki algorytmu genetycznego. Parametrami dla skryptu powinny być m.in. rozmiar populacji, liczba iteracji, wskaźnik na funkcję przystosowania, wskaźnik na funkcję selekcji, wskaźnik na funkcję krzyżowania, wskaźnik na funkcję mutacji.
- Napisz skrypt losujący problem plecakowy dla zadanej liczby przedmiotów — n.
- Napisz funkcję obliczającą wartości funkcji przystosowania dla podanego w parametrze chromosomu reprezentującego problem plecakowy.
- Napisz funkcje realizujące: selekcję ruletkową, krzyżowanie jednopunktowe, mutację (wszystkie te funkcje powinny przyjmować na wejście całą populację).
- Dla wylosowanego problemu plecakowego przeprowadź działanie algorytmu genetycznego. W każdej iteracji odnotuj, a następnie przedstaw na wykresie:
  1. średnie przystosowanie populacji,
  2. przystosowanie najlepszego osobnika w danym pokoleniu,
  3. przystosowanie najlepszego osobnika wykrytego w dotychczasowej historii.

E **Ćwiczenie 4.2** **Napisz funkcje realizujące selekcje rankingową i turniejową oraz krzyżowanie dwupunktowe (wykorzystaj program z Ćwiczenia 4.1).** Polecenia do wykonania:

- Napisz dwie funkcje selekcyjne realizujące: selekcję rankingową i turniejową.
- Porównaj działanie selekcji koła ruletki, rankingowej i turniejowej.
- Napisz funkcję do krzyżowania dwupunktowego.
- Dla wylosowanego problemu plecakowego przeprowadź działanie algorytmu genetycznego. Przeprowadź eksperymenty numeryczne porównujące działanie poszczególnych metod selekcji (koła ruletki, turniejowej, rankingowej) oraz poszczególnych metod krzyżowania (1-PX, 2-PX). W każdej iteracji odnotuj, a następnie przedstaw na wykresie:
  1. średnie przystosowanie populacji,
  2. przystosowanie najlepszego osobnika w danym pokoleniu,
  3. przystosowanie najlepszego osobnika wykrytego w dotychczasowej historii.

E **Ćwiczenie 4.3 Porównaj rozwiązanie dyskretnego problemu plecakowego przez algorytm genetyczny z rozwiązaniem dokładnym.** Napisz skrypt rozwiązujący problem plecakowy w sposób dokładny. Bazując na programie napisanym do Ćwiczenia 4.1 dla wylosowanego problemu plecakowego (lub kilku) sprawdź, czy algorytm genetyczny zwraca ten sam wynik, co rozwiązanie dokładne.

E **Ćwiczenie 4.4 Napisz skrypt rozwiązujący problem komiwojażera za pomocą algorytmu genetycznego (wykorzystaj program z Ćwiczenia 4.1 oraz 4.2).** Polecenia do wykonania:

- Napisz skrypt do losowania problemu komiwojażera dla zadanej liczby miast — n.
- Napisz funkcję obliczającą wartość funkcji przystosowania dla podanego w parametrze chromosomu reprezentującego problem komiwojażera.
- Zaimplementuj operatory krzyżowania PMX, OX, CX oraz odpowiednią mutację dla rozwiązywanego problemu.
- Dla wylosowanego zestawu miast przeprowadź działanie algorytmu genetycznego. Przeprowadź eksperymenty numeryczne porównujące działanie poszczególnych metod krzyżowania (PMX, OX, CX). W każdej iteracji odnotuj, a następnie przedstaw na wykresie:
  1. średnie przystosowanie populacji,
  2. przystosowanie najlepszego osobnika w danym pokoleniu,
  3. przystosowanie najlepszego osobnika wykrytego w dotychczasowej historii.