1. **Portfel zleceń**
   1. **Zadanie**

Zadanie polegało na optymalizacji portfela zleceń pewnego hipotetycznego przedsiębiorstwa w taki sposób, żeby osiągnąć maksymalny zysk z oferowanych usług. Przedsiębiorstwo posiadało limit na liczbę realizacji danych usług w rozpatrywanym czasie, którego nie można było przekroczyć. Klienci końcowi przygotowali zlecenia, które składały się z wybranej liczby usług różnego typu. Aby zrealizować zlecenie, niezbędne było wykonanie wszystkich zawartych w nim usług. Należy tak dobrać liczbę wykonań poszczególnych zleceń (rysunek 1), aby osiągnąć maksymalny zysk dla firmy.

Obraz zawierający stół

Opis wygenerowany automatycznie

Rysunek 1. Lista otrzymanych zleceń.

* 1. **Sposób rozwiązania**

Do rozwiązania problemu optymalizacji wykorzystano algorytm genetyczny. Na podstawie danych z zadania definiowano funkcję celu oraz ograniczenia, które następnie zaimplementowano w środowisku programistycznym R. Do wykonania zadania wykorzystano bibliotekę **genalg**, która udostępnia metody pozwalające na implementację algorytmu genetycznego.

* 1. **Implementacja w środowisku R**

Deklaracja wektorów pomocniczych v1 i v2:



Wektory pomocnicze stanowią ograniczenia nałożone na liczbę możliwych do wykonania usług danego rodzaju. Wektor allServicesNum przechowuje limity na liczbę wykonania poszczególnych usług i jest niezmienny. Wektor currentServicesNum jest modyfikowany w trakcie działania programu w celu wychwycenia elementów algorytmu genetycznego niespełniających tego ograniczenia.

Dodatkowo wykorzystano dwa kolejne wektory i jedna macierz. xTmp służył do obliczania wartości currentServicesNum w danej iteracji algorytmu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Przedstawione powyżej pętle operują na dwuwymiarowej tablicy zMatrix, która w proponowanym rozwiązaniu reprezentuje listę otrzymanych zleceń.

Obraz zawierający tekst

Opis wygenerowany automatycznie

W wyniku wykonania zamieszczonych powyżej fragmentów kodu generowany jest wektor przechowujący aktualną liczbę możliwych do wykonania usług. Należy następnie sprawdzić czy żaden z elementów otrzymanego wektora nie jest ujemny, co oznaczałoby niespełnienie ograniczeń.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Powyższa pętla sprawdza opisane wcześniej ograniczenie, a w przypadku kiedy wykryje wartość wektora niespełniającą ograniczenia przypisuje do zwracanej wartości bardzo dużą liczbę, która powoduje odrzucenie tego elementu algorytmu genetycznego na korzyść innego, który takie ograniczenie spełnia. W momencie gdy pętla wykona się jednak do końca, elementy wektora v1 przechowują liczbę usług, które należy wykonać, na podstawie czego możemy obliczyć uzyskany zysk z zaproponowanej przez algorytm konfiguracji przyjmowania zleceń.



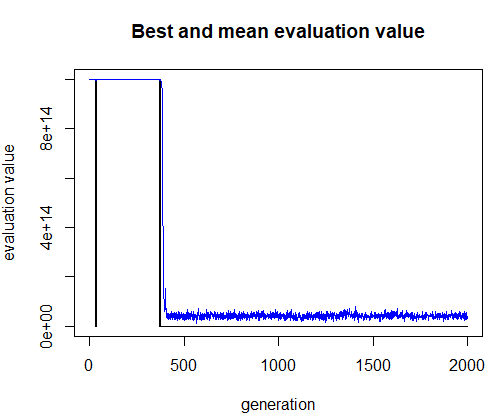
* 1. **Uzyskane wyniki**

W zależności od przyjętej na początku populacji początkowej oraz liczby iteracji otrzymywano zróżnicowane wyniki. Przykład przedstawiono poniżej.

1. Populacja = 500, liczba iteracji = 2000



Wyniki otrzymywane w kolejnych pokoleniach:



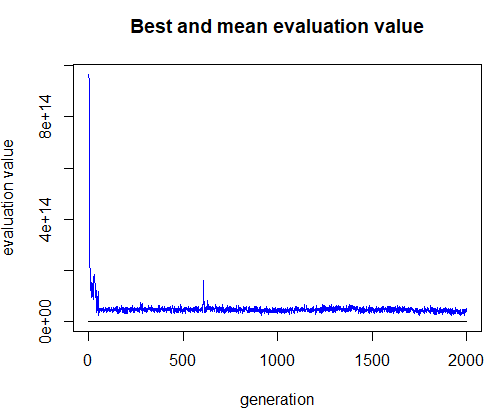
Finalnie otrzymano następujące wyniki typu *float*:



Po manualnym zaokrągleniu i przeniesieniu do programu Excel:

|  |  |  |
| --- | --- | --- |
|  | Przychód za jednorazowe wykonanie zlecenia | **Ile razy wykonano zlecenie?** |
| Zlecenie #1 | 1261 | 7 |
| Zlecenie #2 | 2975 | 4 |
| Zlecenie #3 | 2755 | 0 |
| Zlecenie #4 | 1021 | 17 |
| Zlecenie #5 | 1594 | 0 |
| Zlecenie #6 | 1769 | 1 |
| Zlecenie #7 | 1853 | 2 |
|  |  |  |
|  | **Portfel:** | 43559 |

2. Populacja = 1000, Liczba iteracji = 2000



Finalnie otrzymano następujące wyniki typu *float*:



Po manualnym zaokrągleniu i przeniesieniu do programu Excel:

|  |  |
| --- | --- |
| Przychód za jednorazowe wykonanie zlecenia | **Ile razy wykonano zlecenie?** |
| 1261 | 15 |
| 2975 | 1 |
| 2755 | 75 |
| 1021 | 12 |
| 1594 | 2 |
| 1769 | 0 |
| 1853 | 0 |
|  |  |
| **Portfel:** | 243955 |

* 1. **Wnioski**

Zrealizowane zadanie podobnie jak poprzednie pozwoliło na zapoznanie się ze środowiskiem programistycznym R. Zadanie okazało się o tyle trudniejsze, że wymagało operowania na tablicach dwuwymiarowych w celu sprawdzenia spełnialności poszczególnych ograniczeń. Prawdopodobnie można było użyć innej struktury danych dostępnej w R, jednakże słaba znajomość środowiska spowodowała pochylenie się w stronę znanych struktur danych. Niemniej jednak uważam, że praca była owocna, a uzyskane wyniki okazały się poprawne, co sprawdzono za pomocą solvera zawartego w Excel’u.

Wnioski dotyczące działania samego algorytmu genetycznego pozostają niezmienne w stosunku do tych przedstawionych w sprawozdaniu z poprzedniego zadania.