

1. Treść zadania

Przedsiębiorstwo ma możliwość wyboru dwóch asortymentów spośród czterech wyrobów do produkcji. Zysk z tytułu produkcji tych wyrobów kształtuje się następująco: I-17 zł, II-43 zł, III- 20 zł, IV- 24 zł. Do produkcji niezbędne są dwa środki s1 i s2, których zużycie jednostkowe wynosi odpowiednio: dla wyrobu I potrzeba po 1 jednostce środków, dla wyrobu II- 3 jedn. środka s1 i 2 jedn. środka s2, dla III- 1 jedn. środka s1 i 2 jedn. s2, dla IV – 1 jedn. środka s1. Środka s2 do produkcji wyrobu II potrzeba czterokrotnie więcej niż środka s1. Jaki powinien być wybór i w jakich ilościach wybrane produkty należy produkować wiedząc, że zasoby środka s1 nie mogą przekroczyć 30 jedn., a środka s2- 40 jedn.?

2. Rozwiążanie zadania

Zadanie zostało rozwiązane a pomocą metody sympleksów.

Kroki rozwiązania:

1. Zbudowanie początkowej tablicy sympleksów.
2. Wyznaczenie kolumny wprowadzanej do bazy.
 - a. W przypadku szukania maksimum – wybieramy kolumnę o największej wartości.
 - b. W przypadku szukania minimum – wybieramy kolumnę o najmniejszej wartości.
3. Wyznaczenie wiersza usuwanego z bazy.
4. Przekształcenie układu warunków ograniczających do postaci bazowej względem nowej bazy.
5. Powtarzanie kroków 2 – 4 aż do znalezienia rozwiązania optymalnego.

3. Odpowiedź/interpretacja uzyskanych wyników

Dla danych podanych w treści zadania, w przypadku znalezienia optymalnego rozwiązania został wyznaczony zysk 540 zł. Taki zysk jest osiągany w przypadku produkcji 20 jednostek wyrobu 1 i 10 jednostek wyrobu 3. W znalezionym rozwiązaniu zarówno zasób s1, jak i zasób s2 są w pełni wykorzystane – wykorzystano 30 zasobów s1 i 40 zasobów s2 (nie zostały przekroczone wartości podane w treści zadania).

Wykorzystanie s1:

$$s1 = 20 \cdot 1 + 10 \cdot 1$$

Wykorzystanie s2:

$$s2 = 20 \cdot 3 + 10 \cdot 2 = 40$$

4. Cel i zakres działania programu

Celem programu jest wyznaczenie optymalnej struktury produkcji wyrobów przy zadanych ograniczeniach zasobów. Program umożliwia ustalenie, jakie ilości poszczególnych produktów należy wytwarzać, aby osiągnąć maksymalny zysk lub minimalny koszt, z uwzględnieniem dostępnych surowców.

Program pozwala wybrać dowolną liczbę wyrobów oraz surowców (wraz z modyfikacją ich parametrów, takich jak jednostkowe zużycie zasobów oraz zysk z produkcji danego wyrobu). Możliwy jest również wybór

Program prezentuje wyniki w postaci wyniku funkcji zysku/kosztu oraz liczby jednostek każdego wyrobu, które należy produkować, aby taki zysk (albo koszt) osiągnąć.

5. Dokumentacja zewnętrzna

Instrukcja korzystania z programu krok po kroku:

1. Uruchomienie programu

Otwiera się główne okno aplikacji. Liczba wyrobów i surowców i wszystkie pozostałe dane (takie jak wartości funkcji kosztu czy ograniczenia) są zgodne z treścią zadania.

2. Zmiana istniejących danych dotyczących wyrobów i surowców

W celu zmiany zużycia surowca przez wyrob lub zysk jednostkowy należy edytować odpowiednie pole w tabeli.

W celu zmiany limitu surowca należy edytować odpowiednie pole - „Maks. zasób s1” lub „Maks. zasób s2”.

Po wprowadzeniu zmian nie trzeba nic zatwierdzać - zmiany zostaną automatycznie uwzględnione podczas wykonywania obliczeń.

3. Dodawanie nowego surowca

Należy kliknąć przycisk „Dodaj surowiec”. Pojawi się wtedy nowa kolumna w tabeli, którą należy wypełnić odpowiednimi danymi.

4. Usuwanie surowca

Istnieje możliwość usunięcia surowca. W tym celu należy kliknąć przycisk „Usuń surowiec”. Zostanie wtedy usunięty ostatnio dodany surowiec.

5. Dodawanie nowego wyrobu

Należy kliknąć przycisk „Dodaj wyrób”. Pojawi się wtedy nowy wiersz w tabeli, który należy wypełnić odpowiednimi danymi.

6. Usuwanie wyrobu

Istnieje możliwość usunięcia wyrobu. W tym celu należy kliknąć przycisk „Usuń wyrób”. Zostanie wtedy usunięty ostatnio dodany wyrob.

7. Wybór optymalizacji

Istnieje możliwość wyboru, czy interesuje nas poznanie maksymalnego zysku, czy minimalnego kosztu. W tym celu należy wybrać optymalizację „Max” albo „Min”.

8. Wyświetlenie rozwiązania

W celu wyświetlenia rozwiązania należy nacisnąć przycisk „Oblicz”. Pojawi się nowe okno prezentujące wyniki.

6. Dokumentacja wewnętrzna - opis programu, zastosowanych algorytmów, opis ciekawych miejsc w programie

Program został napisany w języku Python z wykorzystaniem bibliotek Tkinter, CustomTkinter, Matplotlib oraz Numpy. Jest on rozwiązaniem problemu ustalenia optymalnej struktury produkcji dwóch wyrobów przy ograniczeniach dotyczących użycia surowców. W celu prostszego i przyjemniejszego użycia zapewnia on użytkownikowi graficzny interfejs.

Program został napisany w języku Python z wykorzystaniem bibliotek:

- Tkinter – główny interfejs graficzny,
- NumPy – obliczenia macierzowe i wektorowe,
- itertools – generowanie kombinacji wyrobów,
- tkinter.messagebox – komunikaty o błędach i wynikach.

Aplikacja umożliwia rozwiązanie liniowego problemu optymalizacji produkcji przy ograniczeniach surowcowych.

```
if opt_type == "max":  
    pivot_col_idx = np.argmax(c_j_minus_z_j)  
else:  
    pivot_col_idx = np.argmin(c_j_minus_z_j)  
  
pivot_column = constraints_matrix[:, pivot_col_idx]
```

Wyznaczenie kolumny wprowadzanej do bazy różni się w zależności od typu optymalizacji. W przypadku szukania maksimum – wybieramy kolumnę o największej wartości. W przypadku szukania minimum – wybieramy kolumnę o najmniejszej wartości.

```
ratios = np.full(num_resources, np.inf)  
for i in range(num_resources):  
    if pivot_column[i] > 0:  
        ratios[i] = rhs_b[i] / pivot_column[i]  
  
pivot_row_idx = np.argmin(ratios)  
pivot_element = constraints_matrix[pivot_row_idx, pivot_col_idx]
```

Wyznaczanie wiersza usuwanego z bazy odbywa się z użyciem testu ratio. Oznacza to, że minimalny stosunek wyznacza wiersz wychodzący z bazy.

Przekształcenie układu warunków ograniczających do postaci bazowej względem nowej bazy odbywa się w dwóch krokach:

1. Normalizacja wiersza pivotu

```
pivot_row = constraints_matrix[pivot_row_idx, :]  
pivot_row /= pivot_element  
rhs_b[pivot_row_idx] /= pivot_element
```

2. Zerowanie wszystkich pozostałych wierszy

```
for i in range(num_resources):  
    if i != pivot_row_idx:  
        factor = constraints_matrix[i, pivot_col_idx]  
        constraints_matrix[i, :] -= factor * pivot_row  
        rhs_b[i] -= factor * rhs_b[pivot_row_idx]
```

```
if opt_type == "max":  
    best_result = max(valid_results, key=lambda x: x.get("profit", -np.inf))  
else:  
    best_result = min(valid_results, key=lambda x: x.get("profit", np.inf))
```

Najlepszy wynik jest wybierany w zależności od wybranego typu optymalizacji. Gdy użytkownik, chce znaleźć największy zysk, to wybieramy wynik dla którego wartość „profit” jest największa i oznacza to największy zysk. Gdy użytkownik szuka najmniejszego kosztu, to wybieramy wynik, w którym wartość „profit” jest najmniejsza i oznacza ona wtedy najmniejszy koszt.