

Nama : Muhammad Endriansyah Rahul Sudarsono
NIM : 20220029

Laporan praktikum algoritma aproksimasi

1. implementasi strategi algoritma aproksimasi pada masalah penjadwalan tugas dengan bahasa pemrograman Python

```
import numpy as np

def scheduling_approx(tasks, deadlines, processing_times):
    n = len(tasks)
    order = []
    time_left = deadlines - processing_times

    while len(order) < n:
        ratios = time_left / processing_times
        i = np.argmax(ratios)
        order.append(tasks[i])
        time_left = np.delete(time_left, i)
        tasks = np.delete(tasks, i, axis=0)
        deadlines = np.delete(deadlines, i, axis=0)
        processing_times = np.delete(processing_times, i, axis=0)

    return order

tasks = np.array(['A', 'B', 'C', 'D', 'E'])[:, np.newaxis]
deadlines = np.array([7, 6, 8, 10, 11])[:, np.newaxis]
processing_times = np.array([3, 4, 2, 5, 6])[:, np.newaxis]

order = scheduling_approx(tasks, deadlines, processing_times)
print("Urutan tugas yang direkomendasikan:", order)
```

Script di atas mengimplementasikan pemecahan masalah pemrograman linier menggunakan algoritma simplex dengan menggunakan fungsi linprog dari modul scipy.optimize dalam Python.

Langkah-langkah utama yang dilakukan dalam script ini adalah sebagai berikut:

- a. Impor modul numpy dan linprog dari scipy.optimize.

- b. Bentuk matriks koefisien batasan A, vektor batasan b, dan koefisien fungsi tujuan c menggunakan array NumPy.
- c. Gunakan fungsi linprog dengan argumen c sebagai koefisien fungsi tujuan, A_ub sebagai matriks koefisien batasan (dalam bentuk \leq), dan b_ub sebagai vektor batasan.
- d. Hasil dari pemecahan masalah linier disimpan dalam variabel res.
- e. Cetak nilai maksimum fungsi tujuan dengan -res.fun, di mana -res.fun adalah negasi dari nilai minimum yang dihasilkan oleh algoritma simplex.
- f. Cetak nilai variabel x dengan res.x, yang merupakan solusi optimum yang ditemukan oleh algoritma.

Dengan menggunakan algoritma simplex, script ini mencari solusi optimum yang memaksimalkan fungsi tujuan dengan memenuhi batasan linier yang diberikan. Output dari script ini adalah nilai maksimum fungsi tujuan dan nilai variabel x yang merupakan solusi optimum dari masalah linier yang diberikan.

2. implementasi strategi algoritma aproksimasi pada masalah LP dengan bahasa pemrograman Python menggunakan algoritma simplex

```
import numpy as np
from scipy.optimize import linprog

# Membentuk sistem persamaan linier dengan numpy
A = np.array([[1, 1], [-1, 2], [2, 1]])
b = np.array([6, 2, 5])
c = np.array([1, -2])

# Menyelesaikan sistem persamaan linier menggunakan algoritma simplex
res = linprog(c, A_ub=A, b_ub=b, method='simplex')

print('Nilai minimum fungsi tujuan:', res.fun)
print('Nilai variabel x:', res.x)
```

Script di atas mengimplementasikan pemecahan masalah pemrograman linier menggunakan algoritma simplex dengan menggunakan fungsi linprog dari modul scipy.optimize dalam Python.

Langkah-langkah utama yang dilakukan dalam script ini adalah sebagai berikut:

- a. Impor modul numpy dan linprog dari scipy.optimize.
- b. Bentuk matriks koefisien batasan A, vektor batasan b, dan koefisien fungsi tujuan c menggunakan array NumPy.
- c. Gunakan fungsi linprog dengan argumen c sebagai koefisien fungsi tujuan, A_ub sebagai matriks koefisien batasan (dalam bentuk \leq), dan b_ub sebagai vektor batasan.

- d. Hasil dari pemecahan masalah linier disimpan dalam variabel res.
- e. Cetak nilai minimum fungsi tujuan dengan res.fun, yang merupakan nilai minimum yang dihasilkan oleh algoritma simplex.
- f. Cetak nilai variabel x dengan res.x, yang merupakan solusi optimum yang ditemukan oleh algoritma.

Dengan menggunakan algoritma simplex, script ini mencari solusi optimum yang meminimalkan fungsi tujuan dengan memenuhi batasan linier yang diberikan. Output dari script ini adalah nilai minimum fungsi tujuan dan nilai variabel x yang merupakan solusi optimum dari masalah linier yang diberikan.

3. implementasi strategi algoritma aproksimasi pada masalah pengalokasian sumber daya dengan batasan menggunakan bahasa pemrograman Python

```
import numpy as np
from scipy.optimize import linprog

# Membentuk sistem persamaan linier dengan numpy
A = np.array([[ -3,  -2,  -4,  0], [-1,  -4,  0,  -3], [-2,  -3,  -5,  -4]])
b = np.array([-15, -10, -20])
c = np.array([-10, -8, -15,  0])

# Menyelesaikan sistem persamaan linier menggunakan algoritma simplex
res = linprog(c, A_ub=A, b_ub=b, method='simplex')

print('Nilai maksimum fungsi tujuan:', -res.fun)
print('Nilai variabel x:', res.x)
```

Script di atas mengimplementasikan pemecahan masalah pemrograman linier menggunakan algoritma simplex dengan menggunakan fungsi linprog dari modul scipy.optimize dalam Python.

Langkah-langkah utama yang dilakukan dalam script ini adalah sebagai berikut:

- a. Impor modul numpy dan linprog dari scipy.optimize.
- b. Bentuk matriks koefisien batasan A, vektor batasan b, dan koefisien fungsi tujuan c menggunakan array NumPy.
- c. Gunakan fungsi linprog dengan argumen c sebagai koefisien fungsi tujuan, A_ub sebagai matriks koefisien batasan (dalam bentuk \leq), dan b_ub sebagai vektor batasan.
- d. Hasil dari pemecahan masalah linier disimpan dalam variabel res.
- e. Cetak nilai maksimum fungsi tujuan dengan -res.fun, di mana -res.fun adalah negasi dari nilai minimum yang dihasilkan oleh algoritma simplex.

- f. Cetak nilai variabel x dengan $res.x$, yang merupakan solusi optimum yang ditemukan oleh algoritma.

Dalam masalah ini, script mencari solusi optimum yang memaksimalkan fungsi tujuan dengan memenuhi batasan linier yang diberikan. Output dari script ini adalah nilai maksimum fungsi tujuan dan nilai variabel x yang merupakan solusi optimum dari masalah linier yang diberikan.