

## EXERCISE 7: ALGORITMA GREEDY

perkuliahan tatap muka pertemuan 9

### 1 Integer Knapsack problem

Ingatlah bahwa pada Integer (1/0) Knapsack Problem, terdapat tiga strategi greedy yang dapat diterapkan untuk mendapatkan solusi untuk Knapsack Problem, yaitu: (1) Greedy by profit; (2) Greedy by weight; dan (3) Greedy by density. Namun, pada perkuliahan sebelumnya, disampaikan bahwa tidak ada strategi di antara ketiga strategi tersebut yang dapat menjamin solusi optimal.

**Tugas:** Berikan sebuah contoh Integer Knapsack Problem, beserta solusinya dengan menggunakan ketiga strategi (berdasarkan keuntungan, bobot, dan densitas), dan solusi optimal (berikan bukti/argumen untuk menunjukkan bahwa itu memang solusi optimal). Anda tidak diperbolehkan memberikan contoh yang sama dengan rekan Anda.

### 2 Penjadwalan kerja dengan batas waktu

Diberikan  $n$  pekerjaan yang akan dilakukan oleh mesin. Setiap pekerjaan diproses oleh mesin dalam **satu satuan waktu** (dalam hal ini satuan waktunya adalah jam) dan batas waktu setiap pekerjaan  $i$  adalah  $d_i \geq 0$ . Pekerjaan  $i$  akan memberikan keuntungan  $p_i$  jika dan hanya jika pekerjaan selesai sebelum tenggat waktu habis. Bagaimana memilih pekerjaan yang akan diproses oleh mesin agar keuntungan maksimal?

Diberikan 4 pekerjaan ( $n = 4$ ) dengan kriteria sebagai berikut:

- $(p_1, p_2, p_3, p_4) = (50, 10, 15, 30)$
- $(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$

Misalkan mesin mulai bekerja pada jam 6 pagi, maka kita memiliki kendala sebagai berikut:

Job	Deadline ( $d_i$ )	Must be done before
1	2 hours	8 am
2	1 hour	7 am
3	2 hours	8 am
4	1 hour	7 am

Misalkan  $J$  adalah himpunan pekerjaan, maka fungsi tujuan dari masalah ini adalah:

$$\text{Maximize } F = \sum_{i \in J} p_i$$

Ini berarti memaksimalkan total profit, yaitu jumlah nilai profit  $p_i$  untuk  $i = 1$  sampai dengan  $i = n$ .

- Himpunan solusi  $J$  dikatakan *layak* jika setiap pekerjaan di  $J$  dilakukan sebelum tenggat waktu.
- *Solusi optimal* adalah solusi layak yang memaksimalkan  $F$ .

**Tugas:** Selesaikan masalah ini dengan *exhaustive search*. Untuk ini, buat tabel yang berisi kumpulan solusi, urutan pemrosesan, keuntungan total, dan kelayakan (seperti di bawah).

- *Himpunan solusi*: kemungkinan subhimpunan dari pekerjaan yang dipilih
- *Urutan pemrosesan*: urutan pekerjaan yang diproses sehingga tenggat waktu dipatuhi.
- *Total keuntungan*: total keuntungan berdasarkan himpunan solusi yang layak
- *Kelayakan*: apakah himpunan solusi layak/tidak layak

Solusi sesuai urutan pemrosesan	Total profit ( $F$ )	Kelayakan solusi
{ }	0	feasible
{1}	50	feasible
⋮	⋮	⋮
{1, 2}	-	not feasible
⋮	⋮	⋮
{2, 1}	60	feasible
⋮	⋮	⋮

**Keterangan:**

- Pada tabel tersebut, himpunan solusi {1, 2} tidak layak karena jika job nomor 1 dikerjakan lebih dulu, maka mesin baru dapat digunakan pada pukul 8am, sementara job nomor 2 harus selesai pada pukul 7am.
- Himpunan solusi {2, 1} layak karena jika job nomor 2 dikerjakan lebih dulu, maka mesin dapat digunakan pada pukul 7am, dan mesin dapat digunakan untuk mengerjakan job nomor 1 yang harus selesai pada pukul 8am.
- Lengkapi tabel tersebut dengan semua kemungkinan solusi.

### 3 Traveling salesman problem

Diberikan daftar kota dan jarak antara setiap pasangan kota, tentukan rute terpendek yang mengunjungi setiap kota tepat satu kali dan kembali ke kota asal yang memiliki biaya terendah.

Misalkan simpul graf input  $G$  adalah:  $v_1, v_2, \dots, v_n$ , dan tur dimulai dari  $v_1$ . Simpul berikutnya dipilih secara “greedy”, dengan aturan berikut:

*Pada setiap langkah  $i$ , pilih simpul  $v_j$  (di antara simpul yang tersedia) yang bobot sisinya  $(v_i, v_j)$  diminimalkan.*

**Tugas:** Diberikan graf berikut. Terapkan algoritma Greedy untuk menemukan solusi TSP untuk graf ini. Tulis solusi TSP dan hitung bobot total solusi.

