



CII2K3 STRATEGI ALGORITMA

Topik 6: BRANCH AND BOUND







- 1. Penjelasan umum
- 2. Job assignment problem
- 3. 0/1 Knapsack problem
- 4. Travelling Salesperson Problem





Penjelasan umum







Penjelasan umum

- Branch and Bound biasanya digunakan untuk menyelesaikan persoalan optimisasi, yaitu permasalahan meminimalkan atau memaksimalkan suatu fungsi objektif, yang tidak melanggar batasan dari suatu permasalahan
- Branch and Bound umumnya menggunakan pendekatan yang mirip dengan Breadth First Search







Branch and Bound vs Backtracking

o Persamaan:

- ► Menggunakan *state-space tree* untuk pencarian solusi
- Tidak melanjutkan simpul yang tidak mengarah ke solusi
- ▶ Dalam kasus terburuk, kompleksitasnya bergantung dari banyaknya simpul yang mungkin ada di state-space tree

Perbedaan:

- Backtracking umumnya digunakan untuk permasalahan non-optimisasi (namun tetap dapat menyelesaikan permasalahan optimisasi), sedangkan Branch and Bound (BnB) umumnya digunakan untuk permasalahan optimisasi
- ► Pada BnB, diperlukan suatu cara penentuan batas (bound) nilai terbaik fungsi objektif pada setiap solusi yang mungkin
- Backtracking umumnya menggunakan pendekatan DFS, sedangkan BnB menggunakan beberapa aturan tertentu (yang paling umum: best-first search, yang merupakan pengembangan dari BFS)





Ide dasar

- Seperti backtracking, pada BnB kita membuat state-space tree untuk pencarian solusi
- Untuk menentukan apakah suatu simpul menjanjikan, digunakan perhitungan nilai bound
- Seperti pada BFS, kita mempertimbangkan semua simpul pada suatu tingkat (*level*, dengan menghitung nilai *bound* dari simpul-simpul tersebut
- Suatu simpul merupakan simpul yang tidak menjanjikan jika:
 - ► Nilai *bound* dari simpul tersebut tidak lebih baik daripada solusi terbaik sementara (*best solution so far*)
 - ► Simpul tersebut merepresentasikan solusi yang tidak layak (karena melanggar *constraint* yang ada)







Job assignment problem







Job assignment problem

Permasalahan:

Terdapat n pekerjaan dan n orang yang dapat mengerjakan pekerjaan-pekerjaan tersebut, dengan biaya yang beragam untuk setiap orang dan setiap pekerjaan yang ada.

Bagaimana kita dapat memberikan semua pekerjaan pada semua orang yang tersedia. sedemikian sehingga setiap orang yang ada mendapatkan tepat satu pekerjaan, dan didapatkan total biaya yang minimum?





State-space tree untuk assignment problem

- \circ Misalkan kita tinjau n pekerjaan dengan n orang, dimana $c_{i,j}$ merepresentasikan biaya orang ke-i untuk melakukan pekerjaan ke-j
- \circ Solusi dari *assignment problem* dapat ditulis sebagai $\{x_1.x_2,\ldots,x_n\}$, dimana x_i merepresentasikan pekerjaan yang dikerjakan oleh orang ke-i
- Kita dapat mencari solusi dengan mengisi solusi per komponen untuk mendapatkan solusi parsial per tahapnya
- o Artinya, pada setiap langkah ke-i, kita tinjau pekerjaan yang dapat dikerjakan oleh orang ke-i
- Sehingga, level pohon menyatakan urutan orang yang ditinjau, dan (simpul) anak menyatakan pekerjaan yang dapat dipilih untuk melaksanakan pekerjaan tersebut





Nilai bound untuk assignment problem

- Job assignment problem merupakan masalah minimasi, sehingga kita tinjau lower bound untuk permasalahan ini
- Total biaya yang didapatkan oleh solusi dari permasalahan ini, tidak akan lebih kecil daripada total biaya dari biaya terkecil untuk setiap pekerja yang ada
- Lower bound untuk setiap simpul, kemudian dapat disesuaikan dengan solusi parsial yang telah diambil







Contoh:

n = 4 Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

	1	2	3	4
Α	9	2	7	8
A B C D	9 6 5 7	4 8	3	7
С	5	8	1	8
D	7	6	9	4







Contoh:

n = 4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

dengan best-first-search:

• •	igan bost mot souron.						
	{*,*,*,*}						
	cost = 0						
	lb = 2 + 3 + 1 + 4 = 10						

$$cost = 0$$

 $lb = 2 + 3 + 1 + 4 =$







Contoh:

n = 4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

Keterangan: pilih simpul daun dengan Ib terbaik







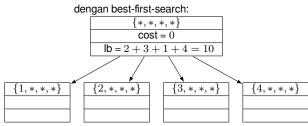
Contoh:

n = 4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:



Keterangan: cek semua anak dari simpul terpilih







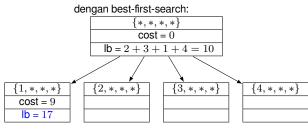
Contoh:

n=4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:



Keterangan:

cost = 9

b = 9 + 3 + 1 + 4 = 17







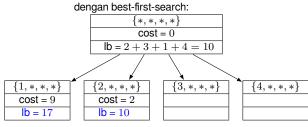
Contoh:

n=4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:



Keterangan:

cost = 2

b = 2 + 3 + 1 + 4 = 10







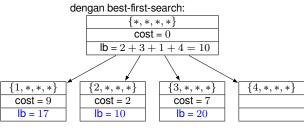
Contoh:

n=4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:



Keterangan:

cost = 7

b = 7 + 4 + 5 + 4 = 20







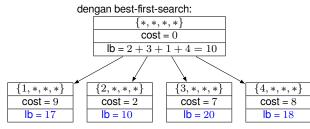
Contoh:

n=4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:



Keterangan:

cost = 8

b = 8 + 3 + 1 + 6 = 18







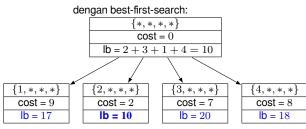
Contoh:

n = 4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:



Keterangan: pilih simpul daun dengan Ib terbaik







Contoh:

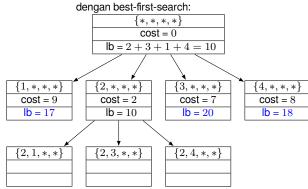
n = 4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

Keterangan: cek semua anak dari simpul terpilih









Assignment problem: contoh branch and bound dengan best-first-search:

Contoh:

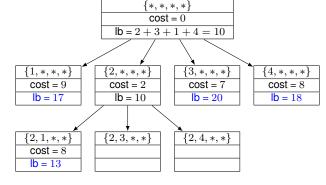
n=4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

1 2 3 4
A 9 2 7 8
B 6 4 3 7
C 5 8 1 8
D 7 6 9 4



$$cost = 2 + 6 = 8$$

 $lb = 2 + 6 + 1 + 4 = 13$







Contoh:

n=4

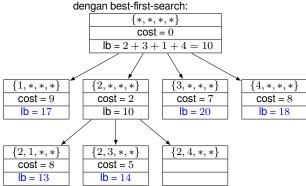
Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

$$cost = 2 + 3 = 5$$

 $lb = 2 + 3 + 5 + 4 = 14$









Contoh:

n=4

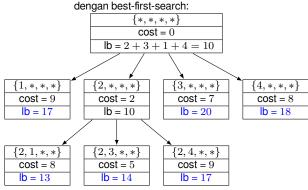
Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

$$cost = 2 + 7 = 9$$

 $lb = 2 + 7 + 1 + 7 = 17$









Contoh:

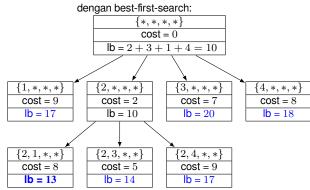
n = 4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

Keterangan: pilih simpul daun dengan Ib terbaik









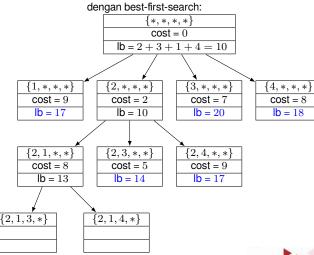
n = 4

Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

Keterangan: cek semua anak dari simpul terpilih







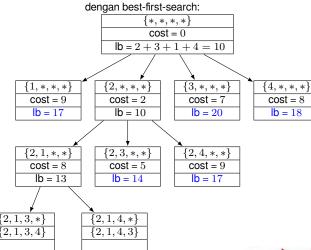
Job: 1, 2, 3, 4

Pekerja: A, B, C, D

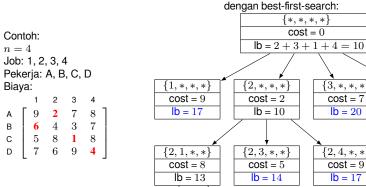
Biaya:

Keterangan:

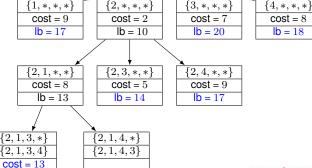
Reterangan: solusi parsial $\{2,1,3,*\}$ secara implisit membentuk solusi akhir $\{2,1,3,4\}$ solusi parsial $\{2,1,4,*\}$ secara implisit membentuk solusi akhir $\{2,1,4,4\}$ secara implisit membentuk solusi akhir $\{2,1,4,3\}$







cost =
$$2 + 6 + 1 + 4 = 13$$

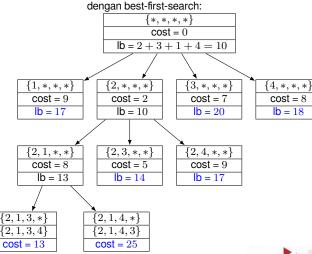








Keterangan: cost = 2 + 6 + 8 + 9 = 25





Contoh:

n = 4

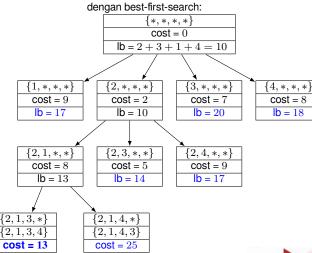
Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

Keterangan:

pilih simpul daun dengan lb terbaik







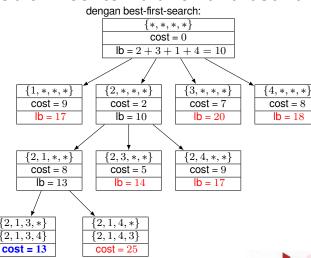
Job: 1, 2, 3, 4

Pekerja: A, B, C, D

Biaya:

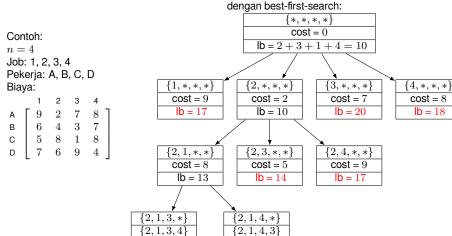
Keterangan:

simpul terpilih telah merepresentasikan hasil akhir, yang berarti daun lainnya tidak menjaniikan (karena tidak ada nilai Ib vang lebih kecil dari 13)









Jadi, solusinya adalah pekerja A, B, C, dan D, secara berurutan, diberikan pekerjaan 2, 1, 3, dan 4, sehingga didapatkan total biaya 13.

cost = 13

cost = 25





Latihan (1)

Suatu perusahaan kotak hadiah mempunyai lima pekerjaan yang berbeda, yaitu Pekerjaan 1, 2, 3, 4, dan 5.

Tugas-tugas tersebut akan diselesaikan oleh lima karyawan (A,B, C, D, dan E). Biaya penugasan seorang karyawan untuk tiap pekerjaan berbeda-beda. Data pada tabel di bawah ini menunjukkan biaya penugasan karyawan perusahaan kotak kado untuk tiap pekerjaan.

Karyawan	Biaya Pekerjaan 1	Biaya Pekerjaan 2	Biaya Pekerjaan 3	Biaya Peker
Α	17.000	15.000	19.000	21.000
В	12.000	17.000	20.000	18.000
С	16.000	21.000	23.000	21.000
D	15.000	19.000	19.000	17.000
E	14.000	20.000	23.000	19.000

Selesaikan permasalahan tersebut dengan menggunakan strategi Branch and Bound, agar biaya total yang harus dikeluarkan perusahaan tersebut seminimal mungkin.





0/1 Knapsack problem







0/1 Knapsack Problem

Permasalahan:

- \circ Diberikan n buah objek dan sebuah knapsack (tas punggung) dengan kapasitas bobot K
- o Setiap objek i mempunyai properti bobot w_i dan keuntungan (profit) p_i
- Objek mana saja yang perlu dimasukkan ke dalam knapsack sehingga keuntungan yang didapat bernilai maksimum?
- Catatan: total bobot objek yang dimasukkan ke dalam knapsack tidak boleh melebihi kapasitas knapsack



State-space tree untuk assignment problem

- o Misalkan kita tinjau kapasitas K serta n objek, dimana setiap objek ke-i, memiliki berat w_i dan profit p_i
- \circ Solusi dari 0/1 knapsack problem dapat ditulis sebagai $\{x_1.x_2,\ldots,x_n\}$, dimana $x_i=0$ jika objek ke-i tidak terpilih dalam solusi, dan $x_i=1$ jika objek ke-i masuk ke dalam solusi
- Kita dapat mencari solusi dengan mengisi solusi per komponen untuk mendapatkan solusi parsial per tahapnya
- o Artinya, pada setiap langkah ke-i, kita meninjau objek ke-i
- Sehingga, level pohon menyatakan urutan objek, dan (simpul) anak menyatakan kemungkinan terpilih atau tidaknya objek tersebut







Nilai bound untuk 0/1 knapsack problem

- Knapsack problem merupakan masalah maximasi, sehingga kita tinjau upper bound untuk permasalahan ini
- Total keuntungan yang didapatkan oleh solusi dari permasalahan ini, tidak akan lebih besar daripada total biaya yang telah didapatkan dari solusi parsial yang terbentuk, ditambah dengan sisa kapasitas dikalikan dengan densitas terbesar objek yang belum terpilih
- Oleh karenanya, kita perlu mengurutkan objek yang ada berdasarkan densitas, secara menurun (karena objek dengan densitas terbesar sebaiknya kita pertimbangkan terlebih dahulu)







Contoh:

$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25







dengan best-first-search:

{*,*,*,*}
W = 0, P = 0
ub = 100

Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

Keterangan:

Urutkan objek berdasarkan densitas, secara menurun

$$w_3 = 5$$

 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$







dengan best-first-search:

{*,*,*,*}
W = 0, P = 0
ub = 100

Contoh:
$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

Keterangan:

Pilih simpul daun aktif dengan ub terbaik

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$







dengan best-first-search:

Contoh: K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
$$p_1 = 40$$
$$p_1/w_1 = 10$$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

 $w_3 = 5$ $p_3 = 25$

 $p_3/w_3 = 5$

Cek semua anak simpul terpilih

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$





Contoh:

$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

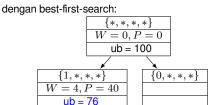
$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

Keterangan:
$$w_3 = 5$$

 $ub = P + ((K - W) \times (w_2/p_2))$ $p_3 = 25$
 $= 40 + ((10 - 4) \times 6)$ $p_3/w_3 = 5$
 $= 76$

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$







Contoh:

$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

 $w_3 = 5$

 $p_3 = 25$

$$ub = P + ((K - W) \times (w_2/p_2))$$

$$= 0 + ((10 - 0) \times 6)$$

$$= 60$$

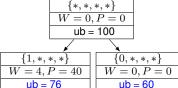
$$p_3 = 25$$

$$p_3 = 25$$

$$p_3/w_3 = 5$$

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$









dengan best-first-search:

Contoh: K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

Pilih simpul daun aktif dengan ub terbaik

$$w_3 = 5$$

 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$







Contoh:

$$K = 10$$

Keterangan:

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$

$$p_1 = 40$$

$$p_1/w_1 = 10$$

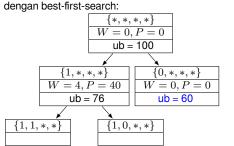
$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

Cek semua anak simpul terpilih

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$





Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$

 $p_2 = 42$
 $p_2/w_2 = 6$

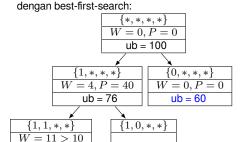
$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

Keterangan:

$$w_4 = 3$$

 $p_4 = 12$

 $p_4/w_4 = 4$





tidak layak



Contoh:

$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

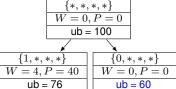
$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

 $w_3 = 5$

 $p_3 = 25$ $p_3/w_3 = 5$

dengan best-first-search:



×
$\{1, 1, *, *\}$
W = 11 > 10
tidak layak

$$\begin{cases} 1, 0, *, * \} \\ W = 4, P = 40 \\ \text{ub} = 70 \end{cases}$$

$$ub = P + ((K - W) \times (w_2/p_2))$$

$$= 40 + ((10 - 4) \times 5)$$

$$= 70$$

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$





Contoh:

$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$

 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

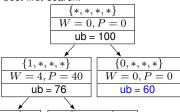
$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

Keterangan: Pilih simpul daun aktif dengan ub terbaik

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$

dengan best-first-search:



<i>y</i>	*
$\{1, 1, *, *\}$	$\{1,0,*,*\}$
V = 11 > 10	W = 4, P = 40
tidak layak	ub = 70







Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

Cek semua anak simpul terpilih

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

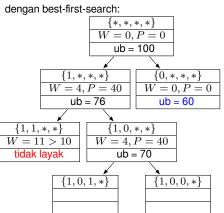
$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_3=5 \ p_3=25 \ {
m Keterangan:} \ p_3/w_3=5$$

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$







Contoh:

$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

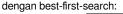
$$ub = P + ((K - W) \times (w_2/p_2))$$

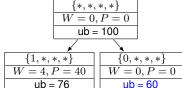
= 65 + ((10 - 9) \times 4)
= 69

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$

















Contoh:

$$K = 10$$

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

Keterangan:

$$ub = P + ((K - W) \times (w_2/p_2))$$

= 40 + ((10 - 4) \times 4)
= 64

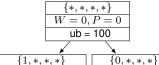
$$w_3 = 5$$

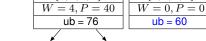
 $p_3 = 25$
 $p_3/w_3 = 5$

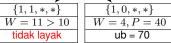
$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$

dengan best-first-search:







*	*
$\{1,0,1,*\}$	{1,0,0,*}
W = 9, P = 65	W = 4, P = 40
ub = 69	ub = 64





Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$\begin{array}{c} w_2 = 7 \\ p_2 = 42 \\ p_2/w_2 = 6 \end{array}$$

Keterangan:

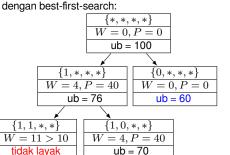
Pilih simpul daun aktif dengan ub terbaik

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

$$p_3/w_3 = 5$$

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$



	•
$\{1,0,1,*\}$	$\{1,0,0,*\}$
W = 9, P = 65	W = 4, P = 40
uh - 69	ub - 64





Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

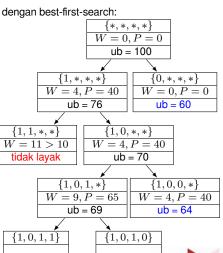
$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

Keterangan: Cek semua anak simpul terpilih

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$





 $\{0, *, *, *\}$

W = 0, P = 0

ub = 60

 $\{1,0,0,\overline{*}\}$

W = 4, P = 40

ub = 64

0/1 Knapsack problem: contoh branch and bound

Contoh:

K = 10

Keterangan:

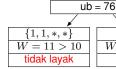
Ob	jek	Berat	Profit
Α	١	3	12
E	3	7	42
C	;	4	40
)	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

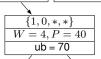
$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$



dengan best-first-search:



 $\frac{\{*, *, *, *\}}{W = 0, P = 0}$ ub = 100



 $\{1, *, *, *\}$

W = 4, P = 40







Contoh:

K = 10

Keterangan:

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

$$w_3 = 5$$

 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_4 = 3$$

 $p_4 = 12$
 $p_4/w_4 = 4$

$$\begin{cases} 1, 0, 1 \\ W = 9, P \end{cases}$$

$$\begin{cases} 1, 0, 0, * \\ W = 4, P = 40 \end{cases}$$

$$\text{ub} = 64$$

$$\begin{array}{c|c}
 & & \\
 \hline
 \{1,0,1,1\} \\
 \hline
 W = 12 > 10
\end{array}$$

$$\begin{array}{c|c}
 & \{1,0,1,0\} \\
 \hline
 W = 9, P = 65
\end{array}$$

$$\begin{cases}
 \{*, *, *, *\} \\
 W = 0, P = 0 \\
 ub = 100
 \end{cases}$$

$$\begin{array}{c|c} & & & & \\ \hline \{1,1,*,*\} & & & & \\ \hline W=11>10 & & & \\ \hline \end{array}$$

$$\{1,0,1,*\}$$

 $W = 9, P = 65$
 $U = 69$
 $\{1,0,0,*\}$
 $W = 4, P = 0$
 $U = 64$

ub = 70



 $\{0, *, *, *\}$

W = 0, P = 0

ub = 60

0/1 Knapsack problem: contoh branch and bound

Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

Keterangan:

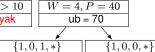
Pilih simpul daun aktif dengan ub terbaik

$$w_3 = 5$$
 $p_3 = 25$
 $p_3/w_3 = 5$

$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$



 $\{1, 1, *, *\}$



 $\frac{\{*, *, *, *\}}{W = 0, P = 0}$ ub = 100

 $\{1,0,*,*\}$



$$\{1, 0, 1, 1\}$$

 $W = 12 > 10$
tidak layak

dengan best-first-search:

 $\{1, *, *, *\}$

 $\overline{W} = \overline{4, P = 40}$

ub = 76

$$\{1, 0, 1, 0\}$$
 $W = 9, P = 65$
value = 65



Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

$$w_1 = 4$$
 $p_1 = 40$
 $p_1/w_1 = 10$

$$w_2 = 7$$
 $p_2 = 42$
 $p_2/w_2 = 6$

Keterangan:

Simpul terpilih telah merepresentasikan solusi utuh, yang berarti daun lainnya tidak menjanjikan (karena tidak ada nilai ub yang lebih besar dari 65)

$$w_3 = 5$$

 $p_3 = 25$
 $p_3/w_3 = 5$

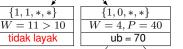
$$w_4 = 3$$
 $p_4 = 12$
 $p_4/w_4 = 4$

$$\{1, 0, 1, 1\}$$
 $W = 12 > 10$
tidak layak

dengan best-first-search:

 $\{1, 0, 1, *\}$

ub = 69





value = 65



Contoh:

K = 10

Objek	Berat	Profit
Α	3	12
В	7	42
С	4	40
D	5	25

Jadi, objek yang dipilih

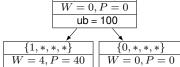
adalah objek C dan objek D

 $w_1 = 4$ $p_1 = 40$ $p_1/w_1 = 10$

 $w_2 = 7$ $p_2 = 42$ $p_2/w_2 = 6$

 $w_3 = 5$ $p_3 = 25$ $p_3/w_3 = 5$

 $w_4 = 3$ $p_4 = 12$ dengan best-first-search: {*, *, *, *}





tidak layak

$$\begin{cases} 1, 0, *, * \} \\ W = 4, P = 40 \\ \text{ub} = 70 \end{cases}$$



ub = 76

$$\begin{cases} 1, 0, 0, * \\ W = 4, P = 40 \end{cases}$$

$$ub = 64$$

ub = 60

$$\frac{\{1, 0, 1, 1\}}{W = 12 > 10} \quad W$$

$$\{1, 0, 1, 0\}$$
 $W = 9, P = 65$
value = 65

dengan total profit 65. $p_4/w_4 = 4$





Latihan (2)

Diketahui permasalahan 0/1 knapsack sebagai berikut. Selesaikan permasalahan tersebut dengan menggunakan algoritma Branch & Bound (Breadth First Search)

Objek	Berat	Profit
Α	10	\$100
В	9	\$63
С	7	\$56
D	4	\$12

dengan kapasitas knapsack K=18





Travelling Salesperson Problem







Travelling Salesperson Problem (TSP)

Permasalahan:

Diberikan n buah kota beserta jarak antar kota. Temukan perjalanan (tour) terpendek dari satu kota keberangkatan yang melalui setiap kota lainnya tepat satu kali dan kembali ke kota asal keberangkatan.





State-space tree untuk TSP

- \circ Misalkan kita tinjau graf tak berarah G dengan n simpul
- o Solusi dari TSP dapat ditulis sebagai v_1,v_2,\ldots,v_n,v_1 }, dimana $v_i\in V_G$ dan untuk $i\neq j,v_i\neq v_j$
- Kita dapat mencari solusi dengan mengisi solusi per komponen untuk mendapatkan solusi parsial per tahapnya
- Artinya, pada setiap langkah ke-i, kita meninjau simpul yang dikunjungi ke-i setelah kita mengunjungi titik asal
- Sehingga, level pohon menyatakan urutan simpul yang dikunjungi, dan (simpul) anak menyatakan kemungkinan simpul yang dikunjungi





Nilai bound untuk TSP

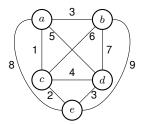
- TSP merupakan masalah minimasi, sehingga kita tinjau lower bound untuk permasalahan ini
- \circ Jika w(a,b) menyatakan bobot sisi dari a ke b, maka total bobot yang didapatkan oleh solusi dari permasalahan ini adalah penjumlahan dari n buah sisi, yaitu

$$w(v_1, v_2) + w(v_2, v_3) + \ldots + w(v_{n-1}, v_n) + w(v_n + v_1)$$

- o Total bobot tersebut dapat kita tulis juga sebagai $([w(v_1,v_n)+w(v_1,v_2)]+[w(v_2,v_1)+w(v_2,v_3)]+\ldots+[w(v_n,v_{n-1})+w(v_n,v_1)])/2$
- o Misalkan $V_G = \{u_1, \dots, u_n\}$, maka total bobot tersebut tidak akan kurang dari
 - $\lceil (s_1 + s_2 + \ldots + s_n)/2 \rceil$, dimana s_i adalah penjumlahan bobot dua sisi terkecil yang bersisian dengan simpul u_i



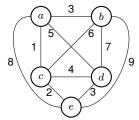












constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

Keterangan:

constraint dapat digunakan karena graf tidak berarah (tentunya constraint dapat diganti dengan constraint lainnya)

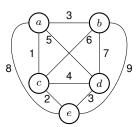






dengan best-first-search:





constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

The contagn is
$$10b = \lceil (1+3) + (3+6) + (1+2) + (3+4) + (2+3) \rceil / 2 \rceil$$

= $\lceil 28/2 \rceil = 14$

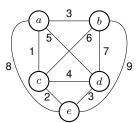






dengan best-first-search:





constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

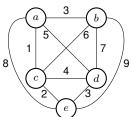
Keterangan:

Pilih simpul daun aktif dengan lb terbaik









constraint:

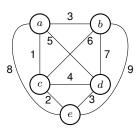
simpul c tidak dapat dikunjungi sebelum b dikunjungi

Keterangan:

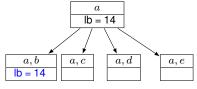
Cek semua anak simpul terpilih







dengan best-first-search:



constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

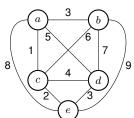
$$lb = \lceil ((1+3) + (3+6) + (1+2) + (3+4) + (2+3))/2 \rceil$$

= $\lceil 28/2 \rceil = 14$



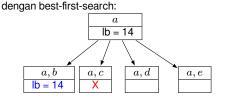






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

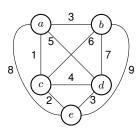


Keterangan:

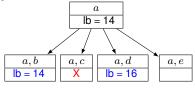
Melanggar constraint karena c dikunjungi sebelum b terkunjungi







dengan best-first-search:



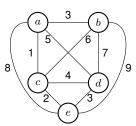
constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

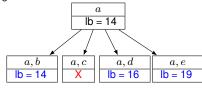
$$lb = \lceil ((1+5) + (3+6) + (1+2) + (3+5) + (2+3))/2 \rceil$$
$$= \lceil 31/2 \rceil = 16$$







dengan best-first-search:



constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

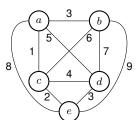
$$lb = \lceil ((1+8) + (3+6) + (1+2) + (3+4) + (2+8))/2 \rceil$$

= $\lceil 38/2 \rceil = 19$



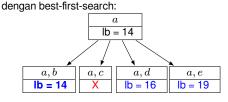






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



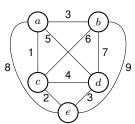
Keterangan:

Pilih simpul daun aktif dengan Ib terbaik



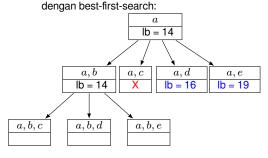






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



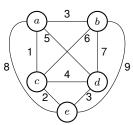
Keterangan:

Cek semua anak simpul terpilih



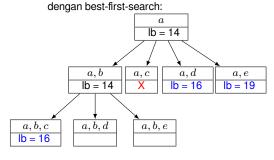






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



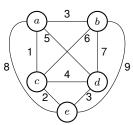
$$lb = \lceil ((1+3) + (3+6) + (1+6) + (3+4) + (2+3))/2 \rceil$$

= $\lceil 32/2 \rceil = 16$



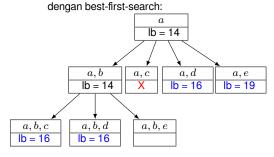






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



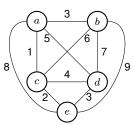
$$lb = \lceil ((1+3) + (3+7) + (1+2) + (3+7) + (2+3))/2 \rceil$$

= $\lceil 32/2 \rceil = 16$



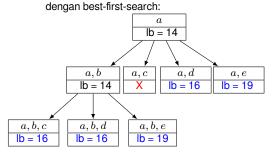






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



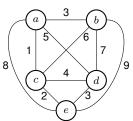
$$lb = \lceil ((1+3) + (3+9) + (1+2) + (3+4) + (2+9))/2 \rceil$$

= $\lceil 38/2 \rceil = 19$



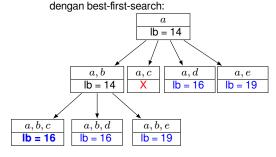






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



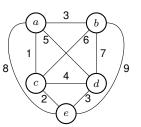
Keterangan:

Pilih simpul daun aktif dengan lb terbaik



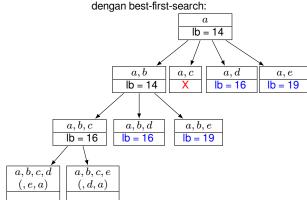






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



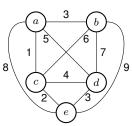
Keterangan:

Cek semua anak simpul terpilih



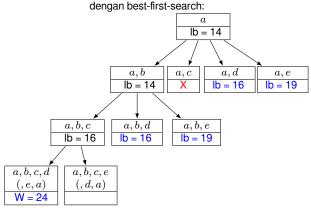






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

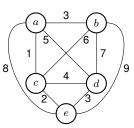


$$W = 3 + 6 + 4 + 3 + 8 = 24$$



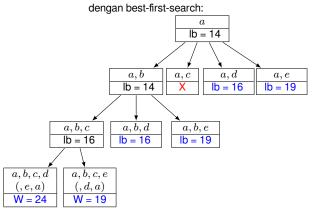






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

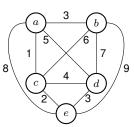


$$W = 3 + 6 + 2 + 3 + 5 = 19$$



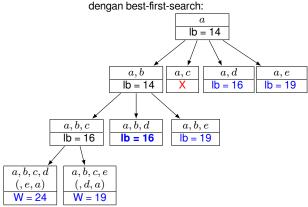






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



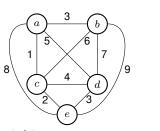
Keterangan:

Pilih simpul daun aktif dengan lb terbaik



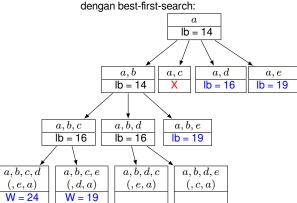






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



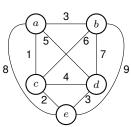
Keterangan:

Cek semua anak simpul terpilih



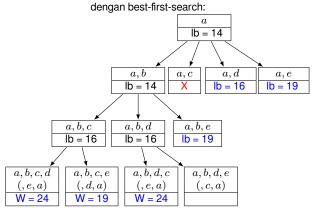






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

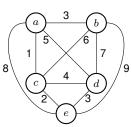


$$W = 3 + 7 + 4 + 2 + 8 = 24$$



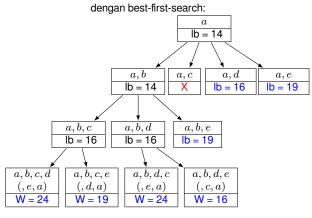






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

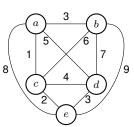


$$W = 3 + 7 + 3 + 2 + 1 = 16$$



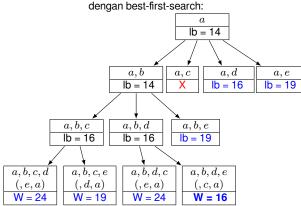






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



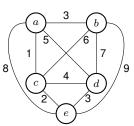
Keterangan:

Pilih simpul daun aktif dengan lb terbaik



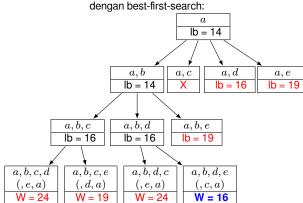






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi

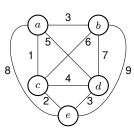


Keterangan:

Simpul yang merepresentasikan solusi akhir terpilih, sehingga daun lainnya tidak menjanjikan (karena tidak ada yang lebih bar dari 16)

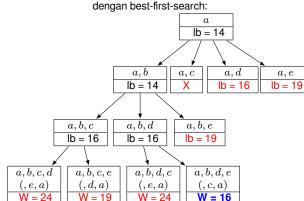






constraint:

simpul c tidak dapat dikunjungi sebelum b dikunjungi



Jadi, solusinya adalah rute a-b-d-e-c-a dengan total bobot 16.

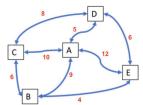






Latihan (3)

Dalam acara undian, dipilih 4 orang sebagai pemenang. Hadiah dari undian tersebut langsung diantarkan oleh pihak televisi ke rumah pemenang. Jarak antar keempat rumah pemenang tersebut dapat digambarkan sebagai berikut, dimana unit A adalah stasiun televisi tersebut, unit B, C, D, E adalah rumah keempat pemenang tersebut.



Gunakan teknik Branch and Bound untuk menentukan urutan rumah yang akan dikunjungi oleh pengirim hadiah tersebut agar jarak yang ditempuh minimum (asumsi, pengirim berangkat dari A dan harus kembali ke A di akhir pengiriman).





Referensi

- T. H. Cormen, C. E. Leiserson, R. L. Riverst, C. Stein. Introduction to Algorithms –3rd Edition, MIT Press, 2009.
- 2. A. Levitin. Introduction to The Design and Analysis of Algorithms –3rd Edition, Pearson, 2011.
- 3. R. Neapolitan, K. Naimipour. Foundations of Algorithms –5th Edition, Jones and Bartlett Learning, 2014.
- 4. Ir. Rinaldi Munir, M.T. Diktat Strategi Algoritmik IF2251.

 Departemen Teknik Informatika, Institut Teknologi Bandung
- 5. Ian Parberry. Lecture notes on Algorithm Analysis. Department of Computer Sciences, University of North Texas, 2001





THANK YOU

